

数値シミュレーション支援環境における プログラム生成過程の可視化及びステアリング

Choompol Boonmee, 川田重夫
長岡技術科学大学, 工学部

E-mail: chm@voscc.nagaokaut.ac.jp

現在研究中の数値シミュレーション支援システム NCAS に可視化機能を追加した。数値計算プログラムができるまでの生成過程を効率的に可視化することによってシミュレーション教育やプログラムの検証に大きく役立つ。シミュレーションの検証には、偏微分方程式や境界条件をどのように差分化・変形され、プログラム生成にどのように行われるかの情報が必要である。数値シミュレーションプログラムの理解のためにこれらの情報の可視化と生成されるプログラムの流れ図を PAD 図を用いて可視化する。ユーザが計算のアルゴリズムを PAD 図を理解することにより容易にプログラムが理解できる。又システムではユーザがそれぞれの過程に介入し変更・追加が行え、より柔軟な数値シミュレーションが行えるようになる。

Visualization and Steering of Program Generation Process in Problem Solving Environment

Choompol Boonmee, Shigeo Kawata

Department of Electrical Engineering, Nagaoka University of Technology,
1603-1 Kamitomioka, Nagaoka, Niigata, 940-21, Japan

E-mail: chm@voscc.nagaokaut.ac.jp

In this study, we add the visualization and steering abilities of program generation process into the partial differential equation based problem solving environment NCAS which we have developed. The system enables a user to easily understand the program and to verify logically the generated program. Especially, in the logical verification of the numerical simulation, the user need the information of how it has been generated. The system visualizes the process of the program generation, the result in each process, and the generated program itself. It also enables the user to steer the program generation process. For example, it allows user to select various simulation methods, and to edit difference equations used during program generation.

1. 導入

我々は数値シミュレーション支援環境の研究開発を行っている。特に偏微分方程式を基礎とした自然現象の数値シミュレーションを対象としている。一般的には数値シミュレーションを行うには、偏微分方程式の差分化や差分式の変形整理等様々な多くの仕事をしなければならない。人間がそれを手で行うことは、非効率である。そのため、数値シミュレーションを自動化するいくつかのコンピュータソフトウェアが開発されてきた[1-6]。しかしながら、それらにはまだ多くの改良すべき問題が残されている。

既存のシステムは内部処理を完全にまたは一部をブラックボックス化しており、入力の問題がどのような扱われているのかをユーザに分かるようにしていない。そのため、いくつかの問題が生じる。

1. 行われる数値シミュレーションが本当に正当か否かを論理的検証することができない。もし計算結果が予想外の場合、結果を信じてよいかという根拠がない。
2. 生成されるシミュレーションプログラムを理解することが困難である。シミュレーションプログラム自動生成ソフトウェアの中には、プログラムソースを提供するシステムがある。しかしそのほとんどは可読性が低く、ユーザが理解し難いソースプログラムを出力する。
3. 数値シミュレーションプログラムが生成されるまでの過程が可視化されていないため、その過程の一部を変更したり、差分式の変更を行ったりすることができない。

現在まで、偏微分方程式問題の数値シミュレーションプログラム生成の支援システムである NCAS システムを開発してきた。今回は偏微分方程式と他の諸情報が与えられると、システムがどのような過程を経てプログラムを生成するかをユーザが分かるように可視化する。またそのプログラム生成過程が効率的にグループ化されるため、プログラムの理解やシミュレーションの正当性の論理的検証に役立つと考える。数値シミュレーションはそれぞれの問題に対して、特有の手法が存在する。また同じ解法であっても数値演算の順番を変更するだけで計算結果が大きく違ってくる場合がしばしばある。従ってユーザによる修正・変更できる機能が望まれる。本システムではそれぞれの作業を細かく分離して可視化し、ユーザによる変更・誘導またはステアリング(steering)を可能としている。

2. NCAS システム構成

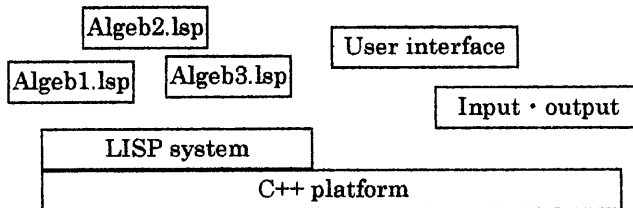


Fig.1.NCAS system platform

NCAS システムの開発プラットフォームとしては C++言語と LISP 言語からできている。

C++のプラットフォームの上に LISP システムを構築した。LISP プログラムの部分では数式処理をし、C++の部分では入出力や画面表示やユーザインターフェースなどの役割をする。

システムの動作を Fig.2 に示す。入力使用ファイル(Input specification file)から問題に関する諸情報を読み込み、差分化や差分式変形など様々な記号处理的な過程を経る。数式処理が必要なプロセスは、LISP のライブラリから適切なプログラムをロードして実行する。問題の偏微分方程式が差分化・式変形がそれぞれのプロセスで行われ最終的に計算プログラムが生成される。

