

シミュレーションプログラム自動合成システム

5K-2

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

1 はじめに

本研究では科学技術数値シミュレーションの支援ツールとして自動プログラミングシステムを研究開発している。ユーザがシミュレーションを行うために必要な情報を与えれば、本システムは記号処理を用いて自動的に数値シミュレーション用のソースプログラムを合成する。またシミュレーションプログラムによって計算された結果を可視化する可視化システムもサポートしている。特に本文ではシステムのプラットフォームとなる Standard LISP の改良及び知識ベースについて述べる。

2 離散化部及びプログラム合成部

本システム構成は大きく分けて3つの部分に分けられる。離散化部、プログラム合成部、可視化部である。

離散化部では離散化に際して、問題の偏微分方程式群を線形計算が行えるように項の移動・項のまとめ等を行い、式を整理する。本システムの離散化部とプログラム合成部は、記号処理を得意とする LISP 言語で開発上で開発されている。例えば、次の様な偏微分方程式が与えられる。

$$\frac{\partial u}{\partial t} - u \frac{\partial u}{\partial x} + a \frac{\partial^2 u}{\partial x^2} = 0 \quad (1)$$

システム内部ではこの式を次のようにリストの形で表現している。

$$(+ (fdet u) (* -1 u (fdesx u)) (* a (fdesx2 u))) = 0 \quad (2)$$

時間的に前進差分で空間的中心差分でこれを離散化を行うと、次のようになる。

$$\frac{u_i^{n+1} - u_i^n}{dt} - u \frac{u_{i+1}^n - u_{i-1}^n}{2 * dx} + a \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{dx^2} = 0 \quad (3)$$

これをシステム内部では u_{i+1}^{n+0} を (u 0 1) のように書き表し、前置表現で表現すると次のようになる。

$$\begin{aligned} & ((+ (* (^ dt -1) (+ (u 1 0) (* -1 (u 0 0)))) \\ & (* -1 (u 0 0) 1/2 (^ dx -1) (+ (u 0 1) (* -1 (u 0 -1)))) \\ & (* a (^ dx -2) (+ (u 0 1) (* -2 (u 0 0)) (u 0 -1))) \\ & = 0) \end{aligned}$$

プログラム合成部では、これらの情報を利用してプログラム合成を行う。これも LISP 言語上で開発している。現在、FORTRAN プログラムを合成している。

3 システムプラットフォーム

本システムでは開発環境として、大きく2つに分けられる。一つは記号処理や数式処理が比較的実現しやすい LISP 言語で開発された部分、もう一つは GUI (Graphical User Interface) 環境を開発しやすい C 言語で開発

¹ Computer-Assisted Numerical Simulation Code Generation

² Choompol Boonmee, Yasuhiko Manabe, Kunihiko Iijima, Toshiaki Shibui, Hiroyuki Tsuritani and Shigeo Kawata, Nagaoka University of Technology, Nagaoka 940-21, Japan

された部分である。LISP の部分では離散化部とプログラム合成部が、C の部分では可視化部が実現されている。現在本システムの LISP の部分では COMMON LISP を採用している。このシステムは非常に大きく、実行する際に大容量のメモリを必要としており、GUI 環境を実現しにくい。

これらの理由から、記号処理の開発基盤として、この COMMON LISP よりは非常にコンパクトな Standard LISP に将来移行しようとしている。C 言語で書かれた Standard LISP を改良し、2 つに分かれていたシステム全体を C レベルで一体化することができる。Standard LISP の改良にあたっては Standard LISP の C ソースプログラムを合わせて改良を行っている。

4 知識ベース導入

本システムの最初のバージョンでは、ユーザの方からシミュレーションプログラムを作成するのに必要な情報を全部システムに与える形で実現した。(必要な情報が不足していればシステムの方からユーザに聞いてくる。)しかしながら、複数ある解法のなかでも問題の性質によっては、一般的によく用いられるものがあり、使えない解法もある。それはその問題の偏微分方程式群、境界条件群などによっておおよそ分かる。これらから特徴抽出を行い、解法に対する示唆を考えることができる。この知識を管理・推論するのに LISP 言語を使ってプログラミングしている。知識そのものはリストの形で表現している。例えば、次のような拡散方程式の場合

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \quad (4)$$

この方程式の型が知識ベースにあるか否かを調べる。その際に知識ベース内とパターンマッチングができるように式を正規化する。知識ベースの中の表現の仕方としてこの拡散方程式を次のように記述している。

$$\frac{\partial F}{\partial t} + \alpha \frac{\partial^2 F}{\partial x^2} = 0 \quad (5)$$

リストの形になっていて、知識ベースの中に次のように入っている。

```
(diffusion ( (+ (deriv F t) (* a (deriv (deriv F x) x))) = 0)
(convection ( (+ (deriv F t) (* F (deriv F x)) = 0)
(laplace ... )
(.. )
...)
```

5 まとめ

現在研究開発の数値シミュレーションプログラム合成システムの現状について述べた。COMMON LISP から Standard LISP へ移行し、分数型のデータをあつかえるようにした。さらに幾つかの内部関数の改良を行った。知識ベースについては LISP のレベルで知識ベースの持ち方、与えられた問題つまり偏微分方程式の特徴抽出、推論機構など開発を進めている。

参考文献

- [1] 川田重夫, 飯島邦彦, ブンミー・チュンボル, 真鍋保彦: 記号処理手法による数値シミュレーションコード開発支援システム, 第 34 回プログラミングシンポジウム (情報処理学会), pp.61-72 (1993)
- [2] 佐川暢俊, 金野千里, 梅谷征雄: 「数値シミュレーション言語 DEQSOL」 情報処理学会論文誌, 第 30 巻第 1 号, pp.36-45(1988).