

有限要素モデルのデータ構造と h 法による順応型要素再分割法に関する研究

正員 大坪 英 臣* 正員 久保田 晃 弘*
正員 北村 充* 正員 川村 恭 己**

A Study on Object-Oriented Data Structures for
the Finite Element Method and
 h -Version Adaptive Mesh Technique

by Hideomi Ohtsubo, *Member* Akihiro Kubota, *Member*
Mitsuru Kitamura, *Member* Yasumi Kawamura, *Member*

Summary

An object-oriented data model for the finite element method is described in this paper. In the data model, which is based on the Entity-Relationship model, a finite element is treated as an object.

The proposed data model is applied to the h -version adaptive mesh technique. In the h -version adaptive mesh technique, an element is divided into 2 or 4 elements recursively to reduce the error norm for the whole analysis domain. The present mesh refinement system is constructed using the C++ object-oriented programming language for the implementation of the data model. To verify the mesh refinement system, we have chosen some 2-dimensional stress concentration and singularity problems. The results show that the newly proposed data model is very useful for the h -version adaptive mesh technique. Moreover, the data model shows great promise for the integrated FEM systems.

1. はじめに

著者らは既報¹⁾で有限要素法事後誤差解析プログラム ERRAND-FEM を用いた r 法による順応型要素再分割法を提案し、それが有限要素解の精度向上に有効であることをいくつかの例題で示した。 r 法はオリジナル有限要素モデルの要素数、節点数およびトポロジーをまったく変えずに解の精度向上を目指すものである。したがって、 r 法は「どこまで向上するかはわからないが、とにかくできるだけ誤差を少なくする」という要素再分割法であり、その解は一義的に得られる。それに対して h 法はオリジナルの有限要素モデルの節点を保持しながら、誤差の大きい要素のみをさらに細かく分割することにより解の精度向上を目指すものである。したがって、 h 法は「ある設定した任意の値まで何とか誤差を少なくする」ときに有効な要素再分割法である²⁾。また r 法ではコーナー上やクラック・チップ上の節点を動かすことが困難であるため、構造物の形状が複雑で

ある場合、解の精度改善が効果的に行われなかったことがあるのに対して、 h 法はその効率(誤差/要素数の比)こそ r 法に劣るものの、いかなる形状を持つ構造物に対しても、とにかくある目標値まで解の精度を改善することができる、という特徴を持つ。そこで本研究では r 法に比較してより自動化に適していると考えられる h 法を用いて要素再分割を行う。

一方、有限要素解析のデータ構造に目を向けてみると、従来の有限要素解析のデータ構造は一般に、単なる表形式のデータの羅列であったことがわかる。具体的には Fig. 1 に示すように、解析コントロール・パラメータ・テーブル、要素タイプ・テーブル、要素番号-節点番号コネクティブティ・テーブル、節点座標テーブル、変位境界条件テーブル、表面力境界条件テーブル、物性値テーブル、等の複数の表が並列に置かれることによって、有限要素モデルが表現されていた、ということが出来る。そして計算機内においてもこの複数の表形式のデータをそっくりそのまま配列というデータ型でストアし、モデルのハンドリングを行なっている。逆の見方をすれば、計算機内に表現しやすい配列というデータ型に合わせて、有限要素モデルのデータ構

* 東京大学工学部船舶海洋工学科

** 東京大学大学院工学系研究科

造が決定されていたことになる。この従来の有限要素モデルのデータ構造自体は実に簡便であるが、順応型要素再分割やズームングを行う場合のように、モデルの一部分だけを取り出したり変更しようとした場合、その都度相互に関連するテーブル（配列）をすべてスキャンし、関連するデータだけを取り出してからでないでデータを変更することができない、という点で明らかに非効率的である。例えば前述の h 法により要素再分割を行う場合、ある要素を再分割するたびに、その要素に隣接する要素間の整合性が満足されているかどうかをチェックしたり、関連する境界条件テーブルを書き換えたりする必要があるが、従来の表形式のデータ構造ではこのデータ間の相互検索に時間がかかる、境界条件の書き換えが煩雑になる、等のさまざまな問題点が生じてくる。これは従来の表形式のデータ構造自体が意味を持たず、実際の有限要素モデルが持つ構造との結びつきが希薄であるためであると考えられる。

そこで本論文では、まず実際の有限要素モデルそのものが Fig. 2 に示すようなある一定の構造を持っていることに着目し、その現実のモデル構造に密着したオブジェクト指向型のデータ構造を提案し、計算機内にインプリメントする方法について述べる。そして次にそのデータ構造およびインプリメンテーションを用いて、 h 法による要素再分

割を行った例を示し、その有効性について議論する。

2. オブジェクト指向型の FEM データ構造

本研究では Fig. 2 に示す実際の有限要素モデルのデータ構造を、オブジェクト指向の考え方をを用いて計算機内にインプリメントする。すなわち Fig. 3 に示すように、実際の解析対象の一部である有限要素一つ一つをそれぞれオブジェクトであると考え、それら要素間の幾何学的関係は相対的なリレーションで表現する。この有限要素一つ一つを表すオブジェクトを要素プリミティブと呼ぶ。これはいわゆるエンティティ-リレーションモデル³⁾の考え方と本質的に同一である。現在 CAD における設計モデルと FEM における解析モデル間のデータの一貫性および変換の困難さが、さまざまなところで問題となっている。この原因の一つとして設計の場と解析の場の間での対象物表現に関するデータ構造の不一致があげられる。本研究で提案するオブジェクト指向型のデータ構造は、この設計と解析間のデータの統合化を目標としたものであり、その結果、両者の結びつきがより柔軟かつ強固になるものと考えられる。本研究と同様に有限要素モデル・ハンドリングの高度化を目的としたデータ構造に関する研究はすでに幾つか存在するが^{4,5,6,7)}、それらの多くは 4 分木あるいは 8 分木モデルを応用したものである。これらのツリー型を用いたデータ構造と比較して、本研究で用いたオブジェクト指向型のデータ構造はよりモデルの実体に近づいており、高度なモデル・ハンドリングの柔軟かつ直感的なインプリメントが可能になると考えられる。

