

アダプティブ有限要素法

産業技術総合研究所 計算科学研究部門
手塚 明, 土田英二

Jan.23, 2004

Copyright © 2004, Akira Tezuka , Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 時間割(目安)(13:30~18:00) ■

13:30-14:30

・4つのキーポイント, 書籍の意図の説明, 導入(1, 2章)

14:30-15:00

・メッシュジェネレーション(3章)

休憩(10分)

15:10-16:00

・アダプティブFEM: 誤差推定(一次元, 二次元)(4章)

16:00-16:50

・アダプティブFEM: メッシュ制御法(5章)

休憩(10分)

17:00-17:30

・アダプティブFEM: 応用(量子MD)(6章)

17:30-17:50

・アダプティブFEM: 応用(マクロ計算力学)(6章)

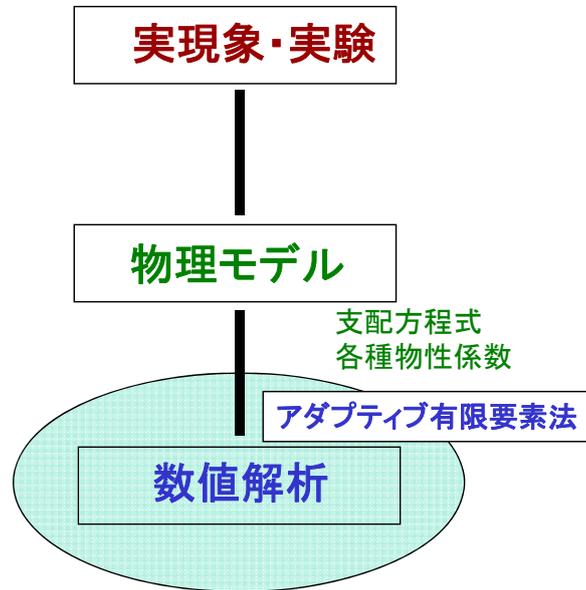
17:50-18:00

・コード説明・質疑応答(7章)

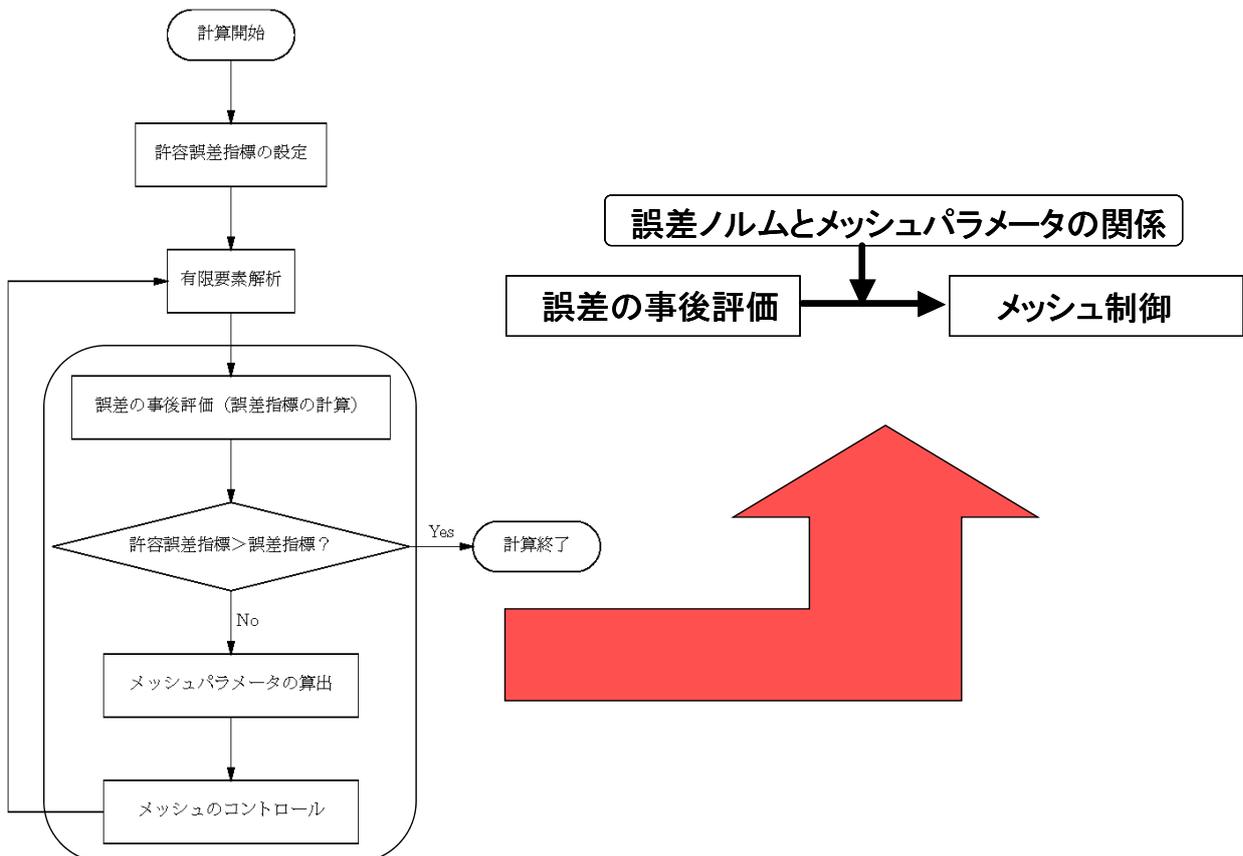
Copyright © 2004, Akira Tezuka , Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ アダプティブ有限要素法の守備範囲 ■

	使いやすさ	高速度解析	大規模解析	高精度解析	高複雑度解析
要素分割	レ		レ		レ
精度保証		レ		レ	レ
行列計算		レ	レ		
並列計算			レ		

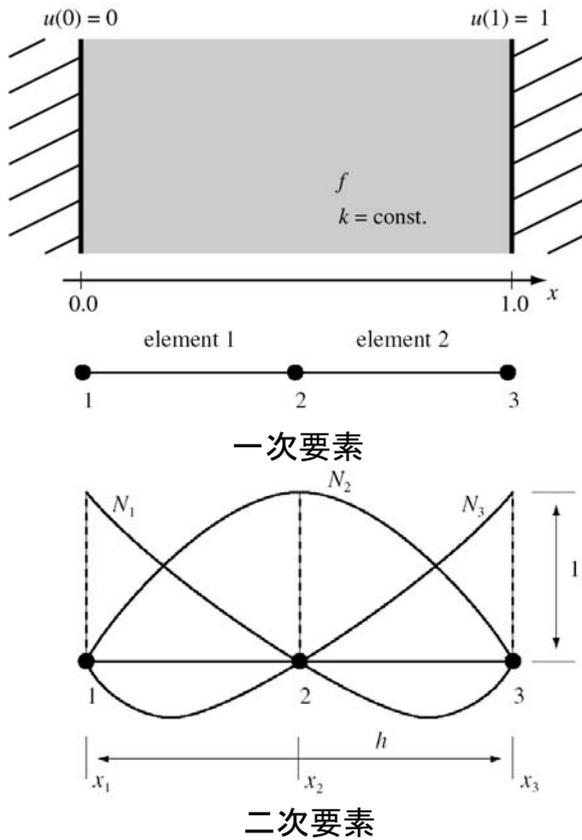


■ アダプティブFEMのフロー ■



例題: 一次元定常熱伝導問題

はじめに



$$k \frac{\partial^2 u}{\partial x^2} = -f$$

ここで、 u は温度(座標 x の関数)、 k ($=1$)は熱伝導係数、 f ($=1$)は熱発生率

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

$$u(1) = 1 \quad (u = 1 \quad \text{at} \quad x = 1)$$

有限要素法と誤差ノルム

はじめに

厳密解: $u = \frac{3x - x^2}{2}$

有限要素解 (一次要素):
$$\begin{cases} u^h = (u_2 - u_1)x + (u_1x_2 - u_2x_1) = \frac{5x}{4} \text{ (要素1)} \\ u^h = (u_3 - u_2)x + (u_2x_3 - u_3x_2) = \frac{3x+1}{4} \text{ (要素2)} \end{cases}$$

有限要素解 (二次要素): $u^h = \frac{3x - x^2}{2}$

L2誤差ノルム: $\left(\int_{element} (u - u^h)^2 dx \right)^{1/2}$ H1誤差ノルム: $\left(\int_{element} (u - u^h)^2 + \left(\frac{\partial u}{\partial x} - \frac{\partial u^h}{\partial x} \right)^2 dx \right)^{1/2}$

H1セミ誤差ノルム: $\left(\int_{element} \left(\frac{\partial u}{\partial x} - \frac{\partial u^h}{\partial x} \right)^2 dx \right)^{1/2}$

		温度(節点2)	誤差ノルム			ノルムの比		
			L ₂	H ¹	H ¹ セミ	L ₂	H ¹	H ¹ セミ
一次要素モデル	(要素1)	5/8	0.016	0.103	0.102	0.952	1.000	0.993
	(要素2)		0.016	0.103	0.102	0.975	1.000	0.982
二次要素モデル		5/8	0.000	0.000	0.000	1.000	1.000	1.000

ノルムの選択によって、「誤差の値」が異なる。

■ アダプティブFEM:4つのキーポイント ■

はじめに

#1. 違う性質の2つの組み合わせ

- ノルム 次元依存性なし
- メッシュ 次元依存性あり

#2. 誤差ノルムでの評価とは？

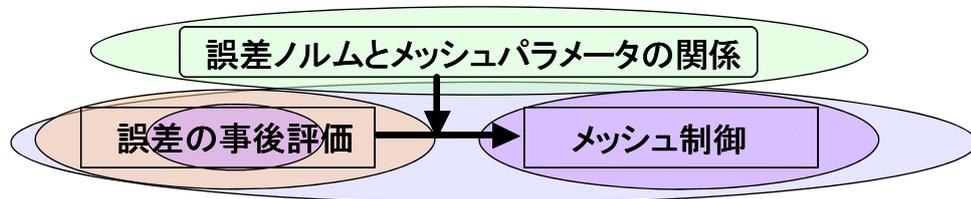
ノルムの性質

#3. 基礎理論を逸脱する形で理論実用化へ

標準誤差評価式

#4. パラメータ依存

誤差近似, メッシュ制御



■ #1. 違う性質の2つの組み合わせ ■

はじめに

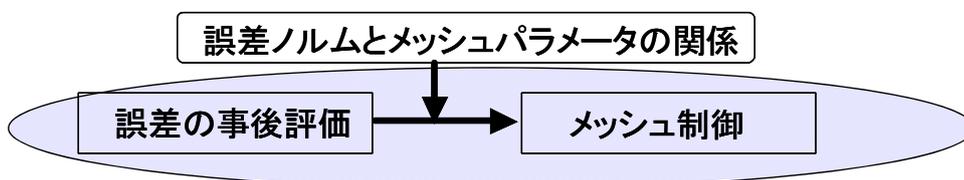
FEM理論=ノルム ←次元依存なし $\doteq \int_{\Omega} d\Omega$ (面積・体積)
 FEM解析=メッシュ ←次元依存あり $\doteq \Omega$ (形状)

$$L2 \text{ 誤差ノルム} : \left(\int_{element} (u^h - u)^2 dx \right)^{1/2}$$

$$H1 \text{ 誤差ノルム} : \left(\int_{element} (u - u^h)^2 + \left(\frac{\partial u}{\partial x} - \frac{\partial u^h}{\partial x} \right)^2 dx \right)^{1/2}$$

$$H1 \text{ セミ誤差ノルム} : \left(\int_{element} \left(\frac{\partial u}{\partial x} - \frac{\partial u^h}{\partial x} \right)^2 dx \right)^{1/2}$$

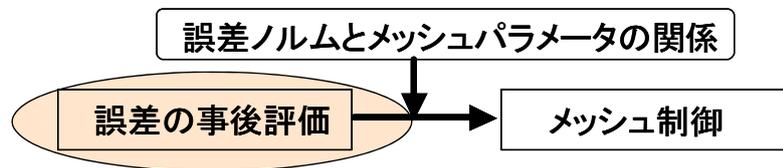
同じ「面積・体積」でも「形状」は違う。
 ノルムとメッシュの関係も同じ。



■ #2. 誤差ノルムでの評価とは？ ■

評価する定義量の違いにより、誤差の評価値も異なる

		温度(節点2)	誤差ノルム			ノルムの比		
			L_2	H^1	H^1 セミ	L_2	H^1	H^1 セミ
一次要素モデル	(要素1)	5/8	0.016	0.103	0.102	0.952	1.000	0.993
	(要素2)		0.016	0.103	0.102	0.975	1.000	0.982
二次要素モデル		5/8	0.000	0.000	0.000	1.000	1.000	1.000



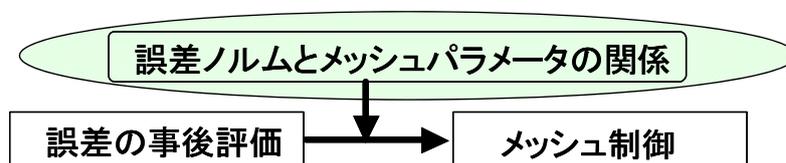
■ #3. 基礎理論を逸脱する形で理論実用化へ ■

標準誤差評価式 (p19, p91, p195)

→ 解析領域全体, 均質メッシュに有効

違う性質の2つをつなぐ架け橋

アダプティブFEMでは個々の要素, 非均質メッシュにもエイヤっと適用

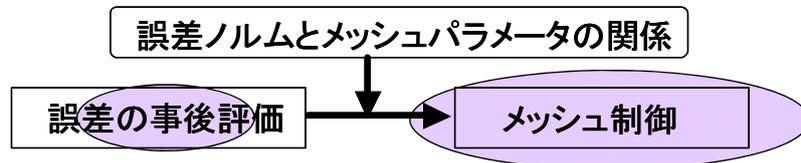


■ #4. パラメータ依存 ■

「メッシュ」が関わる近似部分でパラメータ依存が生じる

- ・誤差推定の近似
- ・メッシュ制御

→2.5, 4.4, 5.1-3, 6.1を参照下さい.



■ 書籍全体について ■

1. 「4つのキーポイント」がわかって頂けたら嬉しい.
2. 添付コードの特徴
 - ・**r, h, p法**すべて搭載
 - ・各種パラメータ依存の**体験可能**
 - ・使い物になるマトリックスソルバー
(skyline solver + **renumbering**)
 - ・**任意高次**有限要素関数の導出ルーチン
 - ・**postscript generator**による可視化(協力:鳥垣博士@日産)

■ チャプター毎のキーポイント ■

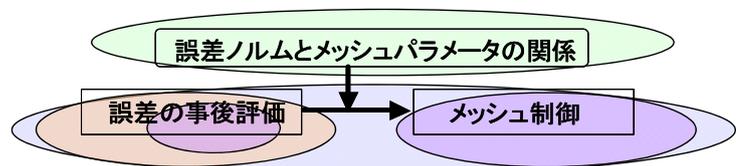
はじめに

1. はじめに
 - 位置づけ
2. 有限要素法のその精度
 - #1, #2, #3, #4
3. メッシュジェネレーション
 - #1
4. アダプティブ法における誤差推定
 - #1, #2, #3, #4
5. アダプティブ法におけるメッシュ制御
 - #1, #4
6. 応用例
 - #1, #2
7. リファレンスコード

A. 有限要素法

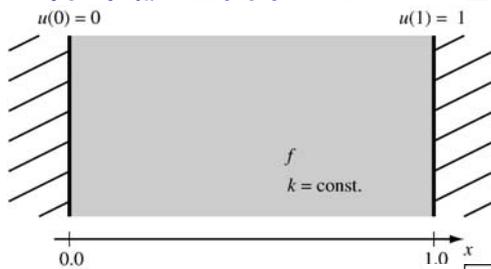
基本事項説明

B. アルゴリズム



Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 標準誤差評価式 ■ (差し当たって、厳密解既知の問題で定式化を考えます)



$$k \frac{\partial^2 u}{\partial x^2} = -f \quad f = 10(1 - e^x)/(1 - e)$$

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

$$u(1) = 1 \quad (u = 1 \quad \text{at} \quad x = 1)$$

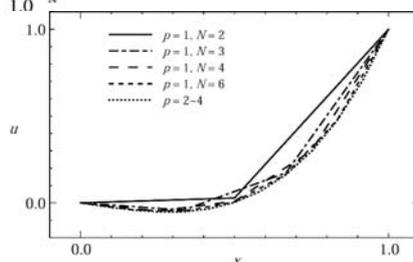


図2. 12 節点値と分割数, 次数

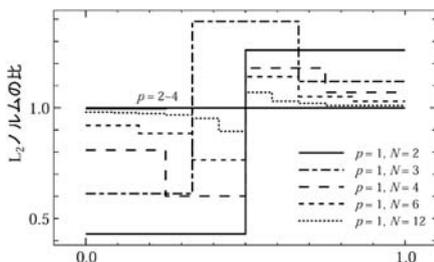


図2. 13 L_2 ノルム

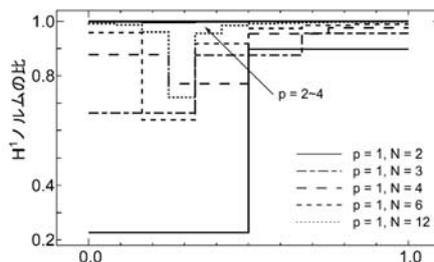


図2. 14 H_1 ノルム

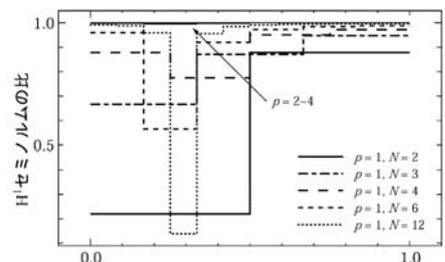


図2. 15 H_1 セミノルム

FE解ノルムと厳密解ノルムの比

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ ノルム, セミノルムの復習 ■

L₂ノルム

$$\begin{aligned} \text{内積: } (u, v) &= \int_{\Omega} uv \, d\Omega \\ \text{ノルム: } \|u\|_0 &= (u, u)^{1/2} \end{aligned} \quad \Longrightarrow \quad L2 \text{誤差ノルム: } \|u - u^h\|_0 = \left(\int_{\text{element}} (u - u^h)^2 dx \right)^{1/2}$$

H¹ノルム

$$\|u\|_1 = \left(\int_{\Omega} (uu + u_{,i}u_{,i}) d\Omega \right)^{1/2} \quad (\text{sum}, 1 \leq i \leq n)$$

H^sノルム

$$\|u\|_s = \left(\int_{\Omega} (uu + u_{,i}u_{,i} + u_{,ij}u_{,ij} + \cdots + u_{\underbrace{,ij\dots k}_{s \text{ indices}}}u_{\underbrace{,ij\dots k}_{s \text{ indices}}}) d\Omega \right)^{1/2} \quad (\text{sum}, 1 \leq i, j, \dots, k \leq n)$$

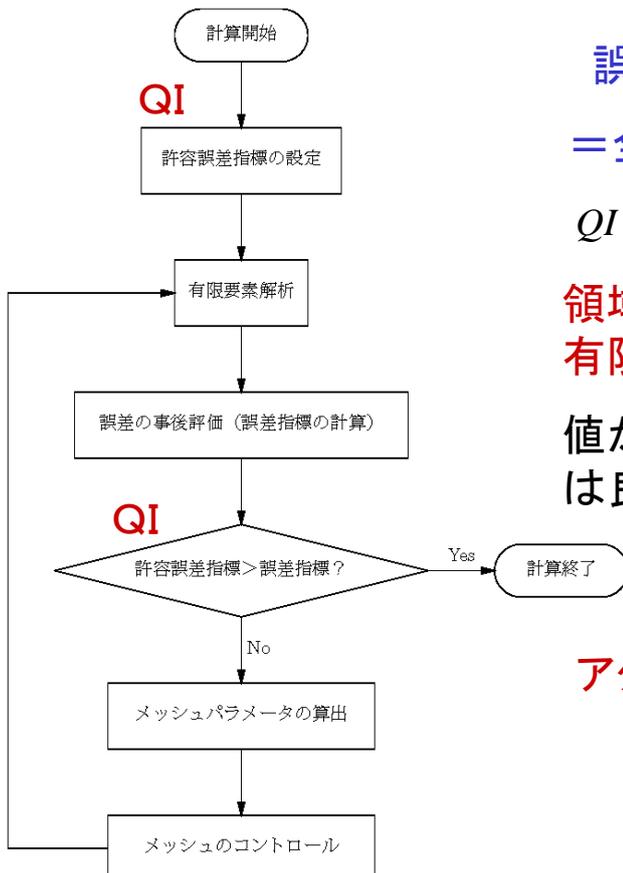
$$\|\mathbf{u}\|_s = \left(\int_{\Omega} (u_i u_i + u_{i,j}u_{i,j} + u_{i,jk}u_{i,jk} + \cdots + u_{\underbrace{,ijk\dots l}_{s \text{ indices}}}u_{\underbrace{,ijk\dots l}_{s \text{ indices}}}) d\Omega \right)^{1/2} \quad (\text{sum}, 1 \leq i \leq m, 1 \leq j, k, \dots, l \leq n)$$

H^rセミノルム

$$\|\mathbf{u}\|_r = \left(\int_{\Omega} u_{\underbrace{,ijk\dots l}_{r \text{ indices}}} v_{\underbrace{,ijk\dots l}_{r \text{ indices}}} d\Omega \right)^{1/2} \quad (\text{sum}, 1 \leq i \leq m, 1 \leq j, k, \dots, l \leq n)$$

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 誤差ノルム率QI ■



誤差ノルム率QI

= 全誤差ノルム / 全エネルギーノルム

$$QI = \|u - u^h\| / \|u\|$$

領域全体についてノルムの観点から有限要素の質を表わす指標

値が小さい程, 全体の有限要素の質は良い.

