

科学技術シミュレーションプログラム合成支援 II

2S-6

離散化とプログラム合成

飯島邦彦、真鍋保彦、Choompol Boonmee、川田重夫
長岡技術科学大学工学部電気系

1 はじめに

本研究は、科学技術シミュレーションプログラム合成支援における離散化とプログラム合成のところである。離散化は差分法とし、ここでは記号処理によりモデル式となる偏微分方程式を差分方程式に変換する一連の処理について述べている。プログラム合成では、離散化された差分方程式や解法を基に、Fortranソースコードを生成することが述べられている。

離散化にせよプログラム合成にせよ、どのような偏微分方程式なのか、次元数はいくつなのか、どういった離散化をし、どのような解法でシミュレーションするのかなど、様々なデータが必要になる。これらの必要なデータは、ある定まった書式の入力仕様ファイルに記述され、システムはこの入力仕様データを基に、離散化やプログラム生成を行う。また、必要なデータが入力仕様記述されていない場合や、システムだけでは選択し厳しい場合には、ユーザに入力や選択を促すものとなっている。

2 離散化

離散化とは、連続量で表された偏微分方程式を離散値で近似し、数値計算するために代数方程式の形に変換することである。離散化の方法には、有限差分法、有限要素法、境界要素法などがあるが、本システムは現在のところ有限差分法を採用している。

差分法は、具体的には未知関数 u の偏導関数である $\partial u / \partial t$, $\partial u / \partial x$, $\partial u / \partial y$, などを差分商に近似したものに置き換えることにより、差分スキームを得る。

本システムでは、差分近似を時間についての偏導関数には、一次精度の前進・後退差分、および二次精度

の中心差分とし、空間についての偏導関数には、二次精度の中心差分、および対流項には一次精度の風上差分といった制約を与え、記号処理により偏導関数を差分商で置き換える。[1][2] 具体的には風上差分近似は、次式のようになる。また、二階以上の偏導関数は、一階微分の再帰的な処理により実現される。

$$u \frac{\partial u}{\partial x} \Big|_{x=x_i} = \frac{u_i + |u_i|}{2} \frac{u_i - u_{i-1}}{\Delta x_{i-1}} - \frac{u_i - |u_i|}{2} \frac{u_i - u_{i+1}}{\Delta x_i} \quad (1)$$

しかし、差分近似を決めただけでは不十分である。偏微分方程式は陽的に解かれるのか陰的に解かれるのか、またメッシュにはスタガードメッシュを用いるため、変数の定義されている場所が、セルの境界なのか中心なのか、セルのどの場所での偏微分なのか、といった情報を得て、適切な差分スキームが得られる。また、境界条件においても離散化する必要があればそれについても行う。

例として図1に示すように、入力仕様中の %p d e の下に偏微分方程式を記述する。偏微分方程式は複数記述でき、記述の順番は差分スキームを生成しソースコード生成に直接反映する。ここで、本システムでは Lisp 言語により記号処理を行っているので、入力する偏微分方程式は Lisp の前置表現としている。

3 プログラム合成

プログラム合成 [2][3] は、離散化によって生成された差分スキームや入力仕様のデータと科学技術シミュレーションプログラムの一般的性質を基に、FORTRANソースコードを生成する。プログラムの主な流れを図2に示す。科学技術シミュレーションプログラムは、差分スキームの解法や与えられた境界条件等により、生成するルーチンが異なってくる。例えば、非定常問題に対する陰解法や、定常問題に対する解法は、連立方程式を解くことになり、行列計算をするわけであるが、その場合どのような行列計算用のサブルーチ

Computer-Assisted Numerical Simulation Code Generation II Discretization and Code Generation

Kunihiko Iijima, Yasuhiko Manabe, Choompol Boonmee, Shigeo Kawata

Nagaoka University of Technology, Nagaoka 940-21, Japan

```

..
%pde
cu
( (+ (fdet u) (* u (fdesx u)) (* v (fdesy u)))
=
( (+ (* -1 (^ rho -1) (fdesx p))
(* (^ Re -1) (+ (fdesx2 u) (fdesy2 u)))) )
cv
( (+ (fdet v) (* u (fdesx v)) (* v (fdesy v)))
=
( (+ (* -1 (^ rho -1) (fdesx p))
(* (^ Re -1) (+ (fdesx2 v) (fdesy2 v)))) )
cp
( (+ (fdesx2 p) (fdesy2 p))
=
(* 2 (- (* (fdesx u) (fdesy v))
(* (fdesx v) (fdesy u)))) )
..

```

図 1: 入力仕様での偏微分方程式の記述

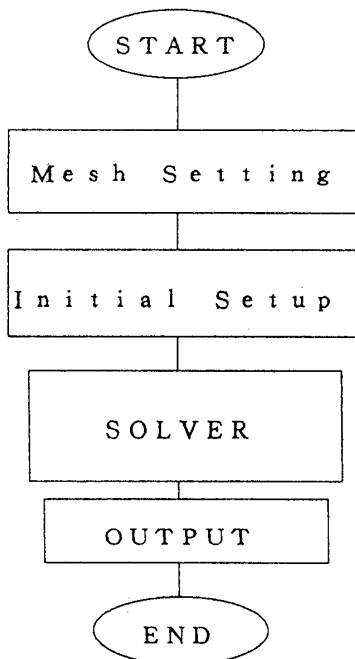


図 2: シミュレーションの流れ

```

DX=1
DHI=30.DO
{{ (I 1 MGI) (J 2 NGRIDY)
P(I,J)=0.0DO
}}
これらの式が、例えばDO番号を100として、
DX=1
DHI=30.0DO
DO 100 I=1,MGI
DO 100 J=2,NGRIDY
P(I,J)=0.0DO
100 CONTINUE

```

図 3: ユーザプログラムの変換

ンを使用するかどうかは、直接ユーザが選べるようにしてある。また、プログラム合成のための入力仕様からの情報に不足があれば、その都度、入力を促す。現在、本システムでは、陽的解法、陰的解法、ADI法およびSOR法をサポートしようとしている。また、個々のシミュレーションに特有であり、システムの処理だけでは十分対処できないところは、ユーザ自身のプログラムの記述ができるようになってきている。このような場合、DOループ番号を考えずによい書式で記述できる。図3にユーザプログラムの例を示す。

4 まとめ

科学技術数値計算シミュレーションソフトウェア合成支援システムの離散化、およびプログラム合成についての概要を説明した。両者とも現在開発中である。今後は、離散化では生成される差分スキームの安定性の解析や有限要素法の離散化もサポートしていきたい。プログラム合成では、ライブラリの追加やそれを取り込みやすいシステム構成の構築を行っていきたい。

参考文献

- [1] 下地貞夫: 数式処理, 森北出版,(1991)
- [2] R.Liska and L.Drska: "A REDUCE package for automation of FInite difference method for solving pDE", *IISAC-90*, Int. Symp. on Symbolic and Algebraic Computation, Tokyo, August 20-24,(1990)
- [3] R.Liska: "Numerical code generation for finite difference scheme solving", *Computational and applied mathematics I*, Proc.13th IMACS Congress, July 22-26,(1991)