なお、本研究ではオブジェクト指向型のプログラミング言語である C++⁸⁾ をシステム開発用の言語として採用した。C++ は構造化プログラミング言語の 1 つである C にオブジェクト指向の概念を導入することを目的として開発されたものである。実際 C++ は C にクラス概念を導入したものであり、C の高速性や効率の良さはそのままに、データや操作手続きの抽象化を図ることを目的としている。オブジェクト指向型の言語としては他に Smalltalk や CLOS 等が有名であるが、本研究では将来的に大規模な解

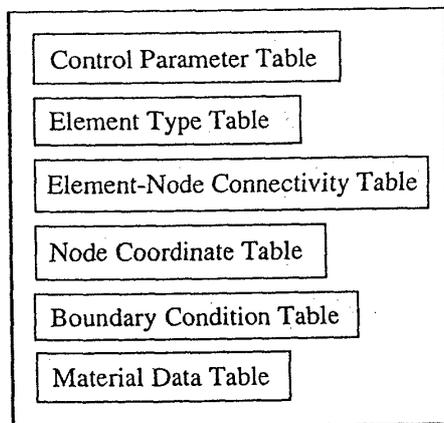


Fig. 1 Table-type data model

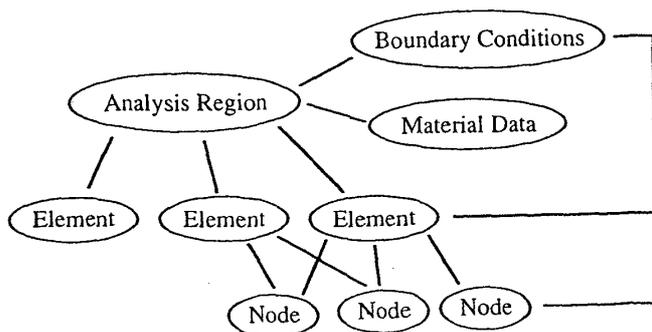


Fig. 2 Data structure of finite element model

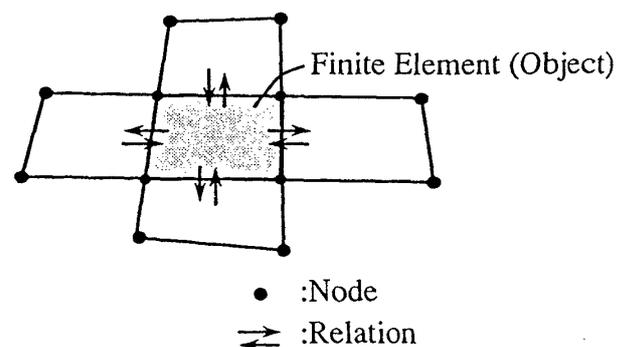


Fig. 3 Element primitive-relation model

析モデルを取り扱うことを考慮しているため、高速・高効率といった点で他のオブジェクト指向型の言語より優れている C++ を用いることとした。さらにこのようなオブジェクト指向型の言語を用いることにより、将来的なシステム拡張の際の生産性もかなり向上すると考えられる。

次に本論文で提案するオブジェクト指向型の有限要素解析モデルについて詳解する。まず最初に、解析領域全体を表すクラス `Analysis_Global` を定義する。そしてそのデータメンバとして、`Element`、`Node` そして `Material` クラスを指すポインタ・テーブルを定義する。節点の座標は `Node` クラスのデータ・メンバとして表現される。`Material` クラスには解析領域の物性値が格納されている。`Element` クラスが前述の要素プリミティブ・オブジェクトであり、これはいわゆる通常の要素ライブラリに相当する部分でもある。ここでは `Element` クラスのすぐ下のサブクラスとして定義されている 4 節点四辺形要素を表すクラス `BQ_Element` (Bilinear Quadrilateral Element) を例に解説する。`BQ_Element` は 4 つのエッジデータ (クラス `Edge`) を持っている。`Edge` クラスは `Startnode` と `Endnode` というエッジの起点と終点のノードを指すポインタ変数を持っている。さらに `Edge` クラスはその下に `Free_Boundary_Edge`、`Distributed_Load_Boundary_Edge`、`Fixed_Boundary_Edge` そして `Relation_Edge` の 4 つのサブ・クラスを持っている。これらはそれぞれ自由境界エッジ、分布荷重境界エッジ、固定境界エッジそしてリレーションをもつエッジを表す。リレーションとは前述のように他の要素オブジェクトとの接合関係を表すものである。この `Relation_Edge` クラスの下に接合する相手の要素 (クラス `Element`) を指し示す `Relation` クラスがあり、その下に様々な要素間の接合状態を表すいくつかのサブ・クラスがあるが、これらについては後述する。最後に以上のデータ (クラス) 構造をまとめたものを Fig. 4 および 5 に示す。