アダプティブ法の制御パラメータ

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 標準誤差評価式 ■ -誤差ノルムとメッシュをつなぐ架け橋-

$$\|u - u^h\|_m \leq ch^\beta \|u\|_r \quad \beta = \min \{k + 1 - m, r - m\}, \quad 0 \leq m \leq k$$

ここで、 k は有限要素補間関数の次数、 m, r はノルム内の微係数の最高次数であり、 c は h と u に依存しない正定数

$$h = \max_{1 \leq e \leq n_{el}} (h_e),$$

where

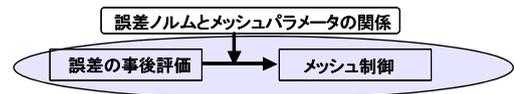
$$h_e = \frac{(\text{length of } \Omega_e)}{(\text{length of } \Omega)} \quad (\text{in } 1D),$$

$$h_e = \frac{(\text{area of } \Omega_e)^{1/2}}{(\text{area of } \Omega)^{1/2}} \quad (\text{in } 2D),$$

$$h_e = \frac{(\text{volume of } \Omega_e)^{1/3}}{(\text{volume of } \Omega)^{1/3}} \quad (\text{in } 3D)$$

ここで、 Ω, Ω_e は解析領域及び要素 e の領域を示し、 h^e は分割された要素 Ω_e における代表長さ

【代表長さ】←(キーポイント#1に関連)



■ 数値実験 ■

(キーポイント#2に関連)

誤差ノルムとメッシュパラメータの関係

誤差の事後評価 → メッシュ制御

L2誤差ノルム

一次要素

$$\|u - u^h\|_0 \leq ch^2 \|u\|_r$$

二次要素

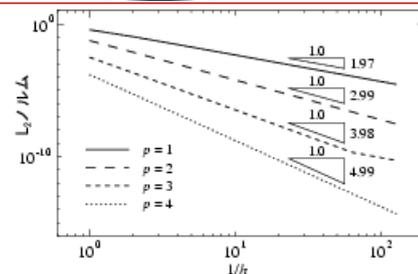
$$\|u - u^h\|_0 \leq ch^3 \|u\|_r$$

三次要素

$$\|u - u^h\|_0 \leq ch^4 \|u\|_r$$

四次要素

$$\|u - u^h\|_0 \leq ch^5 \|u\|_r$$



H1誤差ノルム

一次要素

$$\|u - u^h\|_1 \leq ch \|u\|_r$$

二次要素

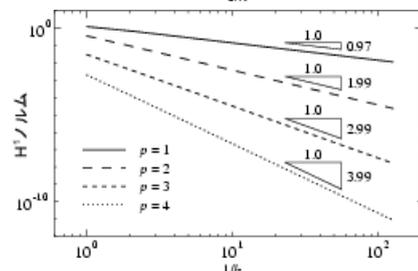
$$\|u - u^h\|_1 \leq ch^2 \|u\|_r$$

三次要素

$$\|u - u^h\|_1 \leq ch^3 \|u\|_r$$

四次要素

$$\|u - u^h\|_1 \leq ch^4 \|u\|_r$$



H1セミ誤差ノルム

一次要素

$$|u - u^h|_1 \leq ch |u|_r$$

二次要素

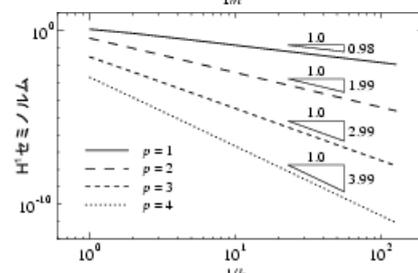
$$|u - u^h|_1 \leq ch^2 |u|_r$$

三次要素

$$|u - u^h|_1 \leq ch^3 |u|_r$$

四次要素

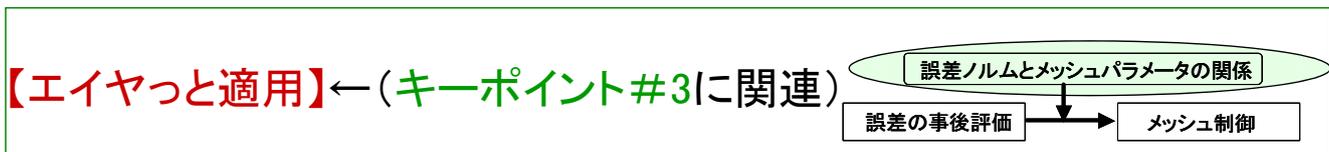
$$|u - u^h|_1 \leq ch^4 |u|_r$$



当たり前のように思っているが、実は...

標準誤差評価式は**解析領域全体**，**均質規則メッシュ**に有効
→ 全体的な細分化・高次化

これを，**エイヤット**，**個々の要素レベル**，**非均質メッシュ**に適用
(応用数学的な保証なし，結果オーライの適用(文献【39】))
→ 部分的細分化・高次化へ



■ 誤差ノルムとメッシュパラメータの関係 ■ (p. 20, p. 91)

えいやつとやったので，**要素レベル**で考える。

$$\|u - u^h\|_m \leq ch^\beta \|u\|_r \quad \beta = \min \{k + 1 - m, r - m\}, \quad 0 \leq m \leq k$$

k は有限要素補間関数の次数、 m, r はノルム内の微係数の最高次数であり、 c は h と u に依存しない正定数

アダプティブ法前後の定数 C の違いは、FE関数の次数 k やFE寸法 h と比べて、無視できるオーダー。 q は右辺ノルムのFE寸法 h の次数への寄与分(n 次元では $n/2$)とする

局所細分化 ($k^{old} = k^{new}$)

$$\frac{(h_e^{new})^{k^{new} + 1 - m + q}}{(h_e^{old})^{k^{old} + 1 - m + q}} = \frac{E_e^{new}}{E_e^{old}} \quad h_e^{new} = \left(\frac{E_e^{new}}{E_e^{old}} \right)^{\frac{1}{k+1-m+q}} h_e^{old}$$

局所高次化 ($h_e^{old} = h_e^{new}$)

↑ 定数 C を消したためにアダプティブ法前後の二つの式(等式とした)を割り算したもの

$$k_e^{new} = \frac{\log\left(\frac{E_e^{new}}{E_e^{old}}\right)}{\log h_e} + k_e^{old}$$

ここで、 E_e^{old} 、 E_e^{new} 、 h_e^{old} 、 h_e^{new} 、 k^{old} 、 k^{new} は厳密解アダプティブ法前後の誤差ノルム、要素寸法及び要素次数

■ E_e^{new} の計算法 ■ (厳密解既知の問題で定式化を考えています)

E_e^{new} は未知量. しかし, 最適メッシュでは各要素が等しい誤差ノルムを持つ性質を利用して, 指定したQIに対して, 近似可能.

$$E_e^{new} = \frac{QI}{100} / \sqrt{\frac{a(u^h, u^h)}{ne}}$$

ここで, ne は総要素数.

局所細分化

$$h_e^{new} = \left(\frac{E_e^{new}}{E_e^{old}} \right)^{\frac{1}{k+1-m+q}} h_e^{old}$$

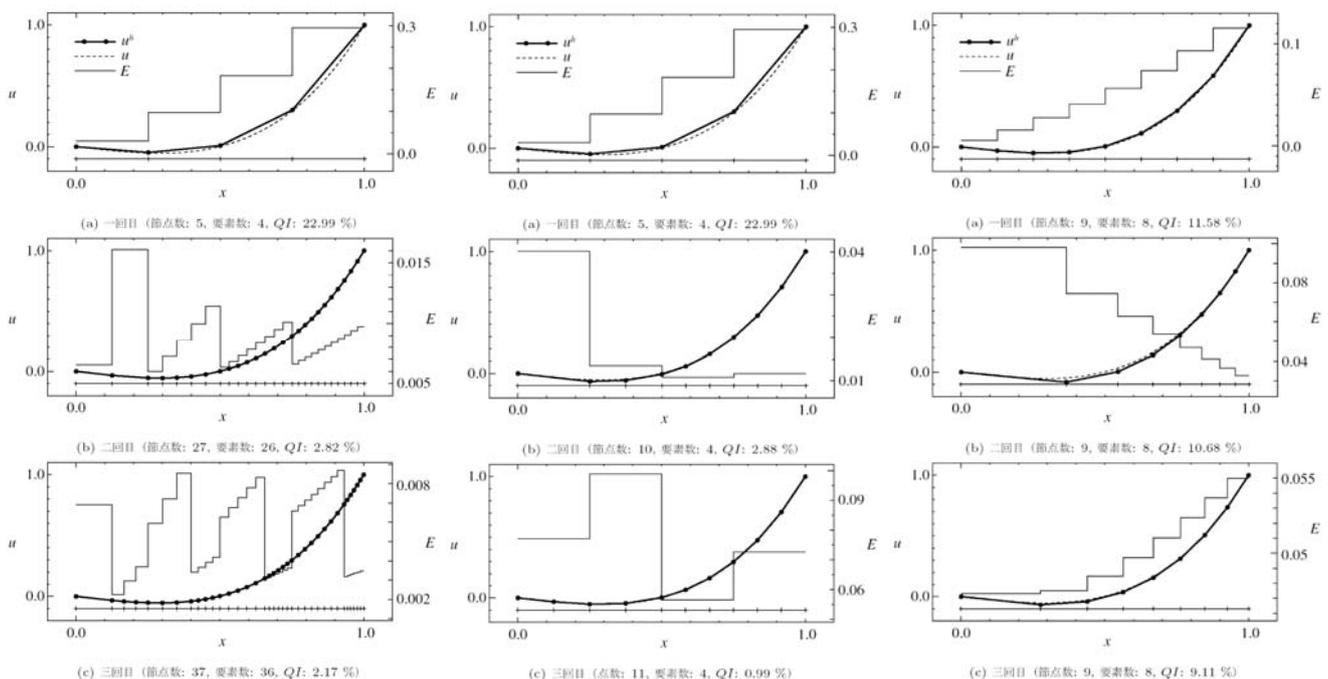
局所高次化

$$k_e^{new} = \frac{\log\left(\frac{E_e^{new}}{E_e^{old}}\right)}{\log h_e} + k_e^{old}$$

数値計算上の問題点

→ E_e^{old} が分母にあるため, E_e^{old} の値が小さい時には h_e^{new}, k_e^{new} の値導出は不安定になる可能性あり.

■ 厳密解アダプティブ法 ■ (図2.16-18均質メッシュ)(laptopPCでの動作デモ)
 「誤差ノルム定義」, 「誤差ノルムとメッシュパラメータの関係」を用いて,
 厳密解が既知の問題について, QIを指定し, 厳密解アダプティブ法.



局所細分化(check.exact.h) 局所高次化 節点移動(check.exact.r)

添付コードを用いて, 体験可能 (図(ps file)も自動生成) (協力: 鳥垣氏)

■ 有限要素解のポスト処理 ■ -アダプティブ法へ- (p. 30)

さて、**厳密解がわからない問題**でどうやって誤差推定するか？

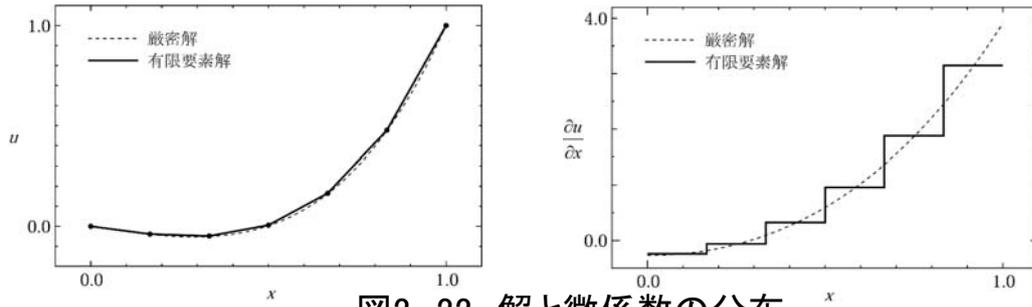


図2. 22 解と微係数の分布

一次要素の微分値は**階段状**の分布
階段状関数からスムーズな連続関数を導出する方法
 → **最小二乗法**

■ 最小二乗法 ■ (p. 31, p. 79)

階段状の一次微分関数 \mathbf{u}_x^h 求めたい一次微分連続関数 \mathbf{u}_x^*

$$\mathbf{u}_x^h = \sum_{i=1}^{node} \mathbf{u}_i^h \frac{\partial N_i}{\partial \mathbf{x}}$$

$$\mathbf{u}_x^* = \sum_{i=1}^{node} \mathbf{u}_{xi}^* N_i$$

最小二乗法

$$\text{Min}_{wrt \mathbf{u}_{xi}^*} \int_{\Omega} (\mathbf{u}_x^* - \mathbf{u}_x^h)^2 d\Omega$$

$$\frac{\partial (\mathbf{u}_x^* - \mathbf{u}_x^h)^2}{\partial \mathbf{u}_{xi}^*} = \sum_{j=1}^{全要素} \int_{要素 j} \sum_{k=1}^{node} 2 (\mathbf{u}_x^* - \mathbf{u}_x^h) N_k dx$$

$$= 2 \sum_{j=1}^{全要素} \int_{要素 j} \sum_{k=1}^{node} \left(\sum_{l=1}^{node} \mathbf{u}_{xl}^* N_l - \mathbf{u}_x^h \right) N_k dx = 0$$

$$\sum_{j=1}^{全要素} \int_{要素 j} \sum_{k=1}^{node} \sum_{l=1}^{node} \mathbf{u}_{xl}^* N_l N_k = \sum_{j=1}^{全要素} \int_{要素 j} \sum_{k=1}^{node} \mathbf{u}_x^h N_k dx$$

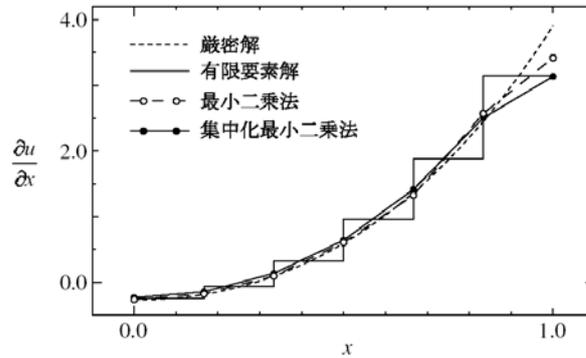


図2. 23

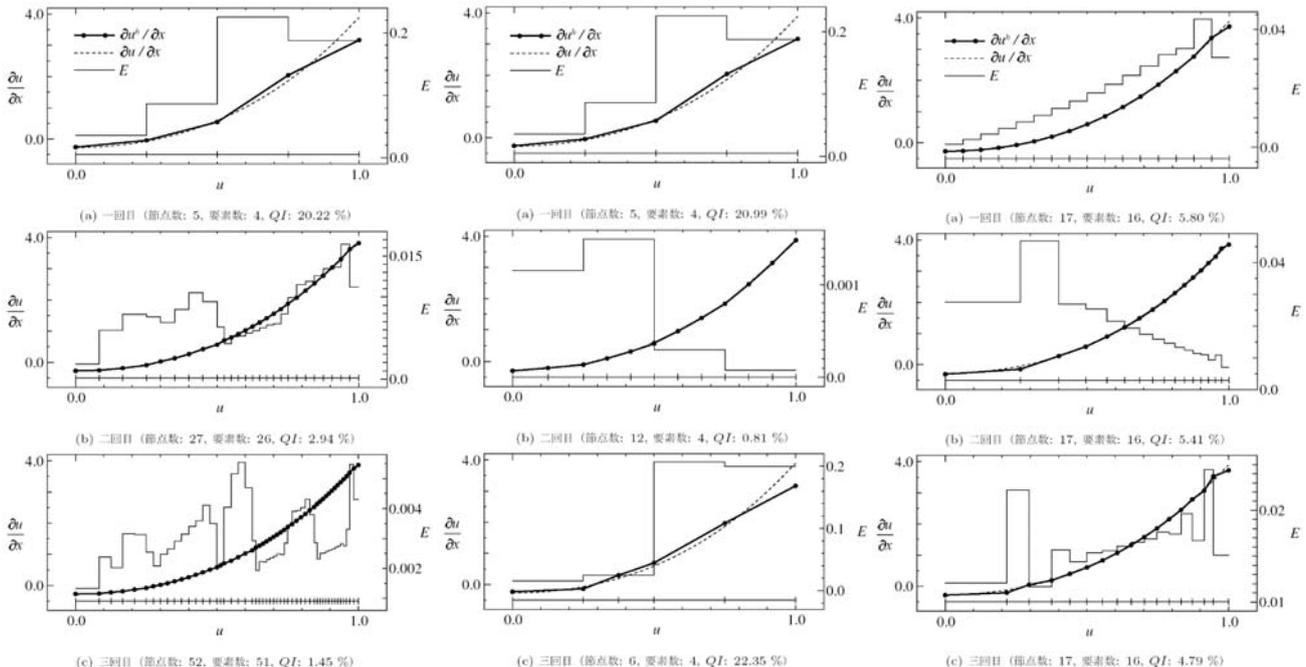
$$\frac{\partial u}{\partial x} \leftarrow u_x^* \left(= \sum_{i=1}^{node} u_{xi}^* N_i \right)$$

H₁セミ誤差ノルム (一次要素の場合)

$$E_e = \left(\int_{element\ e} \left(\frac{\partial u}{\partial x} - \frac{\partial u^h}{\partial x} \right)^2 dx \right)^{1/2}$$

(この方式は、L2誤差ノルム、H1誤差ノルムには使えない)

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST



局所細分化(check.zz.lsm.h) 局所高次化 (check.zz.lsm.p) 節点移動(check.zz.lsm.r)

均質メッシュ(最小二乗法)
(laptopPCで動作デモをします.)

添付コードを用いて、体験可能 (図(ps file)も自動生成) (協力: 鳥垣氏)

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ まとめ: 解のポスト処理によるアダプティブ法 ■ (p. 34, 47)

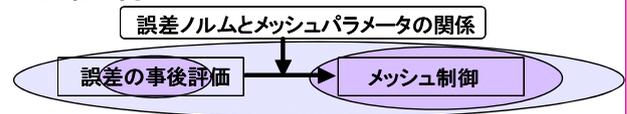
- ・最小二乗近似によるアダプティブ法は厳密解アダプティブ法と、次元問題では、実用的に問題がない。
- ・当たり前的事であるが、アダプティブ法の制御指標となる**QIの値自体も近似値**となる。

これがアダプティブ法のイメージ。
しかし、これでアダプティブ法はすべてか？

No !

- ・この計算方法は、(一次要素の場合), **H1セミ誤差ノルムにのみ有効**であり, **L2誤差ノルム, H1誤差ノルムには使えない**

- ・ **多次元の非均質メッシュでのパラメータ依存**
- ・ **多次元のメッシュ制御はもっと複雑**。
↑ (キーポイント #1, #4 に関連)



Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 2章のポイント ■

この章の point

誤差の評価のポイント

- 精度が要求される物理量は何か？

⇒ それに応じてノルムを選択。

- 種々の評価尺度

⇒ 節点値、誤差ノルム、有限要素解ノルムの収束

⇒ アダプティブ法では誤差ノルム評価

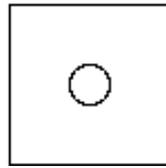
やっと、導入部分が終わりました。お疲れ様です。

肩の凝らない話、メッシュジェネレーションの話をして、その後、多次元のアダプティブFEMの誤差推定・メッシュ制御を話します。

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■メッシュ・ジェネレータの分類■

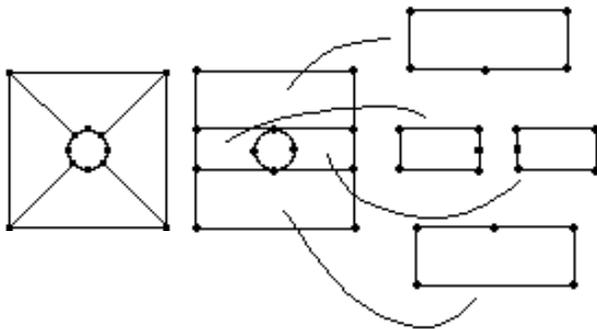
○ 機能による分類



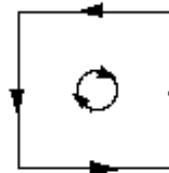
領域

・対話式 → 半自動

・非対話式 → 全自動



人間による前処理

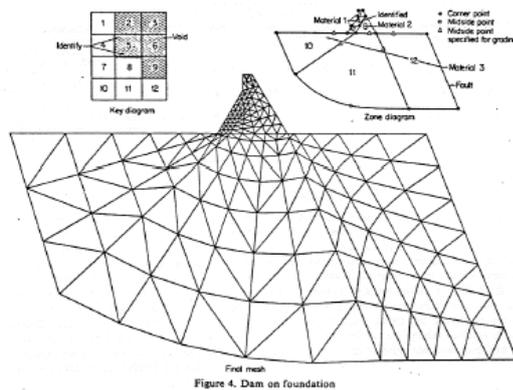


境界形状のみの入力データ

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

○ 方法による分類

・写像法 → 半自動



文献: Zienkiewicz, O.C. and Phillips, D.V.; An Automatic Mesh Generation Scheme for Plane and Curved Surfaces by Isoparametric Coordinate, Int. J. Numer. Methods Eng., 3 (1971), 519-528

・非写像法 → 全自動

アドバンシング・フロント法(逐次法)

ドロネイ網変換法

格子法

バブル法

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■メッシュをめぐる混乱■

1. 自動メッシュの意味

対話式も自動？

二次元: 非写像法OK

三次元: 非写像法？

「CAD→メッシュ・ジェネレータ→FEA」とシームレスに行かない。

2. 「メッシュが切れた」の意味

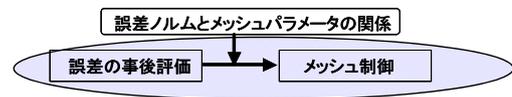
「使い物になるメッシュ」の数学的定義がない。

トポロジ的に切れていても仕方がない。

実計算では、解析手法とのトレードオフ。

キーポイント#1に関連した混乱

(理論とメッシュをつなぐものがない。(アダプティブFEMはその一つ))



Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■メッシュ・ジェネレータへの要求仕様■

- (1) 誰が使っても同一のメッシュが生成可能。
- (2) 境界データのみでメッシュ生成が可能。(非対話式)
- (3) 任意の形状に対応。
- (4) 要素寸法が一定。形状が良好。
- (5) ロバストで生成効率が良い。

- (6) (多重解析領域に適用可。)
- (7) (閉じた完全自動ではない。)
- (8) アダプティブメッシュ生成が可能。

実際には、

- ・領域形状が単純で凸体。
- ・領域を機械系CADより作成。(単純な図形の組み合わせ。)
- ・2.5次元の領域。
- ・要素寸法が一定でなくても良い。
- ・形状が良好でなくても良い。

→要素分割法としての難易度は極端に下がる。

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■「使い物になる」メッシュ生成作業省力化への方向■

1. 絶対に落ちない非対話式メッシャーの開発

TETRA要素生成でも時々落ちる.

