

## 有限要素法用数値シミュレーション言語 FEEL

- 数値解析プログラム作成支援の一試案 -

1 X - 8

藤尾 秀洋, 速水 謙

日本電気(株) C&C 情報研究所

### 1 はじめに

数値シミュレーションの実行環境における計算機の発達の結果、たとえばベクトル・並列計算機の発達や高速で低価格のグラフィックワークステーションの出現による実行環境の向上には目覚ましいものがある。これらは、数値シミュレーションの日常化と大規模化に大きな役割を果たしたが、その結果、科学者や技術者に対し、数値シミュレーションを行う為の手間の増大をもたらし、これを解決し、数値シミュレーションの利用環境を向上させる為の、ソフトウェアの側からの試みも近年盛んに行われている [1] ~ [5]。

今回、その1つの試みとして、有限要素法を用いて数値シミュレーションを行う際に、プログラムの作成を支援するツールを試作した。このシステムは数式処理等の記号処理を1つの基本要素技術としている。数年前迄は数式処理を実用的な時間内で行う為には、大型計算機で実行する必要があった。しかし、昨今の高速ワークステーションの発達は、計算機利用者一人一人にかつての大型計算機に匹敵する計算速度を与え、数式処理をユーザが手軽なツールとして使うことの出来る環境を実現した。また、有限要素法におけるプリ・ポスト処理を行う環境も、高速ワークステーションの守備範囲となった。今回試作したツールは、数式処理を用いて有限要素法のプログラムを自動生成し、メッシュ生成や可視化のためのインタフェースも自動生成することにより、有限要素法数値シミュレーションのための使いやすい統合的な利用環境をワークステーション上で実現することを目指している。

### 2 有限要素法のプログラミングの特徴

有限要素法による数値シミュレーションの作業は、

[プリ処理] メッシュの生成

[数値計算] 連立一次方程式等の組み立てとその求解

[ポスト処理] 誤差評価、結果の可視化

という、3つの部分に分けられる。多くのユーザは、汎用のメッシュ生成パッケージによりメッシュを生成し、自作または汎用の数値計算プログラムとインタフェースをとって結合させ、結果の可視化処理はまた可視化プログラムとインタフェースを取ることによって実現する。このうち、数値計算を行うプログラムは扱う問題によって違いが大きい部分であるが、この部分も次の共通の機能ブロックを持っている。

Finite Element Numerical Simulation Language FEEL.  
Hidehiro Fujio and Ken Hayami  
C&C Information Technology Research Laboratories, NEC  
Corporation.

[メッシュ情報の読み込み]

[連立一次方程式の係数、右辺ベクトルの組み立て]

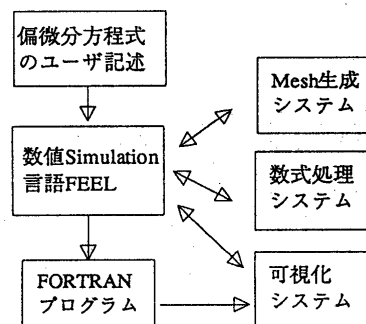
[ディリクレ条件、ノイマン条件の考慮]

[連立一次方程式の求解]

有限要素法の基本的な計算は、要素上で、ある数式を積分しそれを全体に足し合わせるという処理であり、これは全ての有限要素法で共通である。また、その計算を実行する際のメッシュに関するデータ配列の扱いなどは、ほとんど同一のプログラム、データ構造を持っている。そのため、有限要素法はプログラム作成という点で見ると(高速化の技法は別として)それぞれの処理、およびプログラムを部品として扱うことが可能であり、それぞれモジュールとして制作することができるという特徴を有している。これらの意味で、有限要素法のプログラムは型にはまったプログラム構造をしていると言える。

### 3 数値シミュレーション言語 FEEL の処理

試作した2次元有限要素法のプログラム生成の処理系 FEEL(Finite Element Equation Language) は、有限要素法のそれぞれのモジュールに対して、ユーザが必要最低限の記述を与える事により、プログラムを自動生成し、プログラム作成時の手間を軽減することを目的とする。また、生成したプログラムをユーザが後から自由に手を加えることが出来る様に1つの汎用的な有限要素法のプログラム・データ構造を提供することもその目的の一つである。ユーザが与える記述は、計算する領域の形状、メッシュ生成システムに渡すべきメッシュ生成規則、解こうとする偏微分方程式とその境界条件、および近似にもちいる基底関数や数値積分法、結果の表示方法などである。数値シミュレーション言語 FEEL の構造は次の様になる。



数値シミュレーション言語FEELのシステム構成

## メッシュ生成法の記述

領域の形状に関するデータや、メッシュ生成時のコントロールデータを、汎用のメッシュ生成システムに送ることにより、要素分割情報を得る。その際、領域が部分領域に分割される場合にはその部分領域を表すインデックスを生成する。