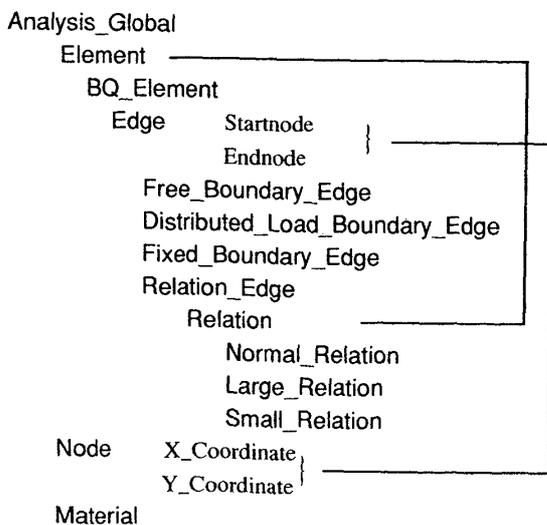


Fig. 4 Class structure

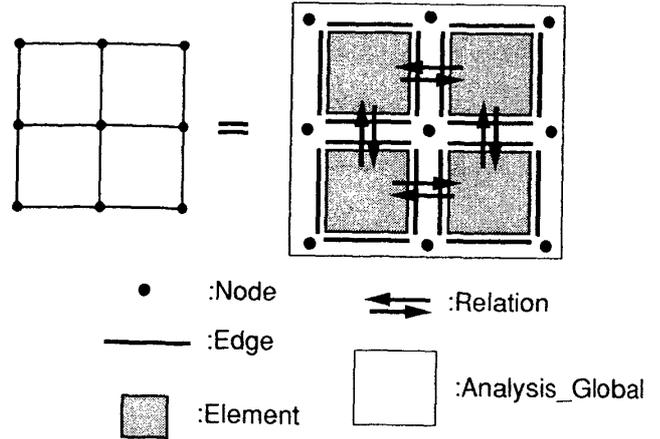


Fig. 5 Object-oriented finite element model

3. h 法による順応型要素再分割

次にこのオブジェクト指向型のデータ構造を用いて h 法による要素再分割を行う方法について説明する。 h 法による要素再分割を行う際にまず問題となる点は、要素間の整合性である。すなわち一般に h 法による要素再分割を行う場合、要素を独立に分割していくと、再分割した部分と再分割しない部分との間で節点の不整合が生じてしまう。この不整合を解消する方法として

- (1) 部分的に三角形要素を用いる。
- (2) 補間ノードを用いる。

の 2 通りの方法がよく用いられる⁹⁾。前者はソルバーに対してなんら変更を加える必要がないかわりに、いったん三角形要素に分割された領域をさらに再分割するのが困難である、といった点で柔軟性に欠ける。それに対して後者の補間ノードを用いる方法は、補間ノード上の変位を他のノードの変位から線形補間して未知数から取り除く必要があるため、ソルバー自体に若干の変更を加える必要があるものの、一貫して 4 辺形要素を使用することができるので柔軟性に富み、何回でも繰り返し要素再分割を行うことが可能である。そこで本研究では後者の補間ノードを用いる方法により要素再分割を行う。なお、補間ノードは原理的に一つの要素辺上にいくつでも使用することが可能であるが、精度上の観点からここでは一つの要素辺上には複数の補間ノードは存在しないという制約を設けた。したがって要素間のリレーションとしては Fig. 6 に示すような Normal, Large, Small の 3 つのリレーションのみを考えれば良いこととなる。

h 法による要素再分割法では初期の粗い解析モデルをもととして、その解析結果に対して事後誤差評価を行い、その結果誤差が大きいと判断された要素から順次その要素をより小さい要素に分割する。 r 法と同様に h 法においても、各要素の誤差ノルムが等しくなるように要素を分割していけばよい¹⁾。本研究ではその要素分割のメソッドとして、Fig. 7 に示すような 4 分割および 2 分割の 2 つのメソ