2. 対話式メッシャーのインタフェース改善

特にHEXA要素. GUIの工夫.

部分領域メッシュ接合の簡単化 (STAR-CD, HICAD/CADAS)
(すべてTETRA要素分割も問題: シェルをTETRA分割???)

3. アダプティブ法の導入

4. メッシュレス法の研究

→どの方向が良いかは皆さん各人で考えてみて下さい.

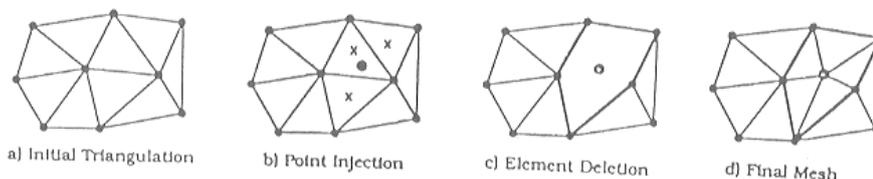
Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ドロネイ変換法■

・原理

点群に一意対応するドロネイ網(≠有限要素)を領域に適合。

ドロネイ網:d次元空間でd+1個の点からなる多面体の外接球内部に他の点が存在しない多面体の集まり



・Watsonのドロネイ網生成法 (他にも種々方法あり.)

0) 領域全体を包含する2個の四面体(ドロネイ網)を生成。

1) ドロネイ網に新たに点Aを加える。

2) 外接球内に点Aを有する四面体を消去。

3) 点Aと点Aの周りで1度のみ使用の面とで四面体を作成。

4) 1)から3)を全ての点を加え終わるまで続行。

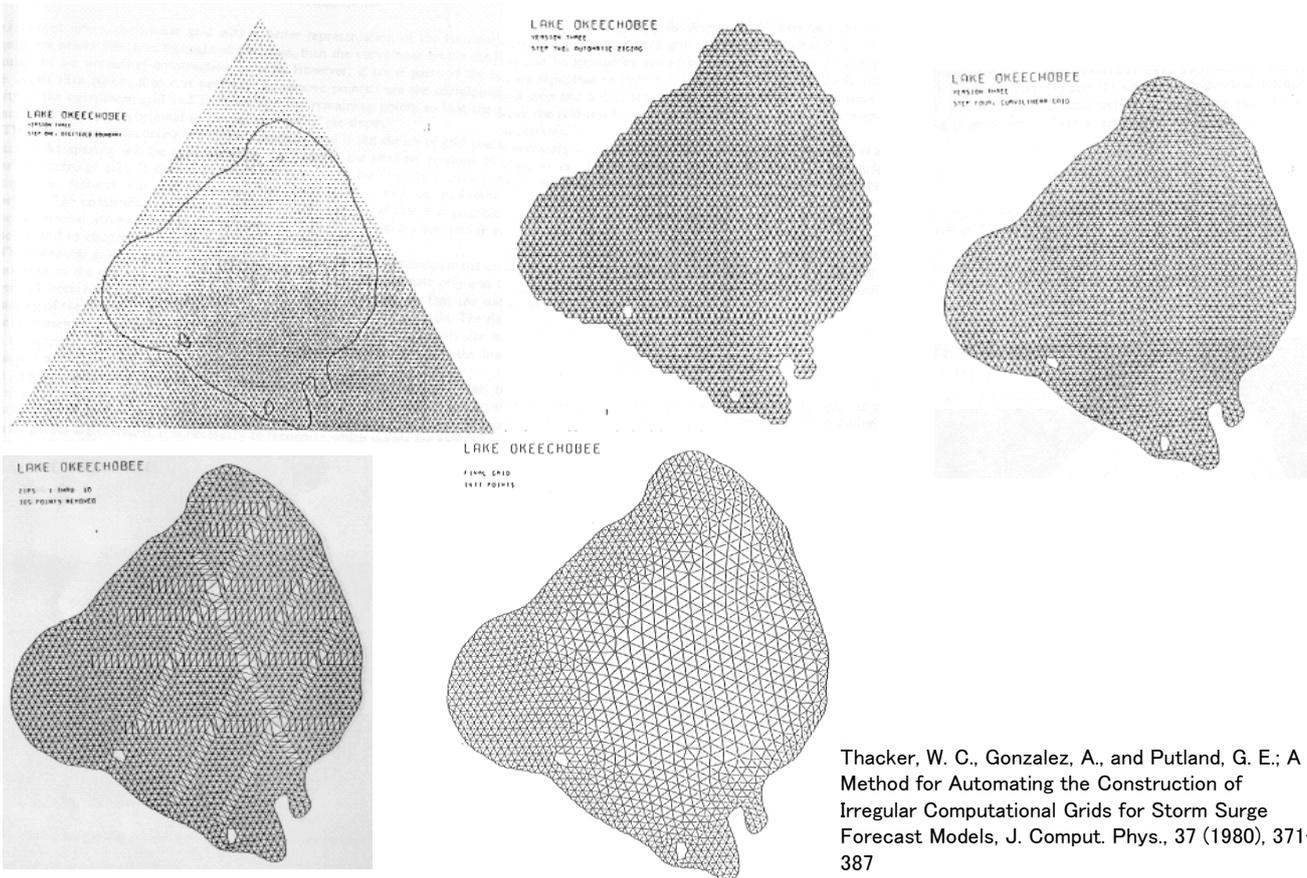
「うまく分布された」n点に対して $O(n^{2-1/d})$ のオーダー。

Watsonの方法を改良したものとしてSloanの方法等。

最高速のドロネイ網生成法でも高々 $O(n \log n)$ のオーダー。

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 格子法 ■

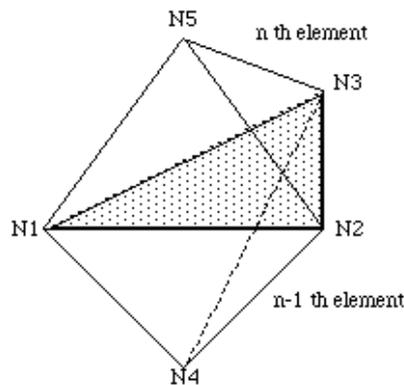


Copyright © 2004, Akira Tezuka , Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 逐次法(アドバンスング・フロント法) ■

・原理

既に生成された要素との干渉をしつつ、基面と要素寸法だけ離れた節点とを結び、逐次、要素生成を行う。



n番目の要素生成

四辺形・六面体版アドバンスング・フロント法にはペイビング法等が提案されている。

Copyright © 2004, Akira Tezuka , Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ バブル法 ■ (p.56-57)

要素生成に先立ち、解析により領域にバブルを充填。
重心を結ぶ事によって要素を生成。

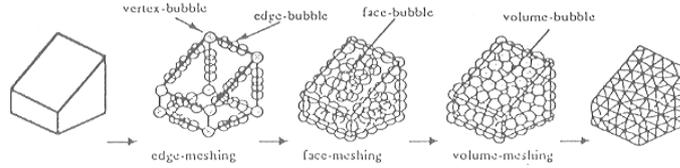


Figure 3. Meshing procedure

	domain	domain perimeter	element	force-balanced bubble config.
edge-meshing	curve	vertices	line segment	
face-meshing	surface	curves	triangle	
volume-meshing	volume	surfaces	tetrahedron	

Figure 4. The summary of mesh generation

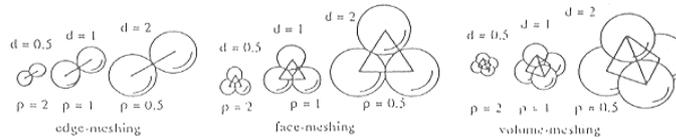


Figure 5. Density function and bubble radius

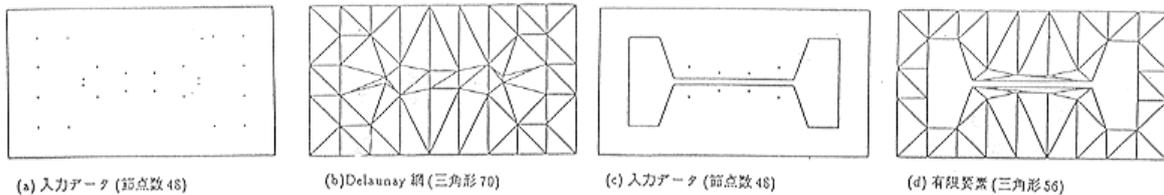
<http://www.tri.ibm.com/projects/meshing/mesh98.htm>

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

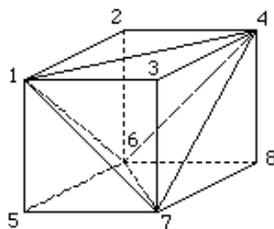
■ ドロネイ変換法とアドバンスング・フロント法の比較 ■ (p.55-56)

・ドロネイ変換法の問題点

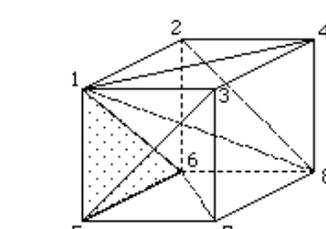
- (1) 極度に歪んだ要素生成の可能性
- (2) 境界との適合



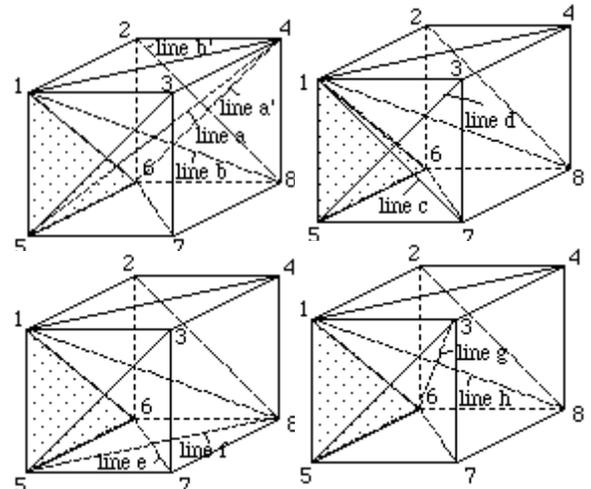
・アドバンスング・フロント法の問題点



分割パターン1



分割パターン2(途中)



頂点が探索不能! →

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ドロネイ変換法とアドバンシング・フロント法の比較■ (続き)

・ドロネイ変換法

内部はOK. 境界との適合が問題. 境界の精度は重要.

・アドバンシング・フロント法

境界メッシングは始めに指定. 内部が問題.

メッシュが切れている？実は

- ・上記の問題なくなるまで節点を追加？
- ・粗い分割から再分割している？

■TETRAメッシャー万能か？■

シェル形状をテトラ分割するのは問題.

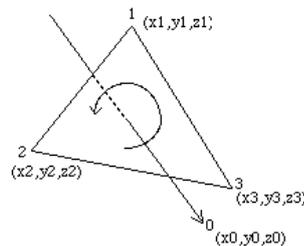
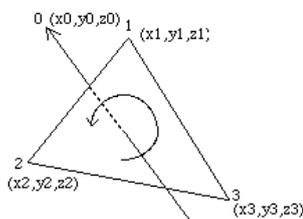
闇雲に細かいメッシュを切ってしまう.

→使いやすい写像系HEXAメッシャーも重要.

■基本ロジック■ (p.208-211)

1) 点が面のどちら側にあるかを判定するロジック

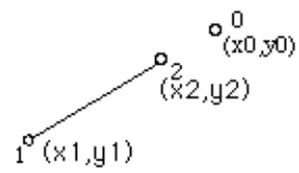
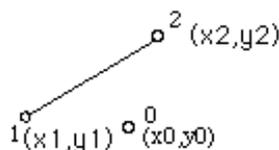
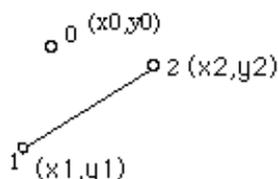
行列式D
$$D = \det \begin{bmatrix} x_1 - x_0 & x_2 - x_0 & x_3 - x_0 \\ y_1 - y_0 & y_2 - y_0 & y_3 - y_0 \\ z_1 - z_0 & z_2 - z_0 & z_3 - z_0 \end{bmatrix}$$



$D > 0 \rightarrow$ 点0は左ネジ方向 $D < 0 \rightarrow$ 点0は右ネジ方向

参考(2次元): 点0と線分1-2との位置関係を判定するロジック

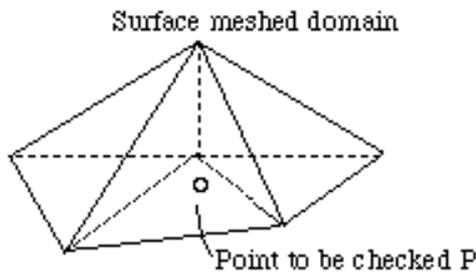
$$D = \det \begin{bmatrix} x_1 - x_0 & y_1 - y_0 \\ x_2 - x_0 & y_2 - y_0 \end{bmatrix}$$



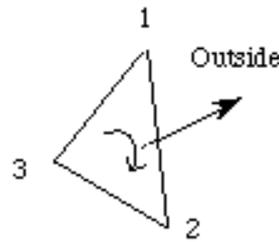
$D > 0 \rightarrow$ 線分1-2の左 $D < 0 \rightarrow$ 線分1-2の右 $D = 0 \rightarrow$ 線分1-2延長上

2) 点が任意領域の内外を判定する方法

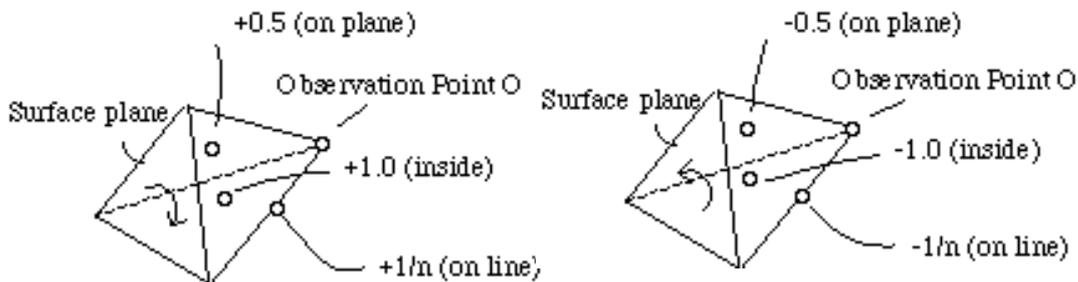
内部／境界に点Pを含むすべての四面体に対して、
 加算規則に基づく総和が0→点Oは領域外, 0以外→領域内(穴の場合は逆)
 ○ Observation point O ここでnは線分に連なる四面体数。



三角形表面分割された領域



表面三角形の向き

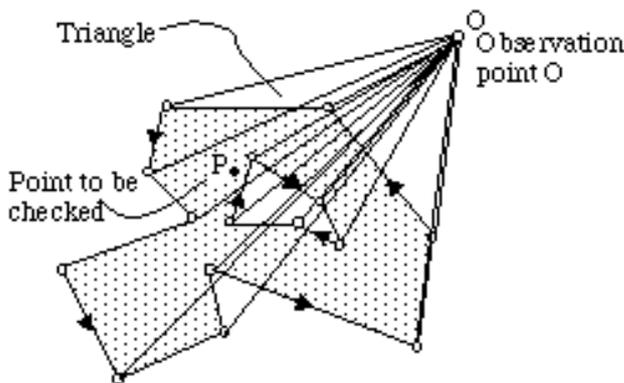


加算規則

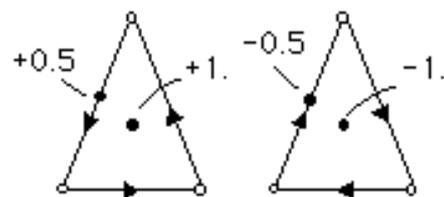
Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

参考(2次元): ある点が領域の内か外か判断するロジック

内部／境界に点Pを含む三角形に対して、加算規則による総和が0→
 点Oは領域外, 0以外→領域内(穴の場合は逆)



任意形状領域

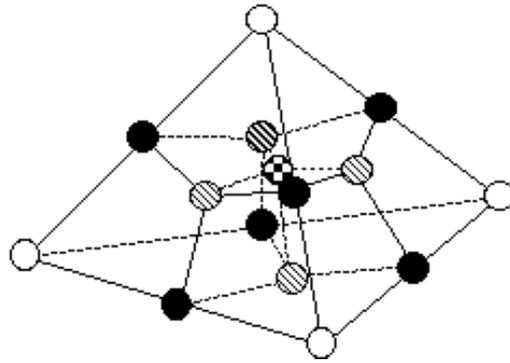


加算規則

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■四面体要素と六面体要素■

1. 四面体から六面体への変換はトポロジ的には可能。
しかし、**形状は良好ではない。**

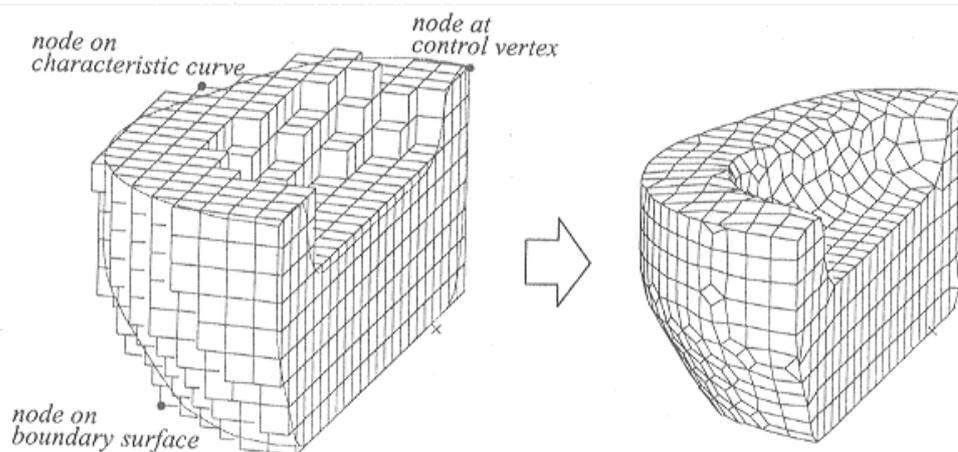


2. 使い物になる非対話式六面体メッシャーは皆無
HEXA dominated mesherの研究

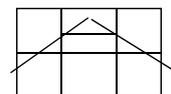
(S.J.Owen and S. Saigel, H-Morph: an indirect approach to advancing front hex meshing, Int. J. Numer. Meth. Engrg., 49,(200), pp.289-312)

Copyright © 2004, Akira Tezuka , Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■マスターグリッドアプローチ■(p.58)



うまく行かない場合がある。



Young-Kyu Lee, Three-Dimensional Automatic Mesh Generation for Remeshing in Finite Element Analysis of Metal Forming Processes, Ph.D. Dissertation, Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, (1998)

Copyright © 2004, Akira Tezuka , Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■メッシュレス法■

モデル生成の負担を解析手法で吸収. 計算機パワー向上が背景に.

1. 粒子法系

EFG, SPH,メッシュフリー法

W.K.Liu教授のサーベイ論文

(S.Li and W.K. Liu, Meshfree and Particle Methods and Their Applications,
http://tam.mech.northwestern.edu/wkl/_WingKamLiu/_links/publications.htm)

2. 固定メッシュ系

FEM系: ボクセル法, 有限被覆法, X-FEM,

差分法系: CIP法, VOF法

キーポイント

各手法に**パラメータ依存性**はないかどうか?

亀裂進展, 衝突付き抜け等, 既存FEM不得意分野はOK.

では, **通常**の解析のモデリングに使用可能か?

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■エレメント・フリー・ガラーキン法■ (p.58-64)

数値積分のためにバックグラウンドセルが必要.