## 偏微分方程式のユーザ入力と離散化式の計算

FEELでは、偏微分方程式の離散化に関する部分のデータは全て数式として取り扱う。基底関数の指定は、基準要素における座標系を用いた数式の形で、対応する節点の座標とともに指定する。また、数値積分法の指定は基準要素での座標系を用いて積分点とその重み係数の数式が入力となる。その際、特殊な係数関数に関する補間法も数式で指定可能である。偏微分方程式は、微分演算子を含む数式として与えられる。これら数式で表現された方程式、および基底関数の情報から連立一次方程式に帰着する場合は、まず与えられた偏微分方程式を数式処理で弱形式に変換する。次に、偏微分方程式中の関数を表す変数を、それに対して指定された基底関数の1次結合の形の数式で置き換える。試験関数も同様に対応する具体的な数式で置換する。そうして得られる数式に対して、数式微分を行って積分前の係数を数式の形で求め、記号的な積分処理を行って連立一次方程式の係数の計算式をあらわす数式を得る。

## 4 記述例

下図は2次元3角形要素で、ユーザが指定する基底関数及び数値積分法の記述例である。

```
define px[3] {
(0,1/2): 1-2*lda3;
(1/2,1/2): 1-2*lda1;
(1/2,0): 1-2*lda2;
};
```

基底関数の指定

```
integral-2d-triangle [SG]{
((g(1,0)+g(0,1)+g(1,1))*3 +
(g(1/2,0)+g(0,1/2)+g(1/2,1/2))*8+
g(1/3,1/3)*27)*AREA/60};
```

3角形要素数値積分法の指定

基底関数は標準3角形での空間座標  $x, y$ 、もしくは重心座標でその対応する節点と共に上図の様に指定する。ここで、 $lda1, lda2, lda3$  は、標準3角形の頂点  $(0,0), (1,0), (0,1)$  に対応する重心座標である。

ユーザの定める数値積分法は上図の様に、標準3角形上での数値積分式を数式の形で表記する。 $g(x,y)$  は被積分関数、 $AREA$  は、この3角形の面積を表すシンボルである。

次図はFEELにおける、偏微分方程式問題の記述例である。FEELは、この図で表された偏微分方程式を弱形式化したのち、未知関数  $u, v, p$  や、既知関数  $fx, fy$  の部分を、実際に基底関数の1次結合の数式で置換して、連立一次方程式の係数となるべき被積分関数の数式を求める。そして、ユーザ指定の数値積分公式を用いるか、ま

```
solve [u,v,p][fx,fy] {
eq: -lap(u)+dx(p) = fx,[p2];
eq: -lap(v)+dy(p) = fy,[p2];
eq: dx(u) + dy(v) = 0,[p1];
dbc: u = 0, on ab,cd,de;
..... 基本境界条件
nbc: dn(v) = 0, on cd;
..... 自然境界条件
}
```

Stokes方程式

たは、被積分が重心座標の多項式で表される場合などには解析的に積分式を求めて、係数の計算式を算出し有限要素法プログラムを生成する。

## 5 おわりに

今回試作したFEELは、有限要素法のプログラムをユーザの数学的な偏微分方程式の問題表記から、有限要素法のプログラムを自動生成するシステムである。それは、連立一次方程式の係数の計算式を導く際にあらわれる、特殊な要素の内挿法や数値積分法に関する計算式をユーザにとって記述が容易な形式の入力情報より数式処理を用いて求め、メッシュ情報を扱うインデックス配列や、可視化のためのライブラリーの呼び出し等と共に、FEELが提案する有限要素法のプログラム構造で、FORTRANのソースプログラムとして出力する。FEELの意図は、プログラムの生成補助システムであり、その目的が、ユーザによって容易に変更可能なプログラムおよびプログラム構造を提供すること、という点が、[1]～[5]のアプローチと違っているといえる。今回試作したシステムは2次元版であるが、3次元版への拡張では方程式の離散化は全く問題なく拡張可能であるが、3次元領域の入力はコンピュータグラフィックスを用いて対話的に行う必要があるだろう。

## 参考文献

- [1] Rice, J.R. and Boisvert, R.F.: 'Solving Elliptic Problems Using ELLPACK', Springer-Verlag(1985).
- [2] Granville Sewell: 'Analysis of a Finite Element Method: PDE/PROTRAN', Springer-Verlag(1985).
- [3] 梅谷征雄他: '数値シミュレーション用プログラミング言語 DEQSOL', 情報処理学会論文誌, Vol.26, No.1, pp.168-180(1985).
- [4] Allain Bensoussan 編: 'MODULEF - A Modular Library of Finite Elements', INRIA(1986).
- [5] 三村泰成他: 'オブジェクト指向型有限要素解析システムの開発', 構造工学における数値解析シンポジウム論文集, Vol.16, pp.601-606(1992).
- [6] 佐々木建昭他: '数式処理システム', 昭晃堂ソフトウェア講座 36(1986).