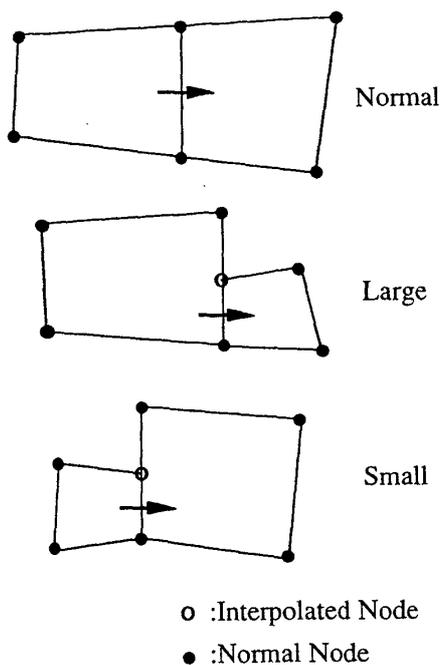


Fig. 6 Three relations

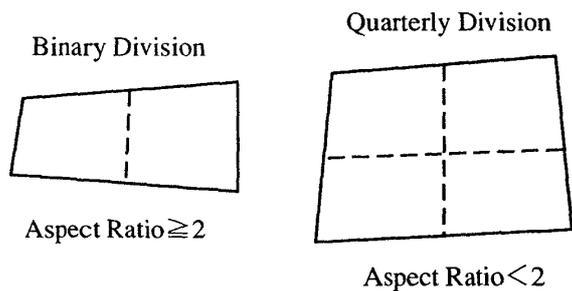


Fig. 7 Binary and quarterly divisions

ッドを作成した。一般の要素に対しては4分割のメソッドを用いるが、要素の形状が歪んでいてそのアスペクト比が2以上のときには、要素形状が正方形に近づくように2分割のメソッドを用いることとする。このような2つの分割メソッドを用意することにより、再分割過程において初期要素形状のゆがみのある程度矯正することができる。なお前述のように、要素を4分割あるいは2分割する際には、「1つの要素辺上には1つの補間ノード」という制約から、要素を全く独立に分割することは不可能で、いちいち接する他の要素との整合性をチェックする必要がある。ここではその問題をデータ構造とメッセージ・センディングの考え方をを用いて解決する。すなわちある要素が分割する際には、まずその要素がその時接している他の要素に「4/2分割した」というメッセージを送信する。そのメッセージを受信した要素は、もしメッセージを送信した要素との間にすでに補間ノードが存在する場合、すなわち受信した要素の送信した要素とのリレーションがLargeである場合、ま

ずメッセージを受信した要素が「4/2分割」しリレーションをNormalに書き換え、さらに他の要素にメッセージを送信する。その後、最初にメッセージを送信した要素が「4/2分割」し、改めてリレーションをSmallに書き換える。以後はそのようなメッセージのやり取りが再帰的に行なわれることによって、無理なく解析領域全体の要素間の整合性が満足されることになる。この過程の一例をFig.8に示す。ここでは要素Aの誤差ノルムが大きいと判定されて、この要素を再分割することを考える。まずAの要素のBに対するリレーションがSmallなので、まずBが分割してからAが分割する必要がある。さらにBの要素Cに対するリレーションもSmallなので、さらにCも分割する必要がある。本研究で用いたデータ構造と再帰的なメッセージ・センディングの考え方をを用いると、このような過程を非常に簡単に実現することができる。

以上のデータ構造とメソッドを組み合わせるh法による要素再分割を行う。前述のようにプログラムの開発にはC++言語を用いたが、さらにユーザーとのグラフィック・インターフェイス部分を開発するために、X-windowシステムおよびC++で書かれたグラフィックス・ライブラリInterViewsを用いた。有限要素法のソルバーや誤差解析ブ

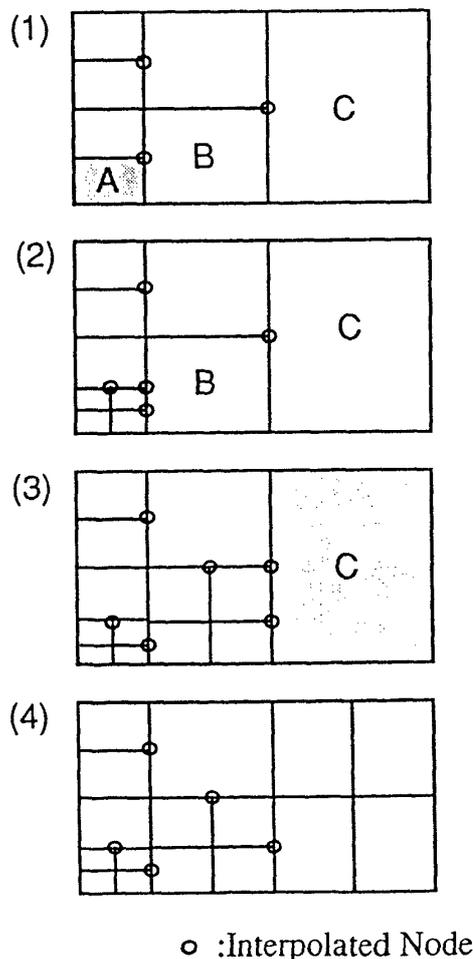


Fig. 8 Element consistency

ログラム (ERRAND-FEM¹⁰) と本研究で開発した要素再分割プログラムとのリンクは、現状では表形式の中間ファイルを介して行なっている。

4. 要素再分割例と考察

まず始めに、既報¹⁾で r 法による要素再分割を行なった、円孔を有する正方形板 (Fig. 9) およびクラックを有する正方形板問題 (Fig. 10) について、 h 法による要素再分割を行った結果を示す。解析条件、物性値等も既報と全く同様である。本論文では要素数を制約条件として要素再分割を行なった。すなわち、目標とする要素数をユーザーがインプットするとシステムは誤差の大きい要素から順次、要素数がほぼその目標数に達するまで、分割を続ける。そして要素が4分割されると4節点4辺形要素の場合の事前誤差評価結果から、その要素の誤差ノルムは1/2になったものと判断し、2分割された場合は $1/\sqrt{2}$ になったものと判断する。なお要素数が連続的に増加できないこと、そして要素間の整合性をとるために要素の持つ誤差の大きさは関係なく分割される要素が存在することから、要素数の目標値と結果的に得られた要素数は厳密に一致はしない。