近似関数は移動最小二乗法により積分点毎に構築

関数値及び微係数は領域内連続

$$u^h(\mathbf{x}) = \sum_{J=1}^m p_J(\mathbf{x}) a_J(\mathbf{x}) \equiv \sum_{I=1}^n \phi_I(\mathbf{x}) u_I$$

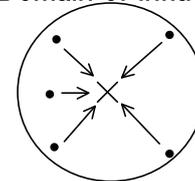
$$\text{Min}_{wrt a_j} \left(\sum_I w_I (\mathbf{x} - \mathbf{x}_I) \left[\sum_{J=1}^m p_J(\mathbf{x}) a_J(\mathbf{x}) - u_I \right]^2 \right)$$

m : number of basis p_j in approx. func.,

n : number of nodes in support domain

ϕ_I : approximation function in MLS

Domain of influence



• Node I
× Evaluation point
→ Interpolation

EFGMでのMLS近似関数

要点

- ・近似関数クロネッカ・デルタの条件 $N_i(x_j) \neq \delta_{ij}$ を満たしていない.
- ・影響範囲の設定により解の精度が大幅に変化
- ・影響範囲内の節点探索による低い計算効率

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■SPH法■

元来は、天体の運動解析のための手法
強形式版と弱定式版あり。

オリジナルはa partition of unityではないのでrigid body motionが記述できるように改良された手法が種々提案されている。

■フリーメッシュ法■

節点毎に周囲点に対して局所的にメッシュを
張り、近似関数を構築
ドロネイ法が局所メッシングには有効な性質
を利用
並列解析向き

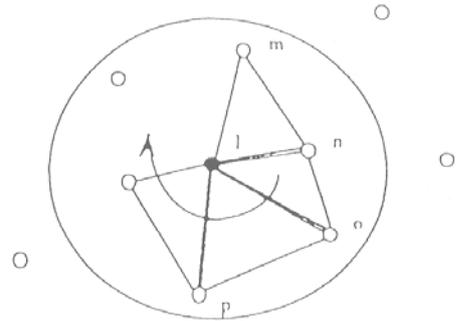
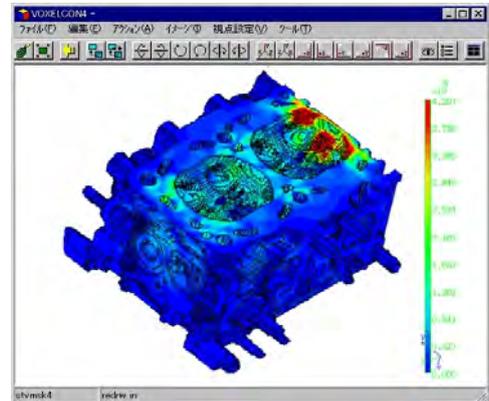
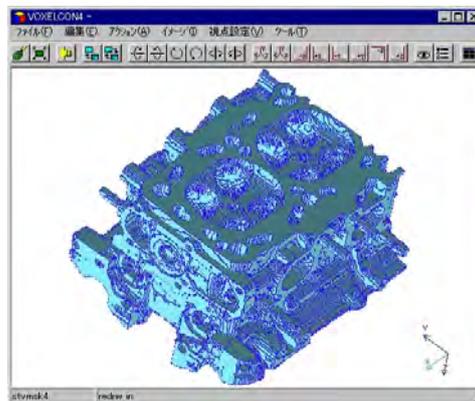


図 節点*l*まわりの一時的な三角形要素

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ボクセル法■

三次元スキャナーで取った情報をそのままメッシュに。
メッシュ生成のための簡易形状モデル生成が不必要。
均質化法ベースのトポロジー最適設計との組み合わせ。
ただし、メッシュはギザギザ。大規模解析が前提。
ボクセル形状は共通なので要素剛性マトリックス計算が一回で良い。
三次元解析をとにかく可能にする方法。



CTスキャナーからの解析例

<http://www.quint.co.jp/pro/vox/index.htm>

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

この章の point

■メッシュ

⇒ トポロジ的にメッシュ分割されているだけでは駄目。

■メッシュモデル生成

⇒ 難易度：

対話式メッシュ生成 ≪ 非対話式メッシュ生成

二次元メッシュ生成 ≪ 三次元メッシュ生成

四面体メッシュ分割 ≪ 六面体メッシュ分割

■三次元解析モデル生成問題点の解決方向

⇒ アドバンシングフロント法=内部の整合性合わせ

ドロネイ網変換法=外部境界との適合

メッシュレス法=モデリング内包解析手法の開発

さて、再び、アダプティブFEMに戻ります。多次元のアダプティブFEMの誤差推定・メッシュ制御を話します。

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

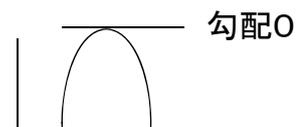
■アダプティブ法以前のメッシュ改善法■

1 歪みエネルギー密度 (SED) に基づく方法

歪みエネルギーの要素間の勾配 (SED) を指標にメッシュを細分化。

・問題点

特異点周りではSED (勾配) の値が小さめに出てしまう。
SEDとメッシュパラメータとの関係が明確ではない。



2 変位と節点位置の双方に関して、エネルギーノルムの最小化

・変位に対する最小化 (通常のFEM) (付録A1参照)

$$\text{Min}_u \frac{1}{2} \mathbf{u}^t \mathbf{K} \mathbf{u} - \mathbf{u}^t \mathbf{f} \Rightarrow \mathbf{K} \mathbf{u} = \mathbf{f}$$

・変位と節点位置に対する最小化

$$\text{Min}_{\mathbf{u} \& \mathbf{x}} \frac{1}{2} \mathbf{u}^t \mathbf{K} \mathbf{u} - \mathbf{u}^t \mathbf{f} \Rightarrow \mathbf{K} \mathbf{u} = \mathbf{f}, \frac{1}{2} \mathbf{u}^t \frac{\partial \mathbf{K}}{\partial \mathbf{x}} \mathbf{u} - \mathbf{u}^t \frac{\partial \mathbf{f}}{\partial \mathbf{x}}$$

・問題点

二番目の式は非線形であり、解くのが困難。
メッシュのトポロジーをも考慮にいれて解くのは不可能。

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■有限要素解析誤差■

有限要素法の質はメッシュの質に依存。
近似解法ゆえ、不適切なメッシュでは無効な解を得る。

■質の良いメッシュとは？■

一般に、歪みのない要素形状。

誤差大の領域では、要素寸法が小あるいは要素次数が高次。
要素寸法や要素次数の変化がスムーズ。

■メッシュの改善■

従来は、有限要素解析の経験のある人間によって行なわれた。

勘による試行錯誤的なプロセスが必要であった。

→アダプティブ法の出現。

■アダプティブ法の考え方■

誤差の事後推定に基づき、有限要素の改善を自動的に行なう方法論。

- ・誤差の事後推定 (a posteriori error estimate) とは
解析結果を基とする誤差の推定。

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ アダプティブFEMとは ■ (これまでの小まとめ) (p. 69)

- ・定量的, 応用数学を背景とした方法
- ・解析誤差のみの制御である。
(物理モデル設定誤差, メッシュ間投影誤差等, 他の誤差は考慮外)
- ・誤差をメッシュを介して制御
(誤差推定をメッシュに完全には反映できない)
- ・ノルムの観点からの誤差推定である。
(選択したノルムによって誤差推定の結果が多少異なる)
(ノルムの観点から収束性能等が評価されるので注意)
(メッシュの方向性の情報は無い。)
- ・誤差ノルム率に対して誤差を定量的に制御



Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 用語の復習 ■(p.70-71)

・ノルム(付録A2)

内積もノルムの一種. L2ノルム, H1ノルム, H1セミノルム等.
構造力学の分野では **エネルギーノルム**が使われる事が多い.

$$\|u\|_{energy} = \left(\int_{\Omega} \sigma_{ij} \cdot \varepsilon_{ij} d\Omega \right)^{\frac{1}{2}}$$

・誤差ノルム

誤差 $u - u^h$ (u : 厳密解, u^h : FE解) のノルム $\|u - u^h\|$ を誤差ノルムという. 厳密解 u は未知ゆえ, 厳密解 u の代わりに **有限要素近似関数より一次高次の関数 u^*** を有限要素解析解 u^h から導出して誤差ノルムを計算

$$E = \|u - u^h\| = \left(\sum_{j=1}^{ne} E_e^2 \right)^{1/2} = \left(\sum_{j=1}^{ne} \|u - u^h\|_e^2 \right)^{1/2}$$

・誤差ノルム率 QI (= 全誤差ノルム / 全エネルギーノルム)

領域全体についてノルムの観点から **有限要素の質** を表わす指標値が小さい程, 全体の有限要素の質は良い.

$$QI = \|u - u^h\| / \|u\| \approx \sqrt{\sum_{j=1}^{ne} E_e^2} / \|u^h\|$$

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 誤差事後推定法の数学的定式化 ■(p. 71)

・残差に基づく方法

誤差の成分 $u - u^h$ について **弱定式化** を要素毎に行ない, その残差と誤差の表面力成分を近似計算し, 誤差ノルムを求める方法. **支配方程式**, FEMの定式化が代わるごとに手直しが必要.

・補間誤差による方法

厳密解を補間した誤差ノルムが領域全体では有限要素解析誤差ノルムの上限を示すという補間定理を要素レベルにまで拡張し, 個々の要素で有限要素誤差ノルムを補間誤差ノルムで代用する方法

・Zienkiewicz-Zhuの方法(ZZ法)

有限要素解析解と有限要素解析解を平滑化した関数の差のノルムを誤差ノルムとみなす方法. 先ほど, 1次元で行った方法. 最も簡便で, **既存FEAコードへの入れ込みも容易**.

では, ZZ法で決め打ちか?

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 各誤差推定法の比較 ■

	対応誤差ノルム	汎用性	高次要素
残差法	すべてのノルム	問題毎にインプリ必要	個別に定式化必要
ZZ法	L_2 ノルム、 H^1 ノルム を除く	あり	汎用性あり
補間誤差法	すべてのノルム	あり	個別に定式化必要

表 4.1 誤差推定法

残差法の研究は良くされてきたが、ZZ法登場と共にZZ法が主流
しかし、ZZ法はn次FE要素で H^n セミにのみ有効。
ここでは、補間誤差法について、詳細に解説する。

■ 補間誤差法 ■

・補間誤差定理

厳密解を補間した誤差ノルムはFE誤差ノルムの上限を示す

$$\|u - u^h\|_{\Omega_e} \leq \|u - u^I\|_{\Omega_e}$$

・基本的な考え方

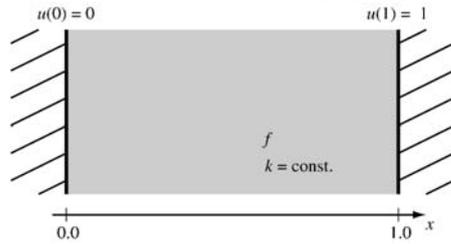
FE誤差ノルム $\|u - u^h\|$ の代わりに誤差の上限を表わす $\|u - u^I\|$ を要素e
の誤差ノルム E_e に使用する

・実際には

厳密解 u の代わりに有限要素近似関数より一次高次の関数 u^* をFE
解 u^h から導出して誤差ノルムを求める

$$E_e = \|u - u^I\|_e = \|u^* - u^I\|_e$$

■ 1次元定常熱伝導問題(一次要素) ■ (p. 73)



$$\frac{\partial^2 u}{\partial x^2} = -f$$

where $f = 10(1 - e^x)/(1 - e)$

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

$$u(1) = 1 \quad (u = 1 \quad \text{at} \quad x = 1)$$

$$N_1(x) = \frac{x_2 - x}{x_2 - x_1}$$

$$N_2(x) = \frac{x - x_1}{x_2 - x_1} \quad (\text{一次要素補間関数})$$

テイラー展開(高次項無視)

$$u_i = u + (x_i - x) \frac{\partial u}{\partial x} + (x_i - x)^2 \frac{\partial^2 u}{\partial x^2} / 2 + \text{HighOrderTerms}$$

厳密解 u を補間した関数 u^I

$$u^I(x) = \sum u_i N_i(x)$$

$$\begin{cases} u^I(x) - u(x) = ((x - x_1)(x - x_2) \frac{\partial^2 u}{\partial x^2} / 2 \quad (\text{補間誤差}) \\ \frac{\partial u^I}{\partial x} = (2x - x_1 - x_2) \frac{\partial^2 u}{\partial x^2} / 2 \quad (\text{補間誤差の微分}) \end{cases}$$

高次要素, 多次元要素でも同様. (p. 75-77)

■ 補間誤差ノルムから誤差ノルムへ ■ (p. 74)

1. f項ベースの方法

$$\frac{\partial^2 u}{\partial x^2} = -f$$

where $f = 10(1 - e^x)/(1 - e)$

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

$$u(1) = 1 \quad (u = 1 \quad \text{at} \quad x = 1)$$

$$\begin{cases} u^I(x) - u(x) = ((x - x_1)(x - x_2) \frac{\partial^2 u}{\partial x^2} / 2 \quad (\text{補間誤差}) \\ \frac{\partial u^I}{\partial x} = (2x - x_1 - x_2) \frac{\partial^2 u}{\partial x^2} / 2 \quad (\text{補間誤差の微分}) \end{cases}$$

→支配方程式依存, 多次元への拡張困難.

2. 誤差ノルムの定義式を用いる(通常の方法)

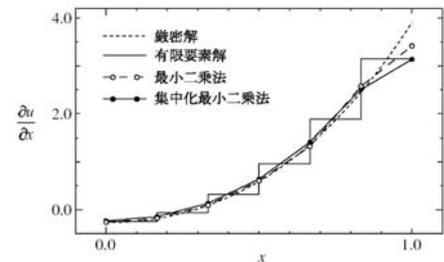
$$E_e = \|u - u^I\|_e = \|u^* - u^I\|_e$$

■ 誤差推定のアルゴリズム ■ (p. 76—)

	計算効率	要素歪みの影響	汎用性	高次要素	有限要素近似関数
最小二乗法	×	○	あり	△	必要なし
陽的最小二乗法	○	×	あり	×	必要なし
Iterative 最小二乗法	△	○	あり	△	必要なし
重み付き平均化法	○	×	あり	×	必要なし
局所パッチ法	△	△	あり	○	必要なし
トラクション法	△	○	問題毎にインプリ必要	○	必要

表 4.2 誤差推定のための解のポスト処理

FEMの性質利用: 局所パッチ法, トラクション法
データ平滑化: 最小二乗系, 重み付き平均化法



Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 陽的最小二乗法 ■ (p. 80)

・最小二乗法

$$\text{Min}_{wrt \mathbf{u}_x} \int_{\Omega} (\mathbf{u}_x^* - \mathbf{u}_x^h)^2 d\Omega \quad \Rightarrow$$

$$where \quad \mathbf{A} = \sum_{j=1}^{\text{全要素}} \int_{\text{要素 } j} \sum_{k=1}^{\text{node}} \sum_{l=1}^{\text{node}} N_l N_k dx,$$

$$\mathbf{u}^* = \mathbf{u}_i^*, \quad \mathbf{f} = \sum_{j=1}^{\text{全要素}} \int_{\text{要素 } j} \sum_{k=1}^{\text{node}} \mathbf{u}_x^h N_k dx$$

問題点: 領域全体節点数の**全体行列を次元の2乗回**(例えば, 2次元であれば2*2回)解く必要. 計算コストがかかる.

・陽的最小二乗 (=行列を解かない)

$$\mathbf{A}^d \mathbf{u}^* = \mathbf{f}$$

$$\mathbf{u}^* = \mathbf{f} / \text{diag}(\mathbf{A}^d)$$

where $\mathbf{A}^d = \text{lumped matrix of } \mathbf{A}$

欠点: 境界近くの値が小さめ. 高次要素で対角項がマイナスになる危険性

・繰り返し陽的最小二乗 (=行列を解かない+繰り返し計算)

$$\mathbf{u}^{*n} = \mathbf{u}^{*n-1} + (\mathbf{A} \mathbf{u}^{*n-1} - \mathbf{f}) / \text{diag}(\mathbf{A}^d)$$

収束すれば, **最小二乗と等価**

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ (陽的)最小二乗法のアルゴリズム ■ (p. 81—83)
 (subroutine lsmstr2 at adap.mesh2d.f)

```

ilump : 0=通常の最小二乗近似, 1=集中化最小二乗近似
jnt(i), i=1, 全節点数: バンド幅縮小のための番号付け替え対応表
jnt(元の節点番号)= バンド幅縮小後の節点番号)

call presky (skyline matrix solver のための skyline height の算出)

nk=0
do 10 kkl=1, 2(変位自由度数)
do 20 kk2=1, 2(微分自由度数)

    nk=nk+1
    if(nk.eq.1) then ( 全体行列 sk の初期化)
        全体ベクター f の初期化
        do 100 j=1, 要素数

            局所行列 a, 局所ベクター b の初期化
            [要素内節点数 node (=4), 要素コネクティビティ mej の設定]
            node=iepj(j+1)-iepj(j)
            do k=1,node
                mej(k)=met(iepj(j)-1+k)
            end do

            do m1=1,2(数値積分点)
                w1:s についての重み
                do m2=1,2(数値積分点)
                    w2:t についての重み
                    call uxyquad4 (積分点 s,t での FE 関数 sh(4)、変位の微分
                        uxy(2), det=determinant の値)
                    if(nk.eq.1) then
                        do k1=1,node(要素内節点数)
                            do k2=1,node(要素内節点数)
                                a(k1,k2)=a(k1,k2)+sh(k1)*sh(k2)*w1*w2*det
                            end do
                        end do
                    end if
                    do k1=1,node(要素内節点数)
                        b(k1)=b(k1)+sh(k1)*uxy(kk2)*w1*w2*det
                    end do
                end do
            end do
        20 continue
    10 continue

        if(ilump.ne.1) then
            call assemb2 [局所行列 a の全体行列 sk への入れ込み]
        else
            (lumped matrix; 対角集中化)
            do k3=1, node (要素内節点数)
                do k4=1, node (要素内節点数)
                    sk(mej(k3))=sk(mej(k3))+a(k3, k4)
                end do
            end do
        end if
        end if
        [局所ベクター b の全体ベクター f への入れ込み]
        do k=1, node (要素内節点数)
            i=jnt(mej(k))
            f(i)=f(i)+b(k)
        end do
        100 continue

        if(ilump.ne.1) then
            if(nk.eq.1) then
                kkk=0
            else
                kkk=2
            end if
            call skyline(for Ax=f :
                kkk=0; 前進消去・後退代入, kkk=1; 前進消去, kkk=2; 後退代入
            do i=1,np (全節点数)
                uxi(nk,i)=f(jnt(i))
            end do
        else
            do i=1,np
                uxi(nk,i)=f(i)/sk(i)
            end do
        end if
    end if
end do
end do
    
```

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 重み付き平均化法 ■ (Winslowの平滑化法 (p. 80, B1))

アルゴリズム

(subroutine lsmstr2 at adap.mesh2d.f)

```

do 10 j=1, 全要素数
    基底座標系の要素重心について
     $\frac{\partial \mathbf{u}_j^h}{\partial \mathbf{s}} \Big|_{\mathbf{s}=\mathbf{0}}$  を算出
     $\frac{\partial \mathbf{u}_j^h}{\partial \mathbf{s}} \Big|_{\mathbf{s}=\mathbf{0}}$  を  $\frac{\partial \mathbf{u}_j^h}{\partial \mathbf{x}} \Big|_{\mathbf{s}=\mathbf{0}}$  に変換
10 continue

do 20 i=1, 全節点数
    pp=0.
    do 40 k=1, 節点 i に連結する要素数
        j=節点 i に連結する要素番号
        r=要素 j の重心と節点 i との距離
         $\frac{\partial \mathbf{u}_i}{\partial \mathbf{x}} = \frac{\partial \mathbf{u}_i}{\partial \mathbf{x}} + \frac{\partial \mathbf{u}_j^h}{\partial \mathbf{x}} \Big|_{\mathbf{s}=\mathbf{0}} / r$ 
        pp=pp+1.
    40 continue
     $\frac{\partial \mathbf{u}_i}{\partial \mathbf{x}} = \frac{\partial \mathbf{u}_i}{\partial \mathbf{x}} / pp$ 
20 continue
    
```

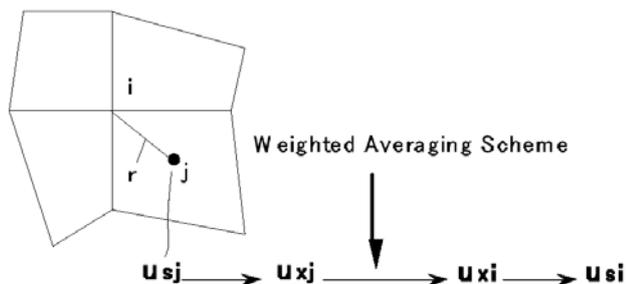


図 4.8 重み付き平均化法

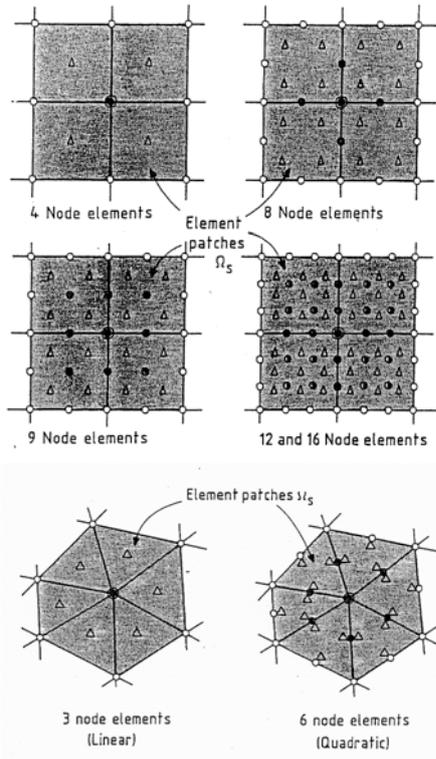
- ・均質なメッシュの内部節点に施す場合, 最小二乗法と等価
- ・連なる要素数が少ない境界節点では, 値の補正が必要
- ・非均質なメッシュではWinslowの平滑化法の特長より, 不必要な誤差が入る可能性



Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 局所パッチ法 ■ (p. 85-89)

EFGでのMLSから重み関数をなくした形。(歴史的には局所パッチ法の提案が前)
 節点に接続する要素群で基底を定義. 要素群内のサンプリングポイントで最小二乗.



局所パッチで求める微分関数

$$u_x^* = Pa$$

$$\text{where } P = [1 \ x \ y \ xy]^T$$

$$a = [a_1 \ a_2 \ a_3 \ a_4]$$

N個のサンプリング点でLSM(節点に接続の要素群積分点)

$$\text{Min} \sum_{i=1}^n (u_x^*(x_i, y_i) - u_x^h(x_i, y_i))^2$$

$$\sum_{i=1}^n P^T(x_i, y_i) P(x_i, y_i) a = \sum_{i=1}^n P^T(x_i, y_i) u_x^h(x_i, y_i)$$

$$\text{where } P^T(x_i, y_i) = [\{P(x_1, y_1)\}, \{P(x_2, y_2)\}, \dots, \{P(x_n, y_n)\}]$$

$$= \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ y_1 & y_2 & \dots & y_n \\ x_1 y_1 & x_2 y_2 & \dots & x_n y_n \end{bmatrix}$$

(MLSで重み関数が単位行列と等価)

$$Aa = b$$

$$\text{where } A = \sum_{i=1}^n P^T(x_i, y_i) P(x_i, y_i)$$

$$b = \sum_{i=1}^n P^T(x_i, y_i) u_x^h(x_i, y_i)$$

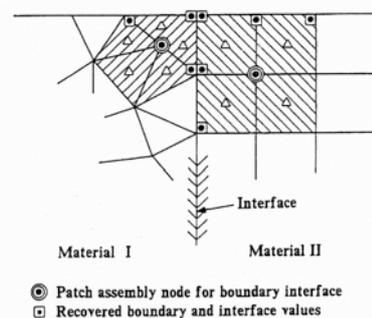
Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 局所パッチ法 ■ (p. 85-89)

計算には基底の数以上のサンプリング点が必要

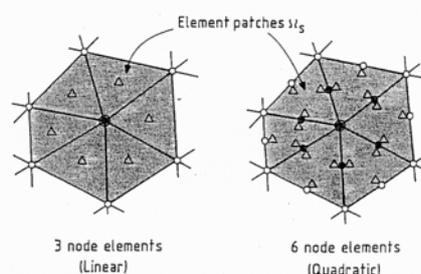
注意点:

1. 境界上節点の周りの処理



2. 中間節点の処理

重複して計算し, 後で平均化. (ポスト処理だからいい加減でもOK?)



Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 局所パッチ法 (SPR) のアルゴリズム ■ (p. 88—89)

高次要素の中間節点を除いた) 非境界節点の抽出

$\frac{\partial \mathbf{u}_i}{\partial \mathbf{x}}$ 等の初期化

```

do i=1,np
  dup(i)=0. ((中間節点で生じる) 複数の局所パッチでの重複カウンター)
  do k=1,4
    uxi(k,i)=0.d0
  end do
end do
do 100 k=1, np0(非境界節点数)
  i=jnt0(k)
  節点 i に連結した要素群: 局所パッチ
  局所パッチ内節点数 nppat 及び節点番号 jntpat の算出
  局所パッチ内非境界節点数 nppat0
    及び節点番号 jntpat0 の算出
  局所パッチ内要素数 nepat
    及び要素番号 metpat の算出
  [局所パッチ法計算のための準備]
do 200 j=1,nepat (局所パッチ要素数)
  局所パッチ要素群でのサンプリング点数の算出 nsample
  局所パッチ要素群でのサンプリング点の座標の算出 xsample
  局所パッチ要素群でのサンプリング点での  $u_x^h$  の算出 uxsample
200 continue
call scpatch(nsample,xsample,uxsample..)
  (局所パッチで  $\mathbf{a}=\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}^T$  の計算)

```

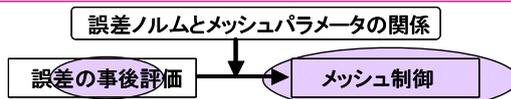
```

do k=1,nppat0 (局所パッチ内非境界節点数)
  i=jntpat0(k)
  nk=0
  do k1=1,2(変位自由度数)
    do k2=1,2(微分自由度数)
      nk=nk+1
       $\mathbf{a}=[a_1 \ a_2 \ a_3]^T$  より局所パッチ内での  $\frac{\partial \mathbf{u}_i^*}{\partial \mathbf{x}}$  ( $uxspr$ ) の導出
      uxi(nk,i)=uxi(nk,i)+ uxspr(nk)
      dup(i)=dup(i)+1.
    end do
  end do
end do
[重複評価の処理]
do i=1,np
  do k=1,4
    uxi(k,i)=uxi(k,i)/dup(i)
  end do
end do
100 continue

```

基底が線形なら global 座標に対する要素パッチの方向性の影響はないが、
 双一次の基底 $[1, x, y, xy]$ 以上では局所的な座標で基底を定義する等、工夫が
 必要. ここでは、簡単のため基底を global 座標に対して $[1, x, y]$ としている.

↑ キーポイント #4 に関連



■ トラクション法 ■ (p. 85—89) (subroutine tracsrt2 at adap.mesh2d)

QUAD4 に対して, QUAD8 を用いた要素レベルの局所問題

$$\mathbf{u}^* \in V_e : a_e(\mathbf{u}^*, \mathbf{u}^*) = \int_{\Omega_e} \mathbf{u}^* f + \int_{\Gamma_e} \mathbf{u}^* q d\Gamma \quad \forall \mathbf{u}^* \in V_e \quad (4.27)$$

$$V_e = \left\{ v = \sum_{i=1}^8 v_i N_i \mid v_i = u_i, i = 1, 4 \right\} \quad (4.28)$$

$$V_{e0} = \left\{ v = \sum_{i=1}^8 v_i N_i \mid v_i = 0, i = 1, 4 \right\} \quad (4.29)$$

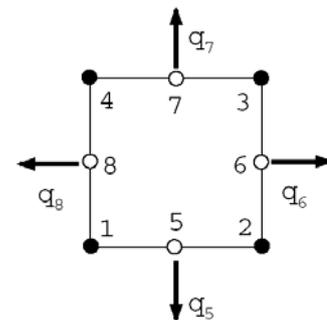


図 4.11 隣接する要素と表面力

高次微分関数

$$\mathbf{u}_{xx}^* \approx \sum_{i=1}^8 \mathbf{u}_i^* \frac{\partial^2 N_i}{\partial x^2}, \quad \mathbf{u}_{xy}^* \approx \sum_{i=1}^8 \mathbf{u}_i^* \frac{\partial^2 N_i}{\partial x \partial y} \quad (4.30)$$

$$\mathbf{u}_{yy}^* \approx \sum_{i=1}^8 \mathbf{u}_i^* \frac{\partial^2 N_i}{\partial y^2}$$

これらを誤差の定義式に代入して, 誤差ノルムを得る.

■ 1Dコードの動かし方:フロー ■

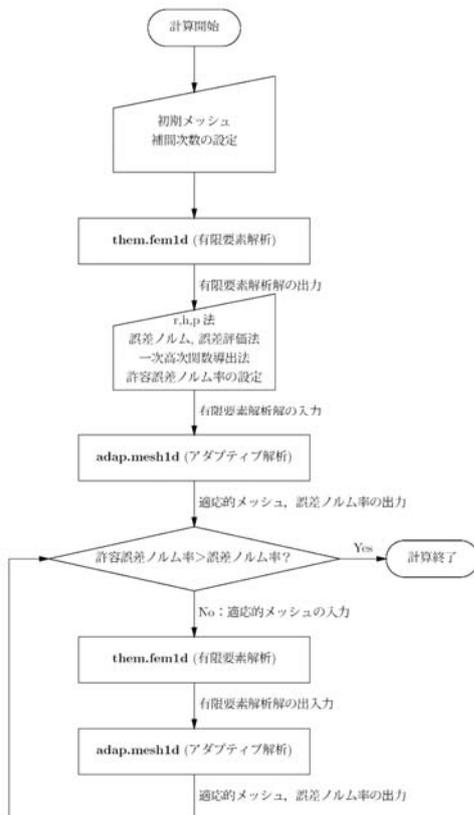


図7.1 1次元アダプティブFEAのフロー

- アダプティブ法の繰り返しが可能.
- check.exact.h等のコマンド内容はmake, commandのファイル内容を参照.
- check.exact.h等を参考にコマンドを作成し, 各種パラメータを振って体験可能.
- 書籍と同様な図がps fileで生成. GSviewで表示可能. (Win) (file-convertでpcxgrayとして*.pcxとsaveすると, 回転等, 加工可能)

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 1Dコードの動かし方:コマンド ■

```
% therm.fem1d
initial FE (0) or adaptive FE (1)?
pth element?
input the number of mesh division,nx
regular (0) or irregular (1) mesh?
.....
renumbered to adapline.dat! different from input data
(↑アダプティブメッシュの節点番号を座標値により新ソート.)
% adap.mesh1d
L2 (1), H1 (2), or H1semi (3) norm?
output for exact (1) or approximated (2)?
error estimate;ZZ(1), semi-exact intp(2), or intp(3)?
(↑f項ベース)
SPR(0), LSM(1), lumped LSM(2) or WAS(3)?
How many % of quality index? ←QIの指定
h-refinement(1),p-adap (2) or r-adap (3)?
% therm.fem1d
.....
% adap.mesh1d
% postgraph
```

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 添付コードでの誤差推定の体験法 ■ (CD-ROM内quick.manual)

【1次元定常熱伝導アダプティブFEM】

・サンプル実行法(要素制御含む)

% make check.zz.spr.h

(SPR 法+ZZ 法による要素細分化アダプティブ法(4.4.3 項及び5.2 節参照))

% make check.zz.spr.p

(SPR 法+ZZ 法による要素高次化アダプティブ法(4.4.3 項及び5.3 節参照))

% make check.zz.spr.r

(SPR 法+ZZ 法による節点移動アダプティブ法(4.4.3 項及び5.1 節参照))

% make check.zz.lsm.h

(LSM 法+ZZ 法による要素細分化アダプティブ法(2.5 節及び5.2 節参照))

% make check.zz.lsm.p

(LSM 法+ZZ 法による要素高次化アダプティブ法(2.5 節及び5.3 節参照))

% make check.zz.lsm.r

(LSM 法+ZZ 法による節点移動アダプティブ法(2.5 節及び5.1 節参照))

コマンド内容はmake, commandのファイル内容を参照. 類似コマンド作成により, 各種パラメータを振って体験可能

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 1次元誤差推定の要約 ■

1. 解のポスト処理 (LSM, lumped LSM, WAS, SPRの比較)

- ・均質・非均質メッシュ及び高次要素とも, SPRが一番良い性能を示す.
- ・非均質メッシュでは, LSM, lumped LSMの精度は悪化.
- ・最小二乗は単なるデータ処理・平滑化. ゆえに, 1次微分関数 u_x^h が階段状の1次要素に最も効果が大きく, 2次要素以上だと折れ線・曲線の平滑化のみ, 厳密解に近く効果はない.
- ・解のポスト処理単体を考えるとSPRがベスト.

2. 誤差評価

- ・誤差ノルムが微小な場合, コンピュータの打ち切り誤差が影響. メッシュパラメータ算出時の割り算により, 誤差が拡大する危険性.
- ・1次要素の場合, 解のポスト処理, 誤差ノルムの値は各手法で異なるものの, メッシュパラメータでの違いはなく, 違いがあっても安全側(メッシュが細かく, 次数が高い方向).
- ・境界近くに応力集中部等ある場合には解のポスト処理の境界近くの精度に注意する必要. (特に非均質メッシュ)

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 2Dコードの動かし方:フロー ■

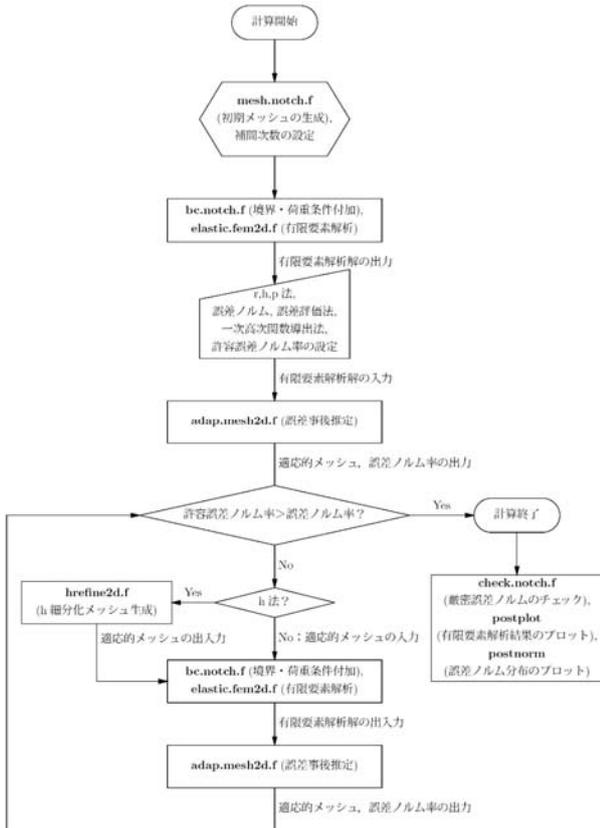


図7.3 1次元アダプティブFEAのフロー

・アダプティブ法の繰り返しには未対応.

・ZZ-spr等のコマンド内容はmake, commandのファイル内容を参照.

・ZZ-spr等を参考にコマンドを作成し, 各種パラメータを振って体験可能.

・書籍と同様な図がps fileで生成. Gsviewで表示可能. (Win)
(file-convertでpcxgrayとして*.pcxとsaveすると, 回転等, 加工可能)

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 2Dコード(誤差推定)の動かし方:コマンド ■

%mesh.notch

Input the # of division (dv should be even)

%bc.notch

%elastic.fem2d

%postplot

(FEモデル, 変位, 応力のプロット)

....

%check.notch

plot exact error norm (1), caliculated error norm(2),
exact FEA norm (3), or caliculated FEA norm(4)?

plot of mesh parameter for h-adap (1) & p-adap(2)

%postnorm

fill subdomains? (yes:1, no:0)

plot node no?(no=0):

plot domain no?(no=0):

need mesh sizes? (no=0:

(誤差ノルム分布のプロット)

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 添付コードでの誤差推定の体験法 ■ (CD-ROM内quick.manual)

【2次元定常熱伝導アダプティブFEM】

・サンプル実行法

```
% make basic-fea
% make exact-error-norm
% make interpolation-spr (←recommended)
% make interpolation-lsm
% make interpolation-lumped-lsm
% make interpolation-was
% make interpolation-tract (←recommended)
% make ZZ-spr (←recommended)
% make ZZ-lsm
% make ZZ-lumped-lsm
% make ZZ-was
% make ZZ-tract (←recommended)
```

(以上のコマンドにより, 図4. 27-4. 32のpostscript fileが生成される.)

コマンド内容はmake, commandのファイル内容を参照. 類似コマンド作成により, 各種パラメータを振って体験可能

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

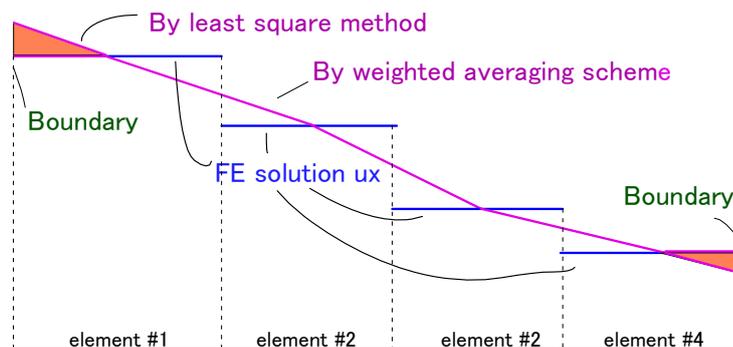
■ 2次元誤差推定の要約 ■

(LSM, lumped LSM, WAS, SPR, TRACの体験が可能)

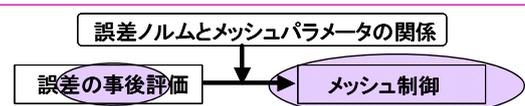
各手法での誤差ノルム値の違いは, 要素寸法・要素次数分布にさほど影響を与えない場合がある.

ただし, 一次元と同様,

- ・誤差ノルムが微小だと, メッシュパラメータ算出で誤差拡大の危険性
- ・境界近くに応力集中等ある場合, 解のポスト処理に注意.



↑ キーポイント#4に関連



Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ (英語) 論文の読み方

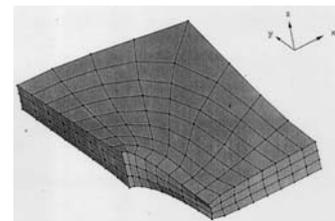
1. 最初にアブストラクト、結論、図、前書き後半を読む。この時点で役に立ちそうな論文かどうかわかる。
2. 次に解析事例を見る。これで理解できなければ、周囲の式を見る。
3. これでもわからなかったら、章立てを見て、必要部分から読み出す。
4. 熟読段階では、パラメータ依存性、恣意性、一般性について注意して読む。
5. 疑問点があれば、著者に e-mail で (電話は相手の時間に割り込むので失礼) 質問する。

お疲れ様でした。
次に二次元のメッシュ制御法 (r法, h法, p法) を説明します。

■ アダプティブ法におけるメッシュ制御 ■

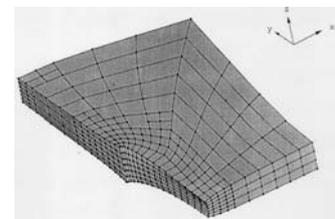
・r法

rはrelocation(再配置)の略。節点数・要素数は不変。解析効率良。要素が等しい誤差ノルムを持つように節点位置を移動。誤差を平均化するだけで、誤差ノルム総量はほとんど変化なし。要素形状が極端に歪む危険性あり、複雑な形状は不得意。



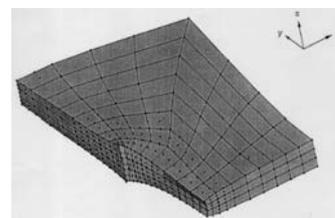
・h法

hはメッシュサイズ (height(高さ)) の略。節点数・要素数とも変化。各要素誤差ノルムが等しくなるように、誤差ノルム大の要素を細分化。要素細分化でモデル自由度は増加ゆえ、解析効率は低下。複雑な形状に対応可能であり、誤差ノルム総量を定量的に制御可。



・p法

pは形状関数のpolynomial(多項式)の略。要素数は不変。誤差ノルム大の要素形状関数を高次化する手法。要素の高次化でより多くの積分点が必要。初期のメッシュが粗くてよいためメッシュ分割が楽。ノルムの観点からは解の収束率は3つの方法の中で一番良い。非線形解析には不向き。



■ アダプティブ法におけるメッシュ制御 ■

3つの方法(実は一次元で体験済み)

- ・ **r 法**(節点移動法)
- ・ **h 法**(要素局所細分化法及び要素再生成法)
- ・ **p 法**(要素高次化法)

		長所	短所
r 法		計算効率が高い、非線形対応	境界での節点移動、要素歪、物理量の投影誤差
h 法	要素局所細分化法	複雑形状対応、非線形対応	全自動メッシャー不必要、物理量の投影誤差比較的小
	要素再生成法		要素寸法分布の細やかな制御可能
p 法		簡易なモデル化	初期メッシュ形状を継承、積分点増加、非線形不向き

表 5.1 r 法、h 法、p 法の長所と短所

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ r法(節点移動法) ■ (p. 111-115)

- ・ **Min-Max問題**

$$\text{Min}_{r\text{-adaptation}} \left\{ \text{Max}_{e=1,ne} E_e \right\}$$

- ・ **必要条件の一つ**

$$E_e = \text{constant for all } e, e = 1, ne$$

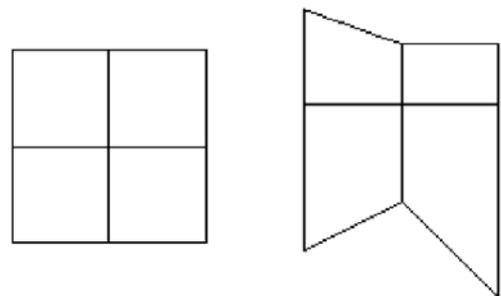
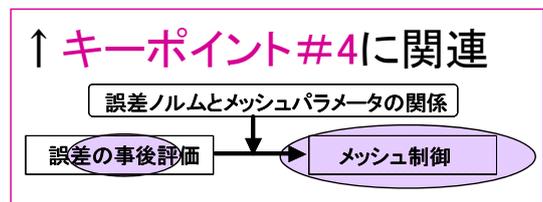


図 5.1 均質メッシュ (左) と不均質メッシュ (右)

- ・ **Winslowの平滑化法**

$$x_n = \frac{\sum_{j=1}^{j_{max}^n} x_j^n w_j}{\sum_{j=1}^{j_{max}^n} w_j}$$

x_n は新節点nの位置, x_{nj} は節点nに連なる要素jの重心位置, w_{nj} は要素jの重み



- ・ 節点 n の移動により, **要素が一つでも裏返った場合**, 節点nの移動は行わない.
- ・ **境界上の節点**は境界上を移動させるために**射影等の特別な処理**が必要.
(曲率を持つ形状では離散化以前のCADデータに基づいた節点移動が必要)
- ・ **Winslowの平滑化法**は**逐次処理**であり, 何回適用するか**の指標はない**.
- ・ **不均質メッシュ**では節点に連なる各要素の**誤差ノルムが等しくても節点移動する**.

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ r法の注意事項 ■

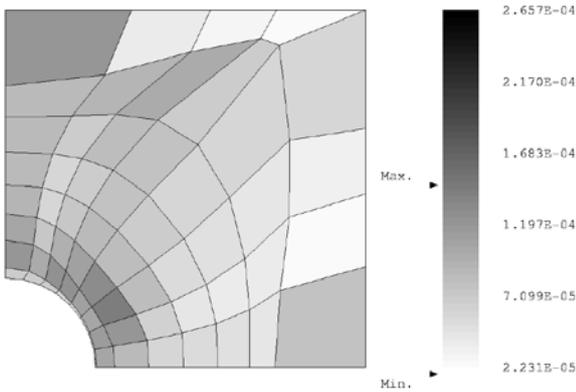


図 5.3 r メッシュの誤差ノルム分布 (重み=誤差ノルム; 平滑化=10 回)

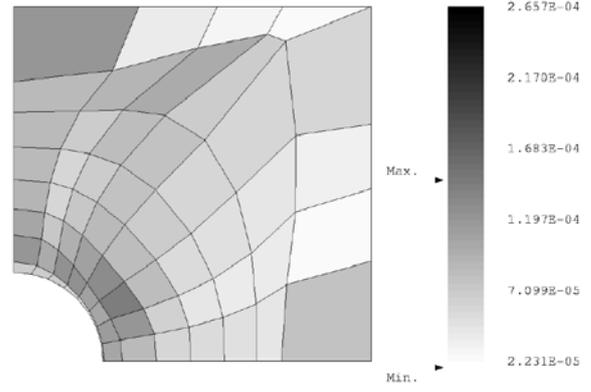


図 5.4 r メッシュの誤差ノルム分布 (重み=誤差ノルム; 平滑化=100 回)

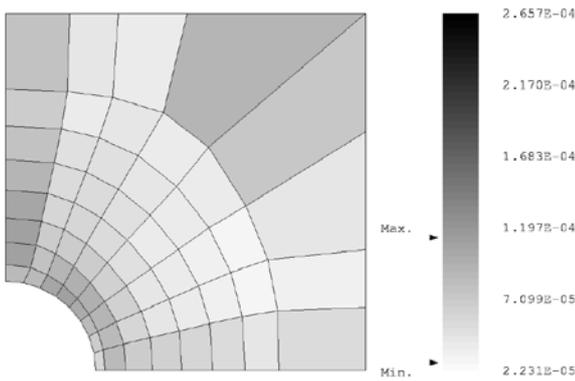


図 5.5 r メッシュの誤差ノルム分布 (重み=1/h; 平滑化=10 回)

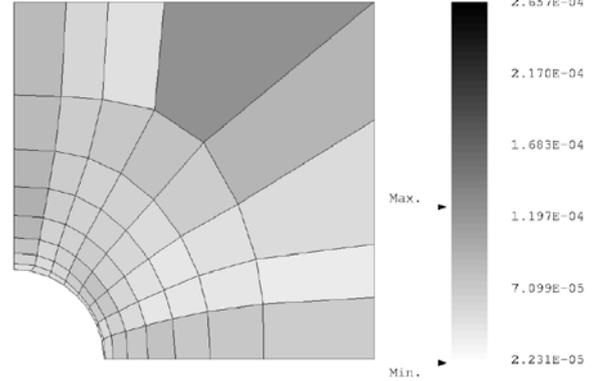


図 5.6 r メッシュの誤差ノルム分布 (重み=1/h; 平滑化=100 回)

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

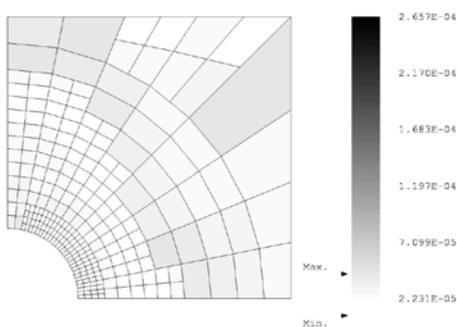
■ h法 ■ (p. 116–125)

・局所細分化法

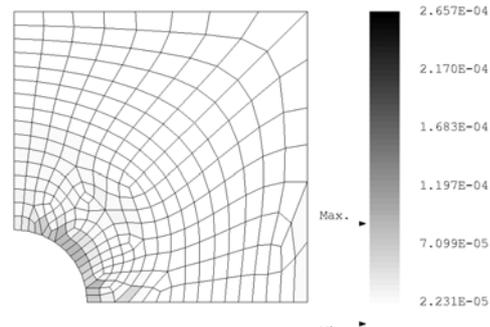
初期要素を要素ごとに局所的に細分化していく方法
→ 節点拘束に注意が必要.