Fig. 11 は円孔を有する正方形板の結果である。Mesh-1 は等分割されたオリジナル・メッシュ、Mesh-2 は r 法による要素再分割結果¹⁾、そして Mesh-3 は h 法による要素再分割結果である。 h 法の場合の目標要素数はオリジナルのほぼ倍の 100 である。いずれの要素再分割結果においても解析領域全体の相対誤差 $\|e\|/\|u\|$ はオリジナル・メッシュの 1/2 以下に減少しており、要素再分割による誤差低減が効果的に行われていることを示している。Fig. 12 は h 法による要素再分割結果における、全要素数 N と相対誤差の関係を示したものである。要素間の整合性をとるために誤差低減の観点からは不必要な要素まで分割するのでグラフに若干の凸凹はあるものの、 h 法による誤差の収束率も r 法

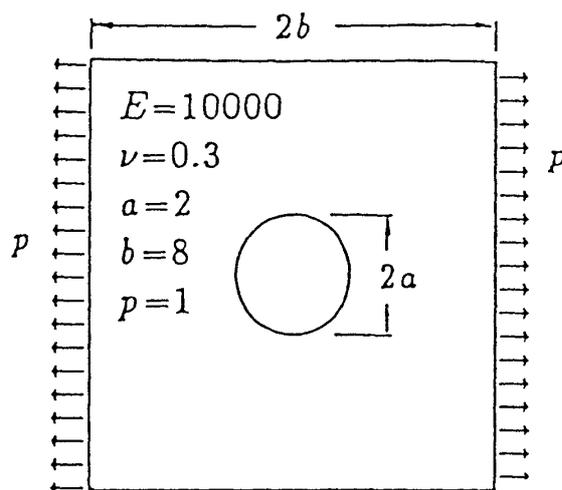


Fig. 9 Square plate with a hole

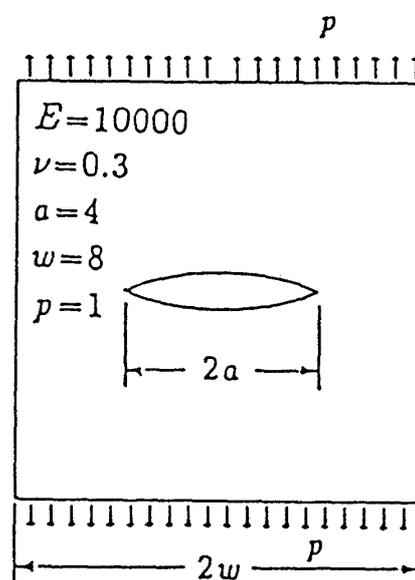


Fig. 10 Square plate with a crack

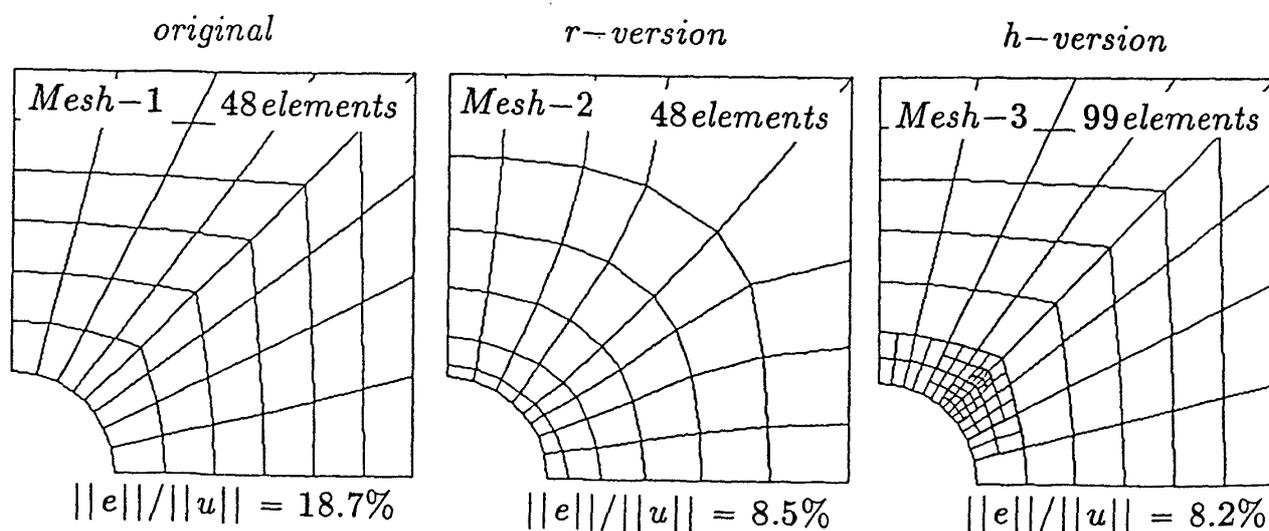


Fig. 11 Original and adaptive finite element models

同様、事前誤差解析結果から得られる均等分割場合の誤差の収束率 $(\sqrt{N})^{-1}$ に比較してかなり良いことがわかる。

Fig. 13 は特異点を有する問題の代表例として取り上げたクラックを有する正方形板の要素再分割結果である²⁾。目標要素数は Mesh-5 が 100, Mesh-6 が 200 である。この図に示すように特異点であるクラック近傍で要素が細くなり、解析領域全体の誤差も減少している。このように特異点を含む問題の場合、 r 法ではクラック近傍での解の精度は向上するものの、解析領域全体の相対誤差は必ずしも減少しない¹⁾。それに対して h 法を用いると解析領域全体の相対誤差が低減し、さらにオリジナル・メッシュ (Mesh-4) では 3.50 であったクラック先端における応力値のピークも Mesh-5 では 7.15, Mesh-6 では 7.67 と 2 倍以上に増加する。

次に典型的な応力集中問題である、非常に小さな rounded-corner を持つ L 型プレート (Fig. 14) の要素再分割結果を示す。Fig. 15 がオリジナルの均等分割された有限要素モデル (Mesh-7) である。要素数は 154 であり、この

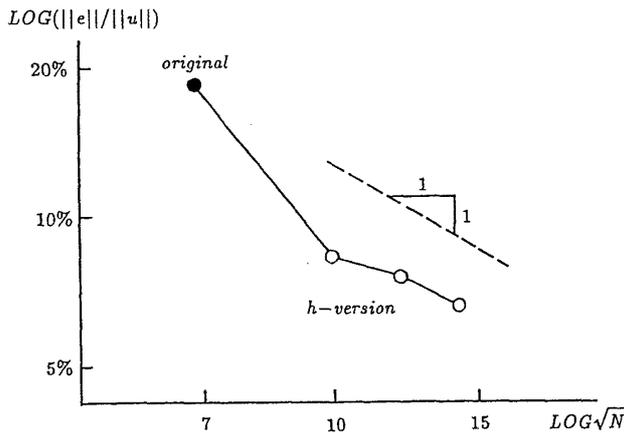


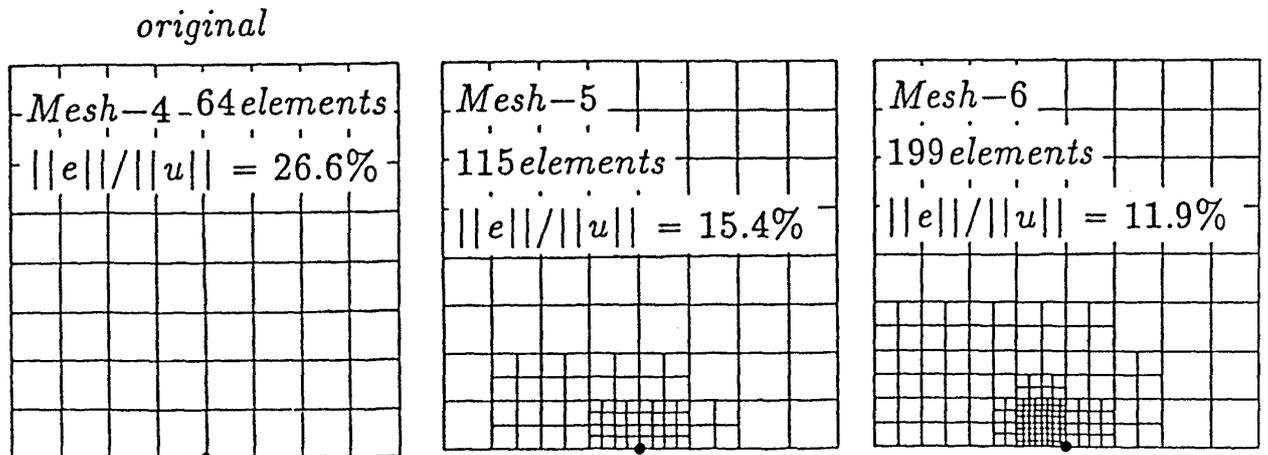
Fig. 12 Convergence rate of relative error

モデルを用いた応力解析結果に対して誤差解析を行った結果、領域全体の相対誤差 $\|e\|/\|u\|$ は 34.4% であった。このモデルをもとに目標要素数を 200, 275, 350 として h 法による要素再分割を行った結果を Fig. 16 に示す。右側の図はコーナー近傍の要素を拡大したものである。この図からコーナー最近傍の要素で 2 分割メソッドが用いられていることやコーナー近傍の要素が他の要素に比較して極めて細かくなっているにもかかわらず、要素の整合性に関して特に問題は生じていないことがわかる。なお、ここで用いた h 法では R 部の誤差が過小評価されているオリジナルメッシュの誤差分布に基づいて要素再分割を行なっているため、R 部はやや粗い要素となっている。実際 h 法で分割した後の誤差は R 部に集中しており (Fig. 17 参照)、真の誤差分布に従えば R 部の要素はより細くなるであろう。

Fig. 17 に ERRAND-FEM によって解析された要素毎の誤差分布を示す。この図を Fig. 15, Fig. 16 とあわせて見ると、応力が集中し誤差が大きくなるコーナー部の要素が細くなり、次に垂直部に比較してより誤差の大きい水平部の要素が細くなっていくことがわかる。この問題の場合も、Mesh-10 では解析領域全体の相対誤差がオリジナル・モデルの半分以下に減少している。

Fig. 18 は Fig. 12 と同様に全要素数と相対誤差の関係プロットしたものである。円孔問題と同じく本例題の場合でも、 h 法による誤差の収束率は、事前解析誤差解析結果から得られる収束率に比較して、かなり向上していることがわかる。

最後にデータサイズの問題について簡単に触れておく。本研究で提案したオブジェクト指向型のデータモデルのサイズを従来の表形式のデータモデルのサイズとを比較すると、2次元問題の場合、前者は後者の約 1.4 倍になる。データ量の増加はほぼリレーションのデータ量に等しい。処理速度に関しては、オブジェクト指向型の場合はデータの