・アダプティブ・リメッシング法

誤差ノルムから求めた要素制御寸法分布より, 節点分布を自動生成し, 要素自動生成を経てアダプティブ・メッシュを得る方法
→ 誤差推定時の要素毎に算出された離散的要素寸法分布をどのように連続的なメッシュに変換するか (節点分布とメッシュ生成) が問題.



局所細分化法



アダプティブ・リメッシング法

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 従来細分化法の問題点 ■(p.116-119)

element 3	element 4
mesh size =h/2	mesh size =h
element 1	element 2
mesh size =h/3	mesh size =h/5

図5.7 メッシュ分布

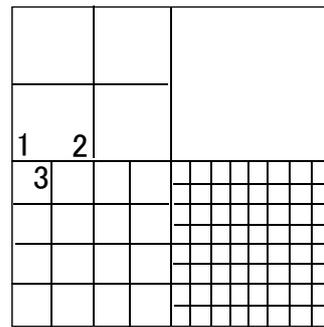
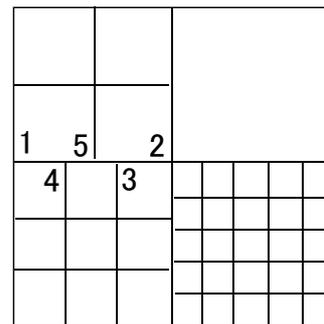


図5.8 局所細分化メッシュ



要素寸法に基づく局所細分化メッシュ

Copyright © 2004, Akira Tezuka , Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 節点拘束法の問題点 ■

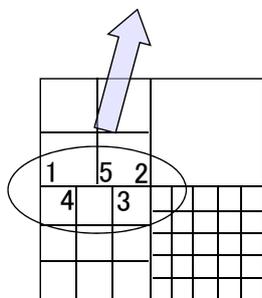
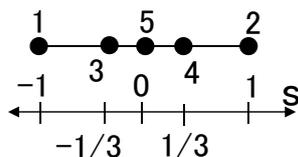


図5.9要素寸法に基づく局所細分化メッシュ

・線形関数による節点拘束

$$\begin{cases} u_1 + 2u_3 = 3u_4 \\ u_4 + u_5 = 2u_3 \\ 2u_3 + u_2 = 3u_5 \end{cases} \Rightarrow \begin{cases} 2u_1 + u_2 = 3u_4 \\ u_1 + u_2 = 2u_3 \\ u_1 + 2u_2 = 3u_5 \end{cases}$$

→非連続線は線形挙動ゆえ, 堅い変形

・高次関数による節点拘束

$$\begin{cases} u^{\text{line1-3-2}} = (u_1 + u_2 - 2u_3)s^2 - (u_1 - u_2 - 2u_3)s \\ u^{\text{line1-4-5-2}} = 9(-u_1 + u_2 + 3u_4 - 3u_5)s^3 + 9(u_1 + u_2 - u_4 - u_5)s^2 + (u_1 - u_2 - 27u_4 + 27u_5)s - u_1 - u_2 + 9u_4 + 9u_5 \end{cases}$$

→要素境界でC0連続性の問題

Copyright © 2004, Akira Tezuka , Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 節点非拘束法の問題点 ■

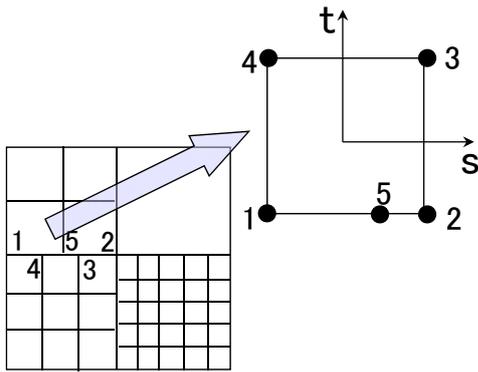
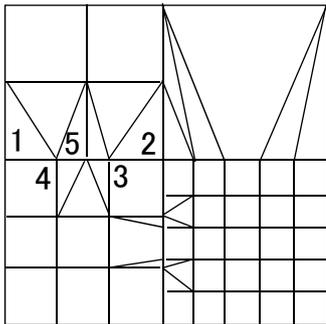


図5.10 Serendipity要素



■ 任意のSerendipity要素

$$n1 = -(3s-1)*(s-1)*(t-1)/16$$

$$n2 = -(3s-1)*(s+1)*(t-1)/8$$

$$n3 = ((s+1)*(t+1))/4$$

$$n4 = -(s-1)*(t+1)/4$$

$$n5 = (9*(s+1)*(s-1)*(t-1))/16$$

→要素境界でC0連続性の問題

■ TRI/QUAD混在要素

→混在要素, 歪んだ要素による悪影響

■ 節点拘束法 ■ (p. 120-122)

$$Cu = 0 \quad (\text{拘束行列式})$$

■ 直接消去法

$$C_1 u_1 + C_2 u_2 = 0$$

$$u_1 = -C_1^{-1} C_2 u_2$$

u_1 : 従属節点, u_2 : 独立節点 (C_1^{-1} 部分が律速になる)

■ ラグランジュ未定常数法

$$J_L(u) = \frac{1}{2} u^t K u - u^t f + \lambda C u \quad \begin{bmatrix} K & C^T \\ C & 0 \end{bmatrix} \begin{Bmatrix} u \\ \lambda \end{Bmatrix} = \begin{Bmatrix} f \\ a \end{Bmatrix}$$

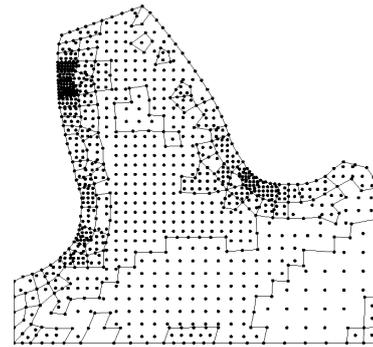
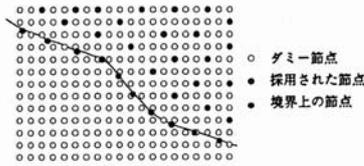
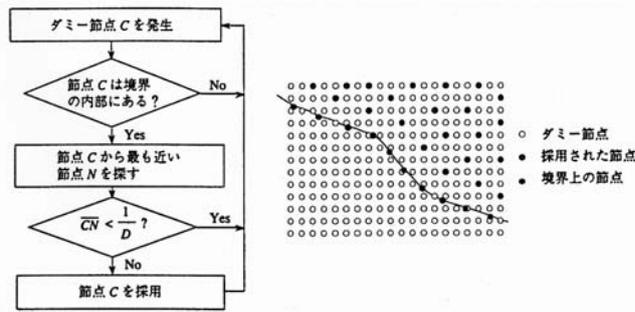
(並列解析では領域分割された個々の行列が特異にならないよう注意が必要)

■ ペナルティ法

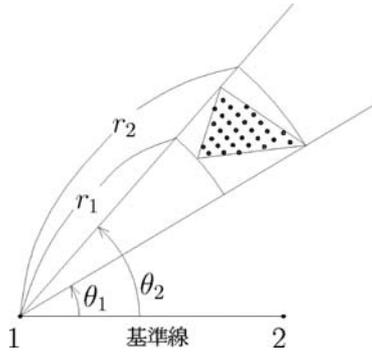
$$(K + pC^T C) u = f$$

K, C, u, f 及び p は全体系の剛性行列, 拘束行列, 変位ベクトル, 荷重項及びペナルティ係数 = 剛性行列項の最大値 \times 全自由度数 $\times 10^n$
(経験的に $n=4.5$ 程度に置くのが無難)

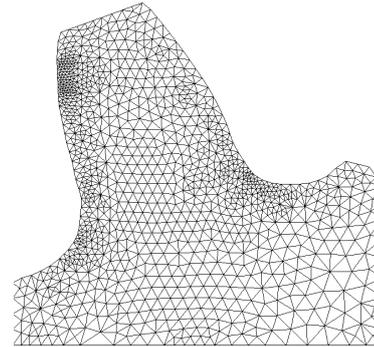
■ アダプティブメッシュ生成 ■ (p. 123-127)



節点分布密度コントロールに基づく方法



放射基準法



部分領域自動生成法

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ p法 ■

▪ quadratic LINE3

$$G_1(s) = (s - s_1)(s - s_2)$$

$$N'_1(s) \leftarrow \frac{G_1(s)}{(s - s_1)}$$

$$N'_2(s) \leftarrow \frac{G_1(s)}{(s - s_2)}$$

$$N_1(s) \leftarrow \frac{N'_1(s)}{N'_1(s_1)}$$

$$N_2(s) \leftarrow \frac{N'_2(s)}{N'_2(s_2)}$$

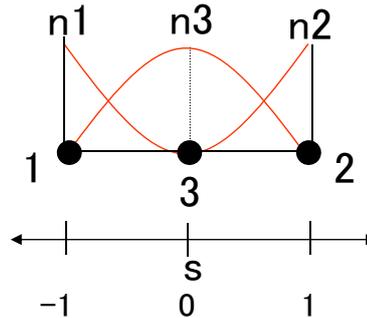
$$G_2(s) = (s - s_1)(s - s_2)(s - s_3)$$

$$N'_3(s) \leftarrow \frac{G_2(s)}{(s - s_3)}$$

$$N_3(s) \leftarrow \frac{N'_3(s)}{N'_3(s_3)}$$

$$N_1(s) \leftarrow N_1(s) - N_1(s_3)N_3(s_3)$$

$$N_2(s) \leftarrow N_2(s) - N_2(s_3)N_3(s_3)$$



二次元, 三次元でも同様
一次微分関数の導出は少々複雑.

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 2Dコード(r法)の動かし方:コマンド ■

```
% mesh.notch
Input the # of division (dv should be even)
% bc.notch
% elastic.fem2d
% adap.mesh2d ← energy error normのみサポート
sed (0), interpolation error (1) or zz error (2)?
by patch(1), lsm(2), lumped-lsm(21), was(3) or traction(4) method?
input the percentage of error you can admit←QIの指定
error wrt norm is +*+*+* %→現状のQIの計算値
r-method (1) or p-method (2) ?
which value as a weight?: error measure=1, inv. of mesh size=2
how many iterations for nodal relocation?
1-np (1), np-1 (2) or both (3)?
% bc.notch
% elastic.fem2d
% postplot
% postnorm
fill subdomains? (yes:1, no:0)
plot node no?(no=0):
plot domain no?(no=0):
need mesh sizes? (no=0):
```

} 図5.3-5.6のような
パラメータスタディ

Copyright © 2004, Akira Tezuka , Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 2Dコード(h法)の動かし方:コマンド ■

```
% mesh.notch
Input the # of division (dv should be even)
% bc.notch
% elastic.fem2d
% adap.mesh2d ← energy error normのみサポート
sed (0), interpolation error (1) or zz error (2)?
by patch(1), lsm(2), lumped-lsm(21), was(3) or traction(4) method?
input the percentage of error you can admit←QIの指定
error wrt norm is +*+*+* %→現状のQIの計算値
r-method (1) or p-method (2) ?
% hrefine2d
% bc.notch
% elastic.fem2d
% postplot
% postnorm
fill subdomains? (yes:1, no:0)
plot node no?(no=0):
plot domain no?(no=0):
need mesh sizes? (no=0):
```

Copyright © 2004, Akira Tezuka , Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 2Dコード(p法)の動かし方:コマンド ■

```
% mesh.notch
```

Input the # of division (dv should be even)

```
% bc.notch
```

```
% elastic.fem2d
```

```
% adap.mesh2d
```

sed (0), interpolation error (1) or zz error (2)?

by patch(1), lsm(2), lumped-lsm(21), was(3) or traction(4) method?

input the percentage of error you can admit ← QIの指定

error wrt norm is +*+*+* % → 現状のQIの計算値

r-method (1) or p-method (2) ?

input the degree of polinomial for uniform mesh (0. for skip)

tensor product element (1) or serendipity element (2) ?

```
% bc.notch
```

```
% elastic.fem2d
```

```
% postplot
```

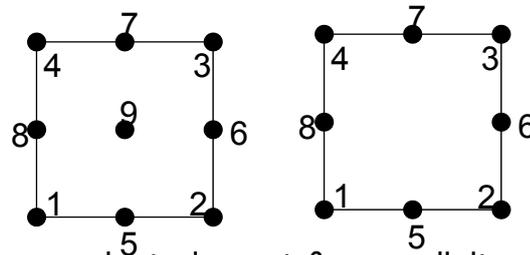
```
% postnorm
```

fill subdomains? (yes:1, no:0)

plot node no?(no=0):

plot domain no?(no=0):

need mesh sizes? (no=0):



tensor product element & serendipity element

Copyright © 2004, Akira Tezuka , Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 添付コードでのr, h, p法の体験 ■ (CD-ROM内quick.manual)

【2次元定常熱伝導アダプティブFEM】

・サンプル実行法

```
% make exact-error-norm
```

(基本メッシュFEAの厳密誤差ノルム分布)

```
% make r-method
```

(r法)

(図5. 3-5. 6のパラメータスタディが可能)

```
% make h-method
```

(h法(局所細分化法))

(註: アダプティブリメッシングはサポートしていない.)

```
% make p-method
```

(p法)

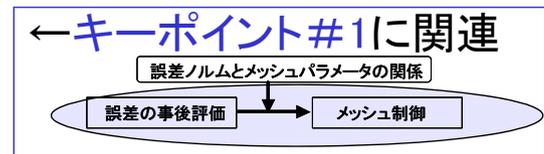
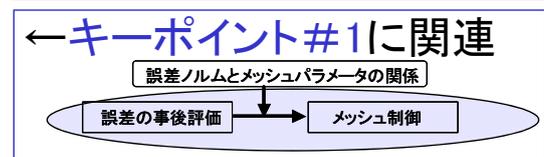
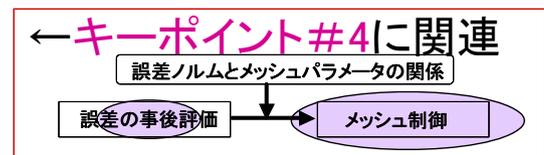
コマンド内容はmake, commandのファイル内容を参照. 類似コマンド作成により, 各種パラメータを振って体験可能

Copyright © 2004, Akira Tezuka , Continuum Modeling Research Group, Research Institute for Computational Science, AIST

お疲れ様でした. 次は応用編です.

■ アダプティブ法の盲点 ■

1. 近似関数 w の問題点
近似関数 w の導出法
2. メッシュ方向性は考慮外
誤差ノルムは方向性を持たない量
3. メッシュ間の投影誤差の影響
流体系解析では重要
4. 良い意味でも悪い意味でも拡大鏡
解析モデルの数学的矛盾を露呈する
5. P 法か h 法か?
ノルムという指標に注意.

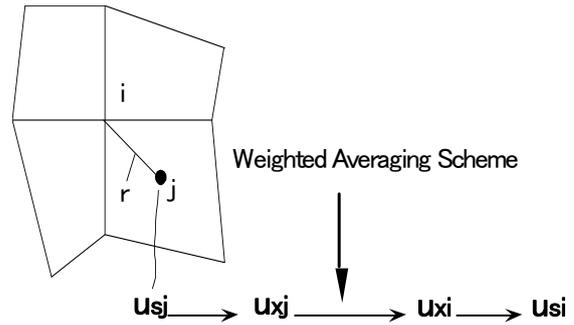
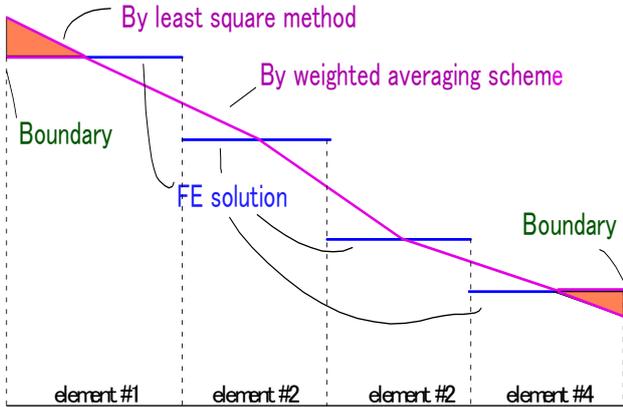
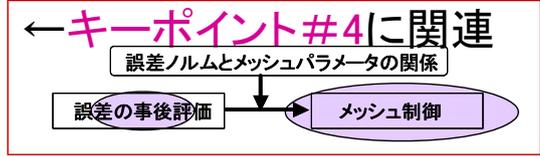


■ 近似関数wの問題点 ■

1. 境界近くの値

「重み付き平均化法」では低めの値

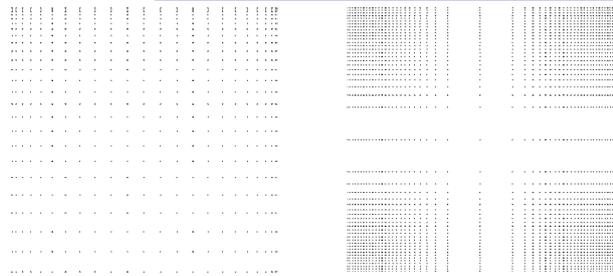
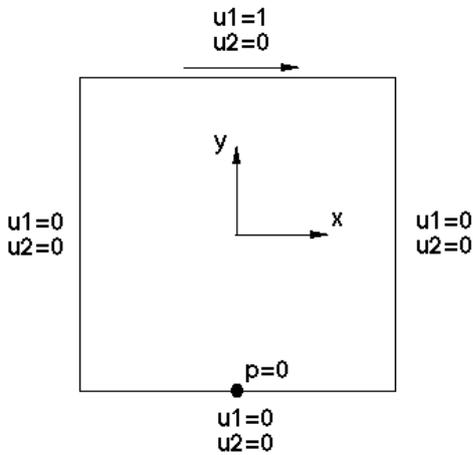
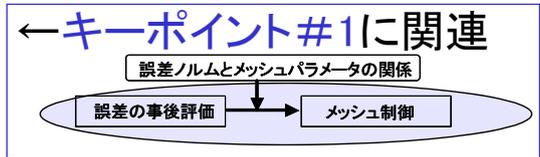
2. スムージングのみでFEMの性質は用いていない方法等



Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

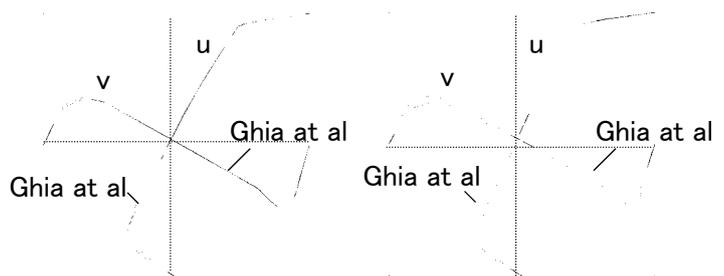
■ メッシュの方向性が重要な問題 ■

例: Driven cavity問題
(非圧縮流れ)



24*24 mesh

40*40 mesh



Space-Time + 安定化GLS FEM

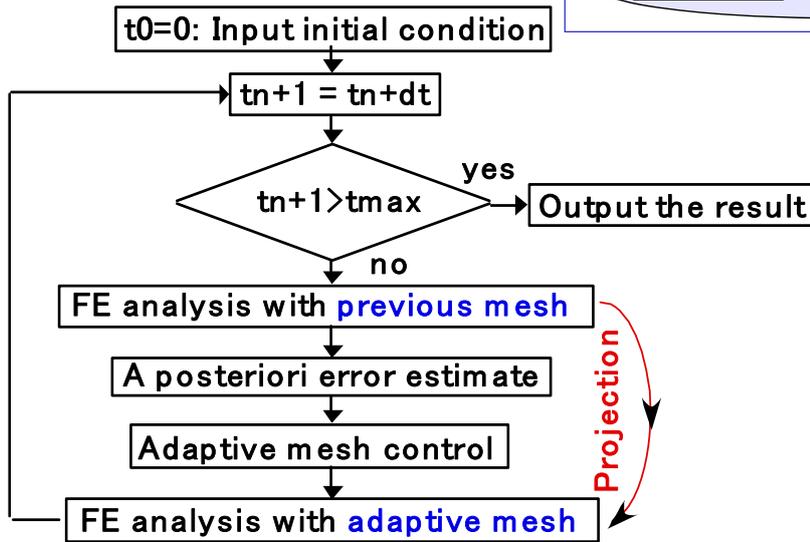
Re=1000, dt=0.02, 2000 time steps

r法でこのようなメッシュを得るのは困難

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

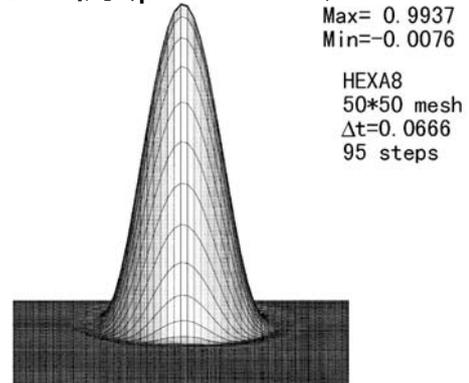
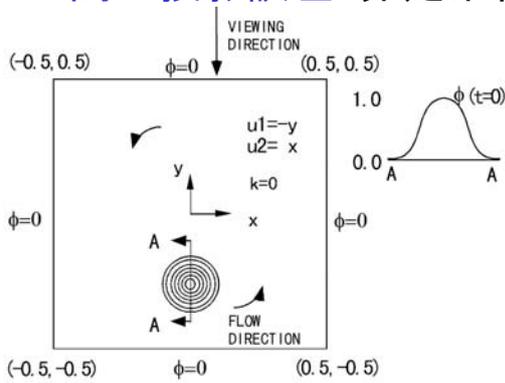
■メッシュ間の投影誤差■ (p.136-144)

←キーポイント#1に関連



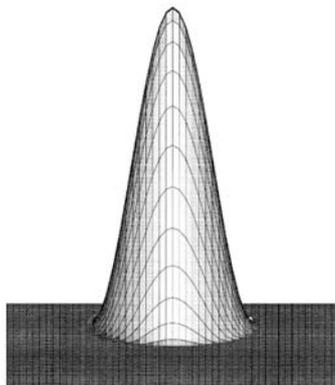
異なるメッシュ間での物理量の射影誤差が同ステップ内の解析誤差減少分を卓越
 →r法, アダプティブリメッシングの問題点

メッシュ間の投影誤差: 非定常移流問題での例 (p.141-144)

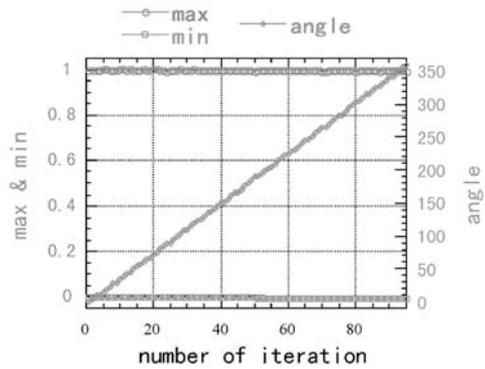


回転するコーン型濃度分布(ベンチマーク)

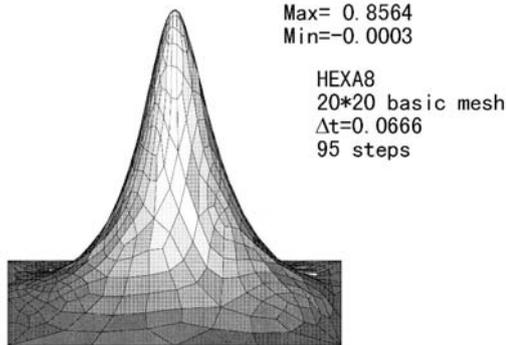
一回転後の濃度phi (50x50固定メッシュ)



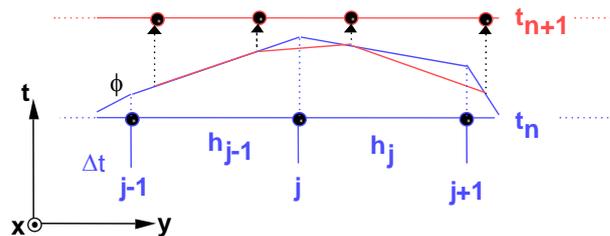
濃度phiの初期条件



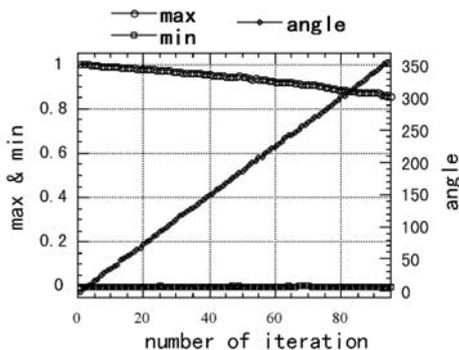
phiの最大値、最小値の変遷



一回転後の濃度 ϕ (アダプティブメッシュ)

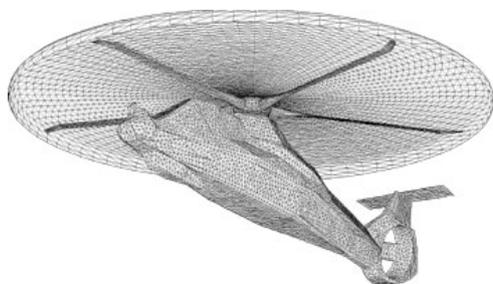


異なるメッシュ間の投影誤差



ϕ の最大値、最小値の変遷

A. Tezuka, Adaptive space-time GLS FEM with direct projection for 2D advection-diffusion problem, International Journal of Computational Fluid Dynamics, 12, pp.241-248, 1999.12.



ヘリコプターの解析 <http://www.mems.rice.edu/TAFSM/HLIGHT/>

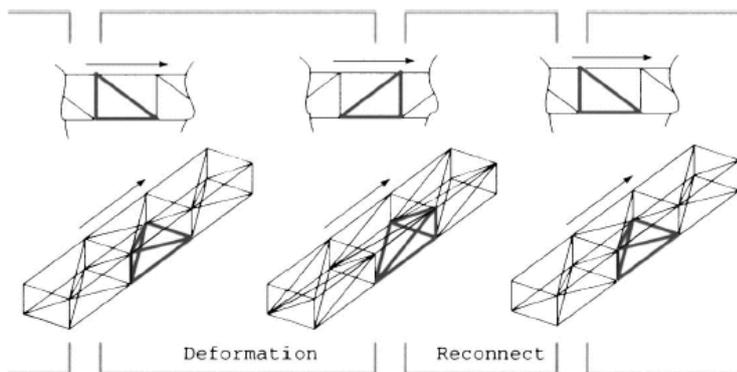
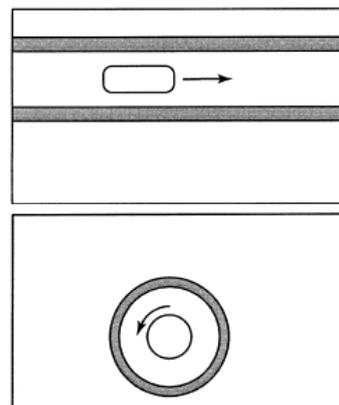


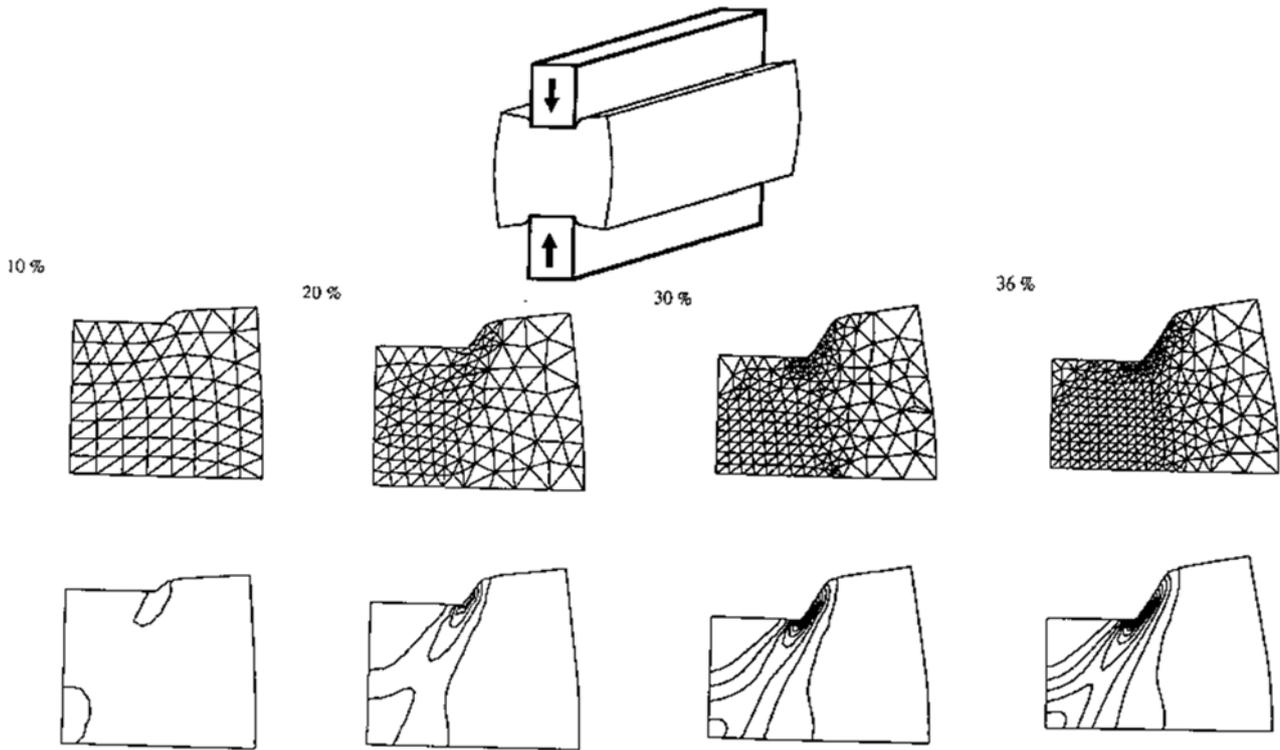
Fig. 1. Shear-slip layer concept in 2D (top) and 3D (bottom): the sequence of deformation followed by reconnecting.



タイムステップ毎のメッシュ切り直しスキーム

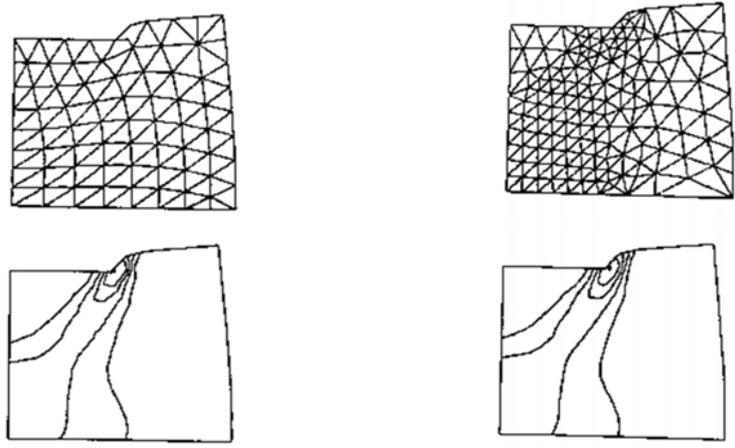
M. Behr and T. Tezduyar, "Shear-Slip Mesh Update in 3D Computation of Complex Flow Problems with Rotating Mechanical Components", Computer Methods in Applied Mechanics and Engineering, 190(2001) pp.3189-3200

■ 投影誤差: 固体力学では大丈夫 ■



アダプティブリメッシングを用いた二次元剛塑性バルクメタルフォーミング

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST



基本メッシュと等価歪み
(アダプティブリメッシング前)

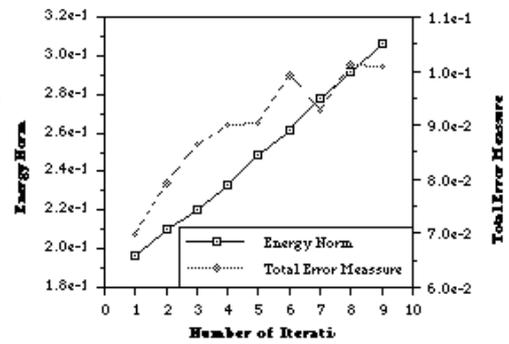
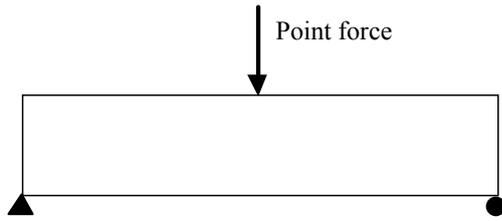
基本メッシュと等価歪み
(アダプティブリメッシング後)

固体力学の場合: N, Yukawa, N. Kikuchi, A. Tezuka, An Adaptive Remeshing Method for Analysis of Metal Forming Processes, Advanced Technology of Plasticity 1990, Vol. 4, (1990-7), pp1719-1728
非定常挙動がおとなしいため, 異なるメッシュ間の投影誤差は微小

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 良い意味でも悪い意味でも拡大鏡 ■

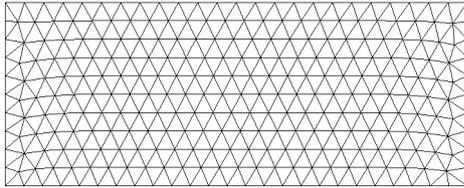
Babuskaのパラドックス(点荷重/点支持)



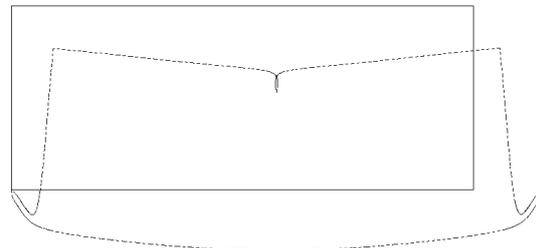
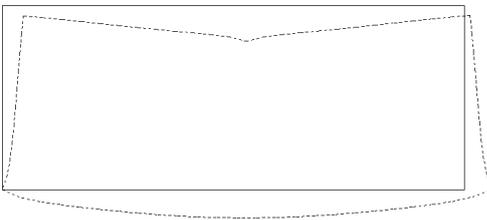
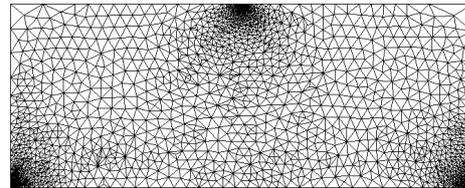
点荷重はソボレフ空間 H^1 でうまく定義できない

誤差ノルムが発散！

285 nodes 499 elements



3563 nodes 6819 elements

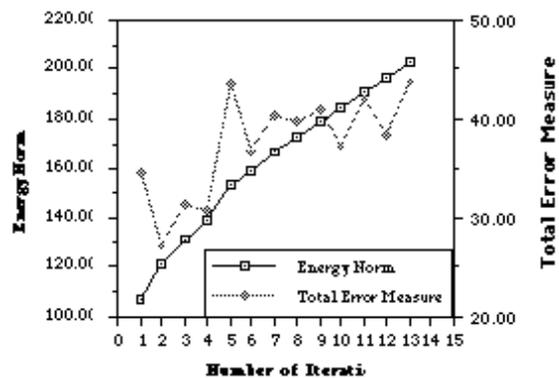
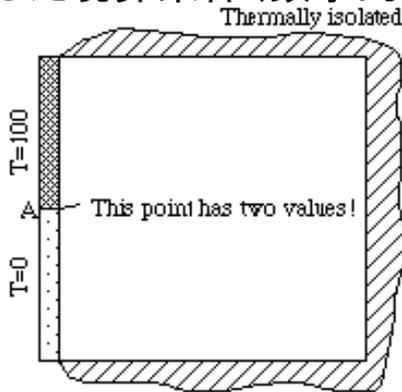


奇妙な変位！

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

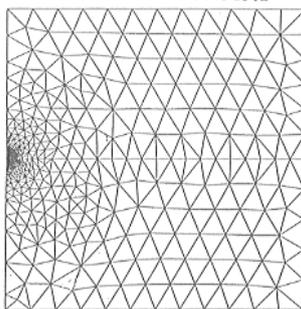
■ 良い意味でも悪い意味でも拡大鏡 ■ (続き)

重複した境界条件(数学的に誤)

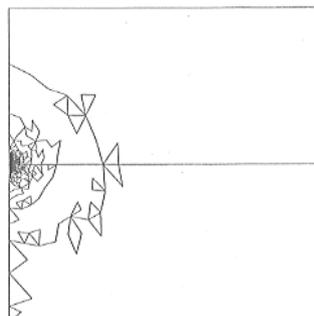


誤差ノルムが発散！

Elements 2473 Nodes 1302

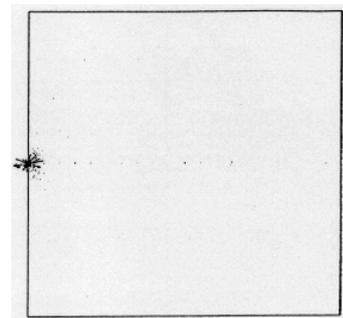


of Subdomains 1573



12th Adapted Meshes

Subdomains on 12th Stage

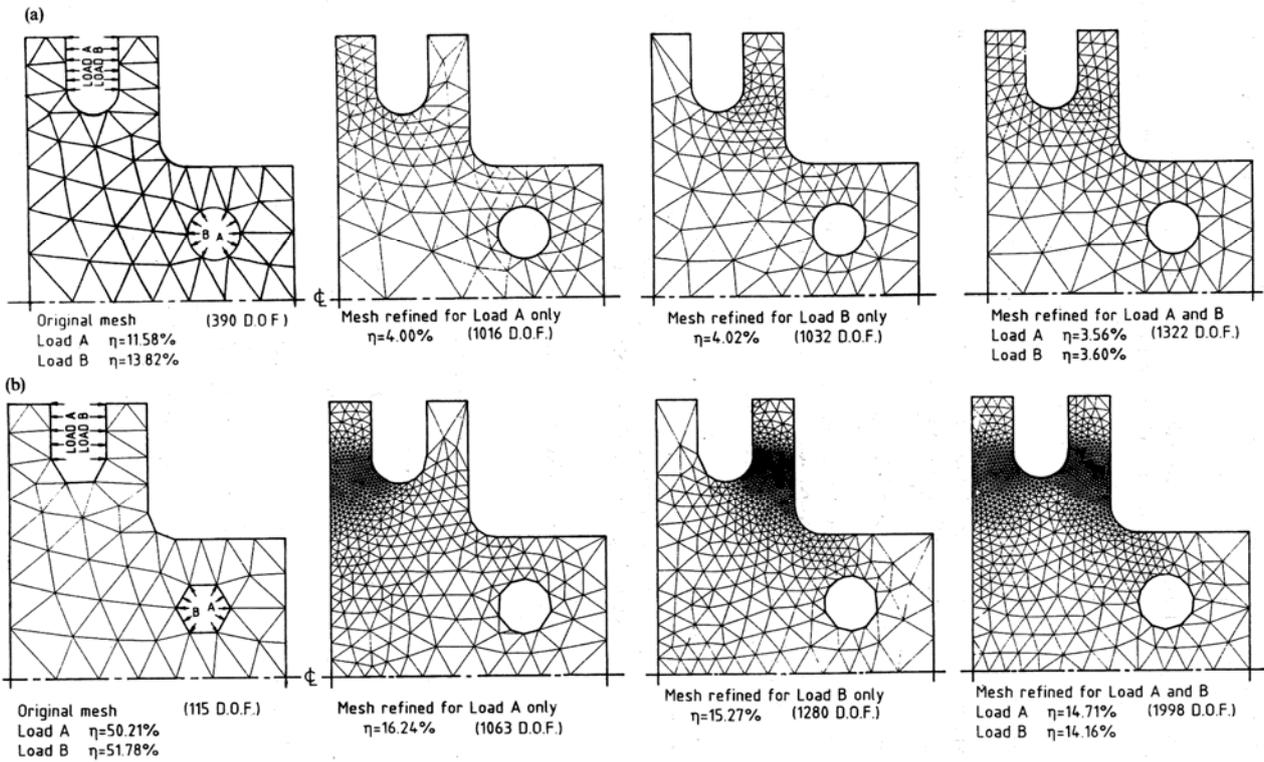


熱流速の分布

手塚明, 菊池昇, 機論A, 58巻555号, (1992), pp2199-2205

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ Zienkiewiczの論文(QIの値) ■

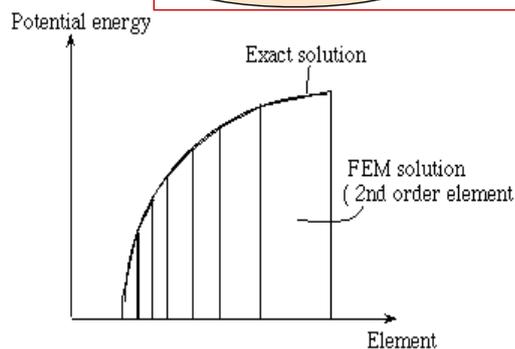
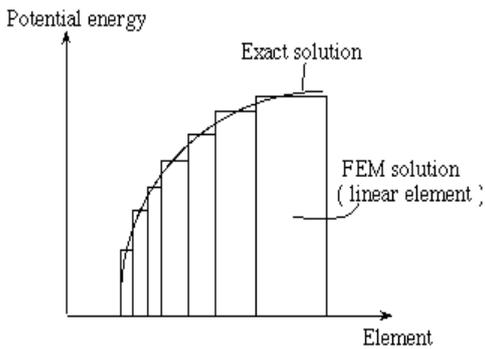


数値に騙されない事！(ノルムで見ているからこうなる)

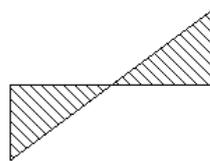
Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ p法かh法か？ ■

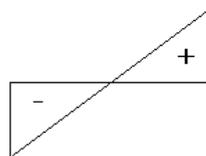
・誤差ノルム率とエネルギーノルム収束率



誤差ノルムの物理的意味



誤差ノルム



エネルギーノルムでの差

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 4つのキーポイント:他の分野は? ■

#1. 違う性質の2つの組み合わせ

ノルム 次元依存性なし

メッシュ 次元依存性あり

→安定化FEMの代表寸法 h の定義!

#2. 誤差ノルムでの評価とは?

ノルムの性質

→高次要素神話?

#3. 基礎理論を逸脱する形で理論実用化へ

→積分点のマジック?

#4. パラメータ依存

→メッシュを取り去ったメッシュレスにパラメータ依存はないか?