• crack tip

Fig. 13 Original and adaptive finite element models

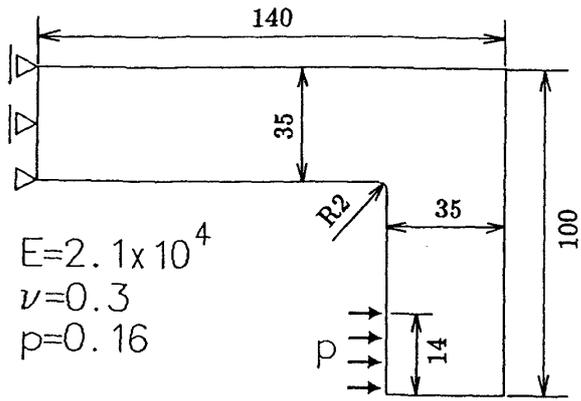


Fig. 14 L-shaped plate with rounded corner

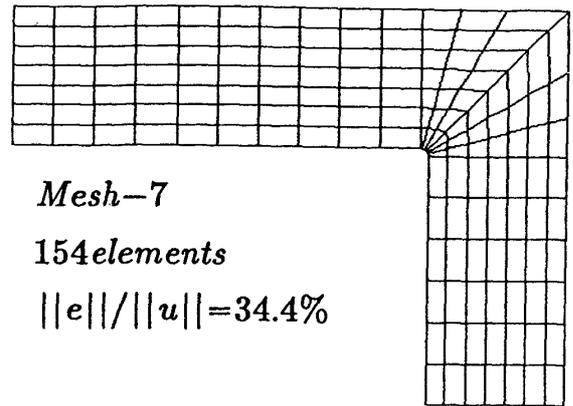


Fig. 15 Original finite element model

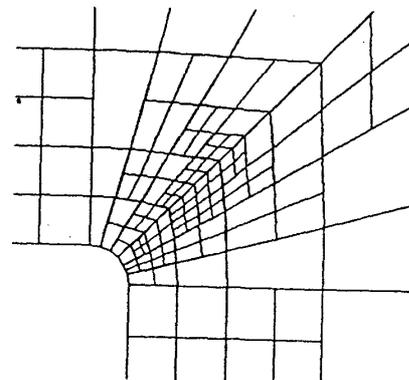
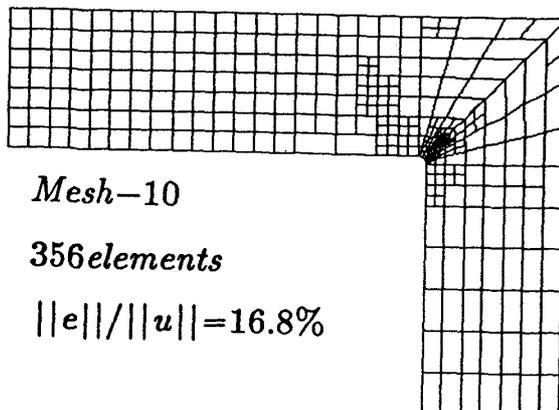
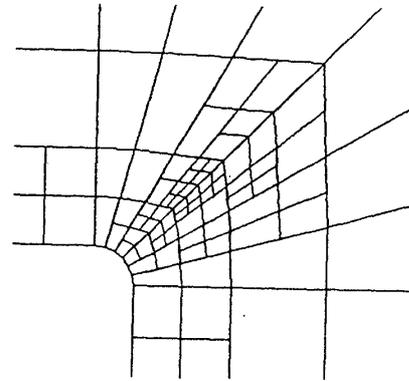
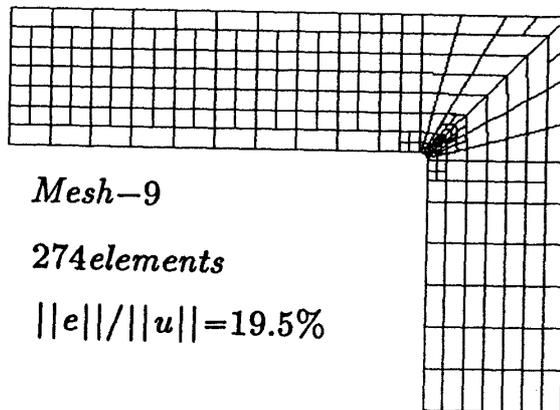
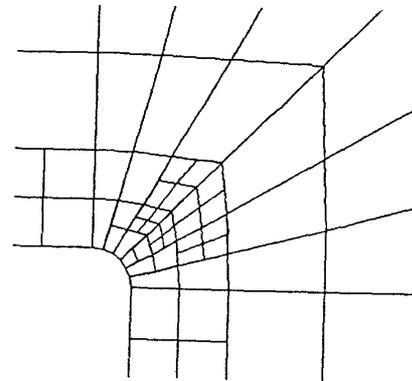
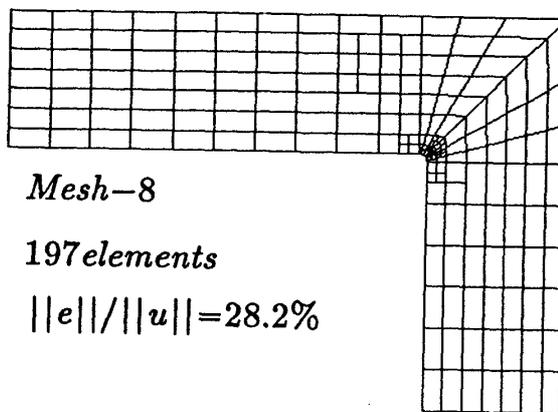


Fig. 16 Adaptive finite element models

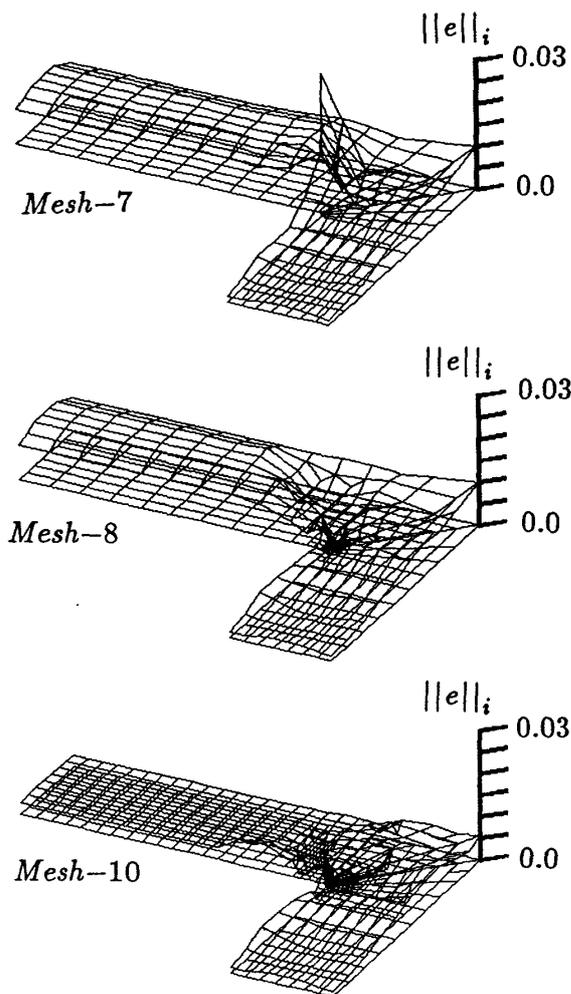


Fig. 17 Estimated error distributions by ERRAND-FEM

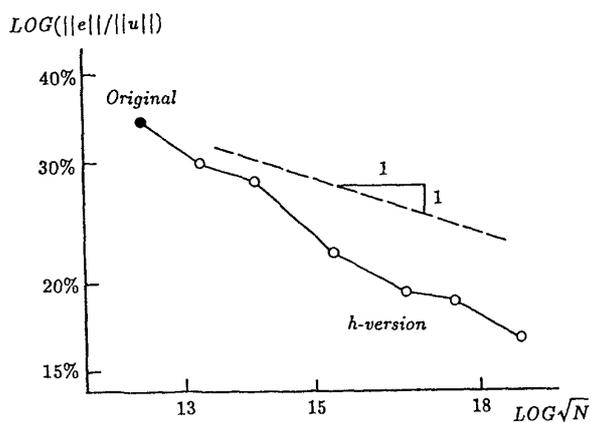


Fig. 18 Convergence rate of relative error

サーチが不必要なため、モデルが大きくなればなるほど後者に比較して向上する。

5. おわりに

本研究では有限要素モデルのより柔軟かつ高度なハンドリングを可能にするための、オブジェクト指向型のデータ構造を提案した。これは有限要素一つ一つをオブジェクトとして取り扱い、要素間の接合関係をリレーションで表現するものであり、有限要素モデルの実体とより密接に結び付いたものとなっている。そしてそのデータ構造を利用してh法による順応型要素再分割を行なった結果をいくつかの例題を用いて示した。その結果、有限要素解の精度向上に極めて有効なh法による要素再分割がより簡便に行えるようになった。しかしながら、本研究で示したデータ構造の応用は特に順応型要素分割だけに限っているわけではない。今後はこのデータ構造をベースとして、有限要素モデル作成/編集システムの開発を行い、有限要素解析の統合化システムの開発を目指す予定である。

参考文献

- 1) 大坪英臣, 久保田晃弘, 北村充: 有限要素法の事後誤差評価結果を利用した順応型要素分割法に関する研究, 日本造船学会論文集, 第167号, (1990)。
- 2) 大坪英臣, 北村充, 久保田晃弘: 有限要素法の最近の動向(その6) - 誤差評価法の実際問題への適応 -, 日本造船学会誌, 第736号, (1990)。
- 3) 小山健夫他: 構造物建造のためのプロセスプランニングの研究, 日本造船学会論文集, 第166号, (1989)。
- 4) Rheinboldt, W. C. et al.: On a data structure for adaptive finite element mesh refinements, ACM Trans. Math. Soft., Vol. 6, (1980)。
- 5) Rivera, M. C.: Design and data structure of fully adaptive, multigrid, finite element software, ACM Trans. Math. Soft., Vol. 10, (1984)。
- 6) Wang, K. C. et al.: A refinement scheme and data structure for bi-quadratic elements with application to viscous-flow problems, Int. J. Num. Meth. Fluids, in review, (1988)。
- 7) Carey, G. F. et al.: A class of data structures for 2-d and 3-d adaptive mesh refinement, Int. J. Num. Meth. Eng., Vol. 26, (1988)。
- 8) Stroustrup, B.: The C++ Programming Language, Addison-Wesley, (1986)。
- 9) 岡田哲男: 船体構造解析に適した自動要素分割法, 造船設計における構造解析と信頼性工学, 日本造船学会, (1990)。
- 10) 大坪英臣, 北村充: 有限要素法による平面応力問題の誤差評価と応力修正に関する研究, 日本造船学会論文集, 第165号, (1989)。