未だに、アダプティブ法はやはり必要! キチンと考慮しないと火傷.

X-FEM:関数定義とベースメッシュの組み合わせ.

ベースメッシュ担当の精度は h 法で面倒を見るしかない!

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

お疲れ様でした.
休憩10分です.
次は量子MDへの応用です.

Short Break

■英語でのプレゼンテーションのポイント

1. 所詮、日本人の英語は日本人の英語ゆえ、US native と異なり、スライドで80%の情報供給（ひとめで内容が把握できるものを）、残りの20%をプレゼンで補強が原則。
2. スライドの限られた紙面で意味を伝えるためには、重要部分のみの効果的な抽出能力が重要。例えば、日本語で100%話すところを、英語の論理に基づき50%の量に削減した情報（言葉、図）で、聴衆に内容を把握させる必要。
3. 発表原稿は作成しても良いが、当日はそれには捕らわれず、なるべくアドリブで発表する。発表原稿丸読み、スライドが発表原稿というのは聴衆に失礼。
4. 大局的な意味が理解してもらえるように、あれもこれも入れ込まない。一番見せたいスライドに向かっていくプレゼンの流れを感じる事。一つの事が主張できれば成功。
5. 大きな声でプレゼンを。最初の数分で引き付ける。中国英語、韓国英語、インド英語、日本英語等、あって良く、発音に神経質なあまり、消極的になってしまつては本末転倒。

■ 1Dコードの動かし方:フロー ■

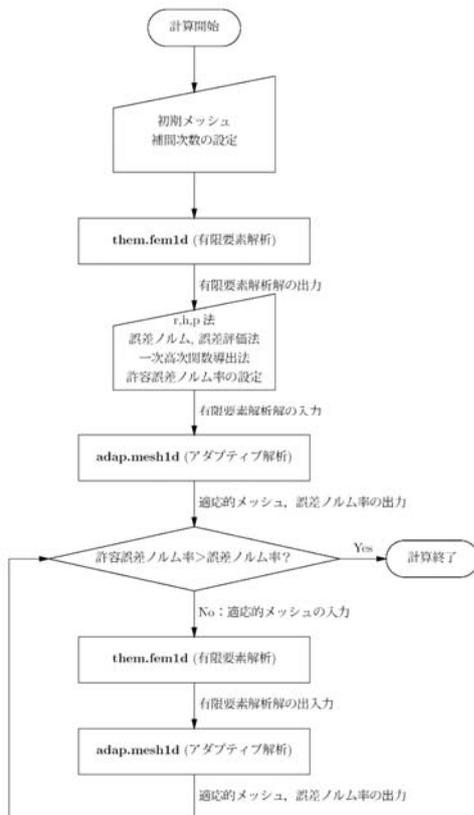


図7.1 1次元アダプティブFEAのフロー

- アダプティブ法の繰り返しが可能.
- check.exact.h等のコマンド内容はmake, commandのファイル内容を参照.
- check.exact.h等を参考にコマンドを作成し, 各種パラメータを振って体験可能.
- 書籍と同様な図がps fileで生成. GSviewで表示可能. (Win) (file-convertでpcxgrayとして*.pcxとsaveすると, 回転等, 加工可能)

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 1Dコードの動かし方:コマンド ■

```
% therm.fem1d
initial FE (0) or adaptive FE (1)?
pth element?
input the number of mesh division,nx
regular (0) or irregular (1) mesh?
.....
renumbered to adapline.dat! different from input data
(↑アダプティブメッシュの節点番号を座標値により新ソート.)
% adap.mesh1d
L2 (1), H1 (2), or H1semi (3) norm?
output for exact (1) or approximated (2)?
error estimate;ZZ(1), semi-exact intp(2), or intp(3)?
(↑f項ベース)
SPR(0), LSM(1), lumped LSM(2) or WAS(3)?
How many % of quality index? ←QIの指定
h-refinement(1),p-adap (2) or r-adap (3)?
% therm.fem1d
.....
% adap.mesh1d
% postgraph
```

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ therm.fem1d ■ (1次元定常熱伝導有限要素解析コード)

入力データ : adap.mesh1d.out (アダプティブ解析用. adap.mesh1dからの出力.)
出力データ : adap.mesh1d.dat (アダプティブ解析用. adap.mesh1dへの出力)
サブルーチン:
—model : 初期, r, h, pメッシュの有限要素モデルの生成
—pkgene1 : 局所“剛性”行列構築
—fkgene1 : 局所“荷重”ベクター構築
[1D共通サブルーチン]
—dshape1 : 1次元任意階層型高次要素のためのFE近似関数・微分関数の構築
—arbshape1 : 1次元階層型高次要素導出
—gshape1 : 1次元基本高次関数の構築
[1D/2D共通サブルーチン]
—skyheight : スカイライン高さ情報の構築
—assemb : 局所“剛性”行列, 局所“荷重”ベクターの全体系への組み込み
—skyline : スカイライン・マトリックスソルバー
—ludcmp : 密行列解法(前進消去)
—lubksb : 密行列解法(後退代入)
—mparam : 誤差ノルムからメッシュパラメータへの変換(4.5節に対応)
—gaussint : ガウスの積分点情報
—heapsort : ヒープソート(数値による並び替え)

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ adap.mesh1d, postgraph ■ (1次元定常熱伝導有限要素解析コード)

・adap.mesh1d (1次元定常熱伝導有限要素解析の誤差事後推定コード)
入力データ : adap.mesh1d.dat (FEA解データ. therm.fem1dからの出力.)
出力データ : adap.mesh1d.out (アダプティブFEA用. therm.fem1dへの出力.)
サブルーチン:
—fenorm : 有限要素解ノルムの算出
—lsmsrt1 : LSM, 集中化LSMに基づく1次高次関数導出(4.4.1項に対応)
—wassrt1 : 重み付き平均化法に基づく1次高次関数導出(4.4.2項に対応)
—patsrt1 : 局所パッチ法による1次高次関数導出(4.4.3項に対応)
—intnorm1 : 補間誤差ノルム算出(4.3.2項に対応)
—zzenorm1 : ZZ法による誤差ノルム算出(4.3.3項に対応)
[1D/2D共通サブルーチン]
—mparam : 誤差ノルムからメッシュパラメータへの変換(4.5節に対応)
—heapsort : ヒープソート(数値による並び替え)
・postgraph* (1次高次関数・誤差ノルムプロットコード. postscriptファイルに出力)
入力データ : plot.ux.dat, plot.enorm.dat (アダプティブ用. adap.mesh1dの出力)
出力データ : fig1.ps (一次高次関数uxのプロット: postscript file)
: fig2.ps (誤差ノルムEのプロット : postscript file)

*このコードは日産自動車(株)総合研究所 鳥垣俊和氏がミシガン大学工学部博士課程在籍中に作成されたgeposc.fを基に手塚が改変したものである。

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 注意事項 ■ (CD-ROM内 (quick.manual))

- ・therm.fem1d 及び adap.mesh1d 間で, h 法, p 法を繰り返し行う場合, 結果のプロットの関係上, therm.fem1d の最後の部分で節点番号の付け替えをしているので注意が必要である.
- ・高次要素の要素内節点番号の要素コネクティビティでの並び方は要素端点が先, 内部が後である. (4.3.2 項の高次要素の節点番号順を参照の事)
- ・therm.fem1d は2.3 節で説明したソース項(熱発生率)が指数関数の熱伝導方程式に基づくものである. このソース項 $e^{-x}/(1 - 10^{-10}e^{-x})$ は高次微分関数が同一であるという性質を持っており, adap.mesh1d の subroutine interr のインプリでは一部その性質を利用している.
- ・厳密解の関数は指数関数であり, ガウスの数値積分は指数関数の積分には厳密には対応していないが, 工学的には十分という事で, これに関わる数値積分は30 点積分としている. 有限要素関数に関する数値積分は, 要素次数に対応した積分点数としている.
- ・4. 5 節でも言及しているが, solution recovery に用いられるLSM 法(最小二乗法)はFEM で求められた変位の一次微分関数を一次高次の関数でスムージングをしているだけである. ゆえに, 変位の一次微分関数が階段状になる一次要素を用いたFEM に一番効果がある.

Copyright © 2004, Akira Tezuka , Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 2Dコードの動かし方:フロー ■

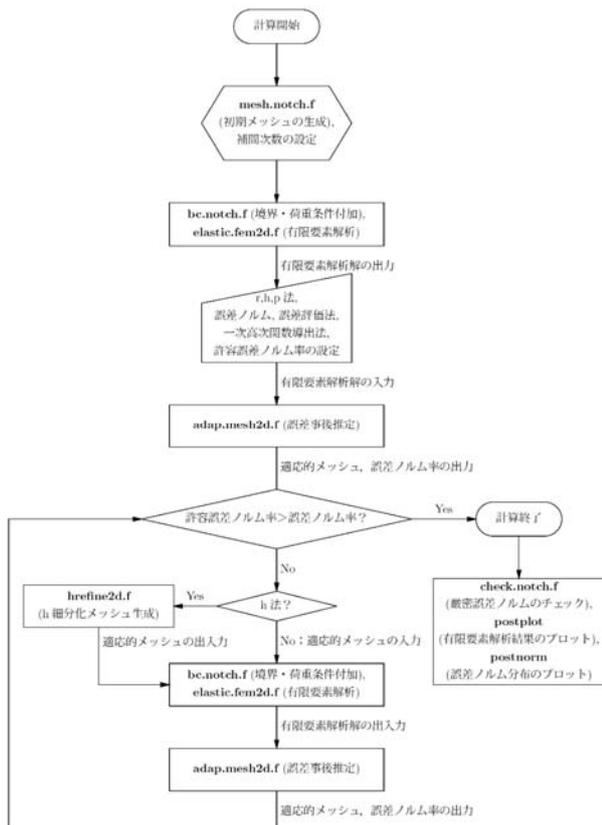


図7.3 2次元アダプティブFEAのフロー

- ・アダプティブ法の繰り返しには未対応.
- ・ZZ-spr等のコマンド内容はmake, commandのファイル内容を参照.
- ・ZZ-spr等を参考にコマンドを作成し, 各種パラメータを振って体験可能.
- ・書籍と同様な図がps fileで生成. Gsviewで表示可能. (Win) (file-convertでpcxgrayとして*.pcxとsaveすると, 回転等, 加工可能)

Copyright © 2004, Akira Tezuka , Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 2Dコード(誤差推定)の動かし方:コマンド ■

```
%mesh.notch
Input the # of division (dv should be even)
%bc.notch
%elastic.fem2d
%postplot
(FEモデル, 変位, 応力のプロット)
....
%check.notch
plot exact error norm (1), caliculated error norm(2),
    exact FEA norm (3), or caliculated FEA norm(4)?
plot of mesh parameter for h-adap (1) & p-adap(2)
%postnorm
fill subdomains? (yes:1, no:0)
plot node no?(no=0):
plot domain no?(no=0):
need mesh sizes? (no=0:
(誤差ノルム分布のプロット)
```

Copyright © 2004, Akira Tezuka , Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 2Dコード(r法)の動かし方:コマンド ■

```
% mesh.notch
Input the # of division (dv should be even)
% bc.notch
% elastic.fem2d
% adap.mesh2d ← energy error normのみサポート
sed (0), interpolation error (1) or zz error (2)?
by patch(1), lsm(2), lumped-lsm(21), was(3) or traction(4) method?
input the percentage of error you can admit←QIの指定
error wrt norm is +*+*+* %→現状のQIの計算値
r-method (1) or p-method (2) ?
which value as a weight?: error measure=1, inv. of mesh size=2
how many iterations for nodal relocation?
1-np (1), np-1 (2) or both (3)?
% bc.notch
% elastic.fem2d
% postplot
% postnorm
fill subdomains? (yes:1, no:0)
plot node no?(no=0):
plot domain no?(no=0):
need mesh sizes? (no=0):
```

} 図5.3-5.6のような
パラメータスタディ

Copyright © 2004, Akira Tezuka , Continuum Modeling Research Group, Research Institute for Computational Science, AIST

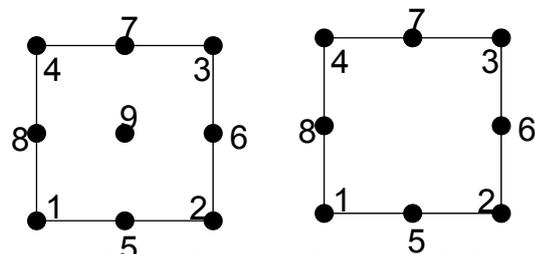
■ 2Dコード(h法)の動かし方:コマンド ■

```
% mesh.notch
Input the # of division (dv should be even)
% bc.notch
% elastic.fem2d
% adap.mesh2d ← energy error normのみサポート
sed (0), interpolation error (1) or zz error (2)?
by patch(1), lsm(2), lumped-lsm(21), was(3) or traction(4) method?
input the percentage of error you can admit←QIの指定
error wrt norm is +*+*+* %→現状のQIの計算値
r-method (1) or p-method (2) ?
% hrefine2d
% bc.notch
% elastic.fem2d
% postplot
% postnorm
fill subdomains? (yes:1, no:0)
plot node no?(no=0):
plot domain no?(no=0):
need mesh sizes? (no=0):
```

Copyright © 2004, Akira Tezuka , Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 2Dコード(p法)の動かし方:コマンド ■

```
% mesh.notch
Input the # of division (dv should be even)
% bc.notch
% elastic.fem2d
% adap.mesh2d
sed (0), interpolation error (1) or zz error (2)?
by patch(1), lsm(2), lumped-lsm(21), was(3) or traction(4) method?
input the percentage of error you can admit←QIの指定
error wrt norm is +*+*+* %→現状のQIの計算値
r-method (1) or p-method (2) ?
input the degree of polinomial for uniform mesh (0. for skip)
tensor product element (1) or serendipity element (2) ?
% bc.notch
% elastic.fem2d
% postplot
% postnorm
fill subdomains? (yes:1, no:0)
plot node no?(no=0):
plot domain no?(no=0):
need mesh sizes? (no=0):
```



tensor product element & serendipity element

Copyright © 2004, Akira Tezuka , Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 2Dコードの説明(1) ■ (2次元アダプティブ有限要素定常弾性解析コード)

・**mesh.notch**(図4.1の問題について初期有限要素メッシュ生成)

出力データ : mesh.notch.dat(基本メッシュモデル. bc.notchへの出力.)

・**bc.notch**(図4.1の問題について初期・アダプティブメッシュへの境界・荷重条件付加)

入力データ : mesh.notch.dat(mesh.notch(基本メッシュ),
adap.mesh2d(r,pアダプティブメッシュ),
hrefine2d(hアダプティブメッシュ)の出力ファイル)

出力データ : elastic.fem2d.dat(有限要素解析用データ. elastic.fem2dへの出力)
[2D共通サブルーチン]

—exnotch : 図4.1の問題の厳密解ルーチン

[1D共通サブルーチン]

—dshape1 : 1次元任意階層型高次要素のためのFE近似関数・微分関数の構築

—arbshape1 : 1次元階層型高次要素導出

—gshape1 : 1次元基本高次関数の構築

■ 2Dコードの説明(2) ■ (2次元アダプティブ有限要素定常弾性解析コード)

・**elastic.fem2d**(2次元定常弾性有限要素解析コード)

入力データ : elastic.fem2d.dat(bc.notchの出力ファイル)

出力データ : adap.mesh2d.dat(誤差事後推定用. adap.mesh2dへの出力。)

: plot.dat(結果表示用. postplotへの出力。)

サブルーチン:

—optinum : バンド幅縮小化のための節点番号付け替え[17]

—stiffmat : 剛性行列及び荷重ベクター作成ルーチン

—pkgene : 局所“剛性”行列構築

—stress: 応力計算ルーチン

[2D共通サブルーチン]

—dshape2 : 2次元任意階層型高次要素のためのFE近似関数・微分関数の構築

—arbshape2 : 2次元階層型高次要素導出

—gshape2 : 2次元基本高次関数の構築

[1D/2D共通サブルーチン]

—skyheight : スカイライン高さ情報の構築

—assemb : 局所“剛性”行列の全体系への組み込み

—skyline : スカイラインマトリックスソルバー

—gaussint : ガウスの積分点情報

■ 2Dコードの説明(3) ■ (2次元アダプティブ有限要素定常弾性解析コード)

・adap.mesh2d(2次元定常弾性有限要素解析の誤差事後推定, r, pメッシュ生成)

入力データ : adap.mesh2d.dat (有限要素解析解データ. elastic.fem2dからの出力.)
出力データ : mesh.notch.dat (r, pアダプティブメッシュデータ. bc.notchへの出力.)
: hrefine2d.dat (メッシュ細分化情報. hrefine2dへの出力.)
: plot.dat (変位・応力出力. postplotへの出力.)

[サブルーチン]

—gene : 節点と要素接続情報等の生成
—sederr2 : 歪みエネルギー密度に基づく誤差推定(4.1.1項に対応)
—lsmsrt2 : LSM, 集中化LSMに基づく1次高次関数導出(4.4.1項に対応)
—wassrt2 : 重み付き平均化法に基づく1次高次関数導出(4.4.2項に対応)
—patsrt2 : 局所パッチ法による1次高次関数導出(4.4.3項に対応)
—tracsrt2 : トラクション法に基づく1次高次関数導出(4.4.4項に対応)
—intnorm2 : 補間誤差ノルム算出(4.3.2項に対応)
—zzenorm2 : ZZ法による誤差ノルム算出(4.3.3項に対応)

[2D共通サブルーチン]

—dshape2 : 2次元任意階層型高次要素のためのFE近似関数・微分関数の構築
—arbshape2 : 2次元階層型高次要素導出
—gshape2 : 2次元基本高次関数の構築
—exnotch : 図4.1の問題の厳密解ルーチン

(次スライドに続く→)

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 2Dコードの説明(4) ■ (2次元アダプティブ有限要素定常弾性解析コード)

・adap.mesh2d(2次元定常弾性有限要素解析の誤差事後推定, r, pメッシュ生成)

(続き)

[1D/2D共通サブルーチン]

—skyheight : スカイライン高さの構築
—assemb : 局所”剛性”行列, 局所”荷重”ベクトルの全体系への組み込み
—skyline : スカイラインマトリックスソルバー
—ludcmp : 密行列解法(前進消去)
—lubksb : 密行列解法(後退代入)
—mparam : 誤差ノルムからメッシュパラメータへの変換(4.5節に対応)
—gaussint : ガウスの積分情報

・hrefine2d(h細分化メッシュ生成)

入力データ : hrefine2d.dat (メッシュ細分化情報. adap.mesh2dからの出力.)
出力データ : mesh.notch.dat (hアダプティブメッシュデータ. bc.notchへの出力.)

サブルーチン:

—jagene : 要素接続情報生成
—refine : 要素細分化ルーチン
—renum : 番号付け替えルーチン
—constr : 拘束節点候補の算出
—macongene : 節点拘束条件式生成

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 2Dコードの説明(5) ■ (2次元アダプティブ有限要素定常弾性解析コード)

・**check.notch** (厳密誤差ノルムのチェックプログラム)

入力データ : check.notch.dat (有限要素・誤差解析データ. adap.mesh2dから)

出力データ : norm.dat, enorm.out (誤差ノルム分布等の出力. postnormへ)

サブルーチン:

—stresse : 応力評価ルーチン

[2D共通サブルーチン]

—dshape2 : 2次元任意階層型高次要素のためのFE近似関数・微分関数の構築

—arbshape2 : 2次元階層型高次要素導出

—gshape2 : 2次元基本高次関数の構築

—exnotch : 図4.1の問題の厳密解ルーチン

[1D/2D共通サブルーチン]

—mparam : 誤差ノルムからメッシュパラメータへの変換(4.5節に対応)

—gaussint : ガウスの積分情報

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ 注意事項 ■ (CD-ROM内 quick.manual)

・elastic.fem2 でのペナルティ係数は節点拘束時の挙動を考慮して, 全自由度数 × 最大剛性マトリックスエントリー値 × 104.5 としている. (5.2.1 節参照)

・誤差ノルムの計算は要素毎の微量の足し算となるため, 計算機の打ち切り誤差の影響により, 計算機毎に多少, 結果の数字が異なる場合がある.

・adap.mesh2d のsubroutine lsmsrt2 でのバンド幅縮小のための番号換え変換アレイ jnt は elastic.fem2 からの入力データによっている.

・hrefine2 内では, 計算効率の良くないアルゴリズムであるが, 説明の簡単化のため, refined mesh の節点として, basic mesh のものをまず入れ込み, subroutine refine で新たに細分化による節点を発生させ, その後, subroutine renum で basic mesh の節点番号に付け替えている.

・hrefine2 のsubroutine renum は番号付け替えルーチンであるが, 一般の場合に有効である.

・adap.mesh2d のsubroutine wassrt2 では, 高次要素の場合に適宜, サンプル点を増やすアルゴリズムとしている.

・adap.mesh2d のsubroutine patsrt2 での基底は, インプリメンテーションをシンプルにするために, global 座標に対して {1,x,y} としている. 基底が線形であれば, global 座標に対する要素パッチの方向性の影響はないが, 双一次の基底 {1,x,y,xy} 以上を用いる場合には局所的な座標に対して基底を定義する等, 工夫が必要である.

Copyright © 2004, Akira Tezuka, Continuum Modeling Research Group, Research Institute for Computational Science, AIST

■ PS file生成コードの注意事項 ■ (CD-ROM内 quick.manual)

- ・ポストスクリプトに興味の向きは適宜, 関連図書を参照されたい.
(Winソフトでは, Gsviewで表示可能)
- ・postplot はQUAD4 要素のみの対応. ゆえに, 高次要素については, 簡易的に要素角点4点のみの情報しかプロットされないので結果の参照には注意が必要.

■ その他の注意事項 ■ (CD-ROM内 quick.manual)

- ・すべてのコードはFORTRAN77 準拠. ダイナミックアロケーションを用いていない.
配列が十分に取られているかどうか, 基本的に読者の方でチェックされたい.
- ・メイン, 各サブルーチン内の変数名は共通としている.
- ・多めに入れたコメント文が理解の一助になれば幸いである.
- ・/1D, /2Dの共通名サブルーチンは同一である.
- ・基本的に汎用コード(厳密解による評価コードは抜かす). 問題依存のサブルーチンについては, [just valid for this problem]等と記載しているので注意されたい.