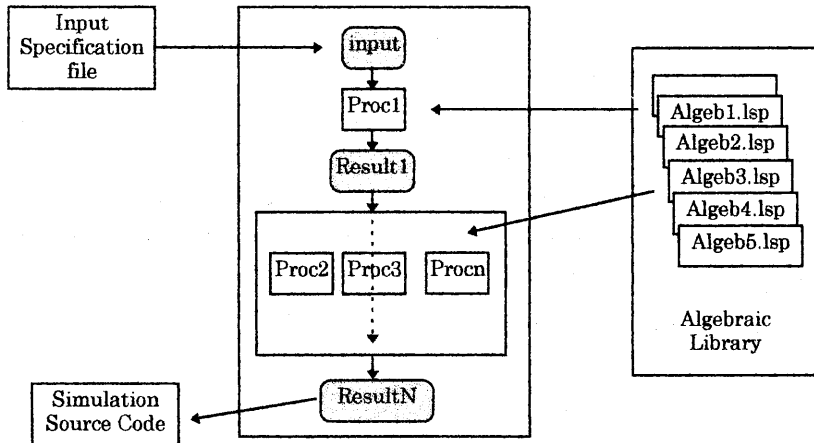


Fig.2.NCAS system.

3. プログラム生成の仕組み

数値シミュレーションプログラムを生成するには、偏微分方程式や境界条件を差分化し、差分式を使用する解法に適切な形に変形して行かなければならない。それらをコンピュータにさせるには記号処理・数式処理機能を持たせる必要がある。NCAS システムでは記号処理や数式処理が得意とされる LISP 言語を用いる。解く変数や数式やプログラムなど全ての情報は LISP の内部のシンボルとリスト表現で実現される。数式を表現するには前置表現 (Prefix form) で実現している。たとえば次のように前置表現で表現されている偏微分方程式があるとす。

$$=(\text{deriv } T \ t) \ (* \ A \ (\text{deriv} \ (\text{deriv} \ T \ x) \ x))$$

ただし、deriv を偏微分演算子と、x,t を独立変数と、T を解く変数とする。この偏微分方程式を次のように差分化される。

$$=(/ \ (- \ (T \ 1 \ 0) \ (T \ 0 \ 0)) \ dt)$$

$$(* \ A \ (/ \ (- \ (/ \ (- \ (T \ 0 \ 1) \ (T \ 0 \ 0)) \ dx) \ (/ \ (- \ (T \ 0 \ 1) \ (T \ 0 \ 0)) \ dx)) \ dx))$$

ただし、(T t x) の t が時間的インデックス、x が空間的インデックスである。同様な過程が繰り返され、最終的に数値計算プログラムが生成される。それぞれの過程を実行する LISP プログラムがモジュール化され Fig2.に示すようにライブラリに格納されている。必要に応じてロードし実行される。それぞれのステップにおける途中結果はアトムと属性リストに格納される。

4. プロセスの可視化

計算プログラムが生成できるまで様々なプロセスが実行される。逐次的に各プロセスを実行すべきプロセスと並列に実行できるプロセスとがある。前者では、前のプロセスが実行され得た結果が出なければ次のプロセスを進められないプロセスである。偏微分方程式が「差分化」というプロセスが実行されなければ差分式ができない。差分式ができないと「差分式の変形」というプロセスも行えない。従って、差分化プロセスを実行した結果を用いて次の差分式の変形プロセスに渡す。

それに対して、複数のプロセスに分岐しそれぞれのプロセスを独立に平行に進めるものがある。たとえば、解くべき変数や偏微分方程式がある場合、それぞれ別々独立の差分化、数式変形プロセスに分けて進めればよい。最後にそれぞれからできた結果をまとめて加工するプロセスに渡す。プロセスの可視化の基本要素として

1. 状態・結果ボックス：各プロセスによって出される処理結果を格納するものである。
2. 処理・実行ボックス：プロセスそのものを表現し、1.の状態・結果を入力とし処理した後にまたその結果を別の状態結果ボックスに入れる。

これら2つのボックスがコンピュータ画面に次の Fig.3 のように現れる。

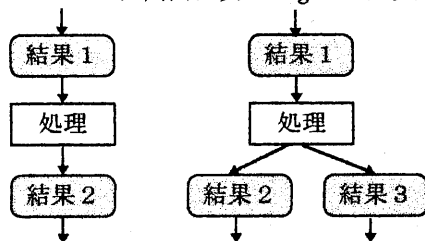


Fig3. 基本要素の組み合わせでプロセス可視化の実現

また、複数の処理からなる複雑なプロセスは、グループ化し分かりやすくする。グループ化により処理実行の効率的な可視化が可能になる。

5. プロセスの可視化

本システムは基本的に偏微分方程式問題を対象とし数学モデルという抽象的なレベルから計算プログラムの設計・生成をするので、プロセスの可視化の他にユーザの理解に役立つ様々な情報を可視化できる。そのため、さらに上記のグループ化もより効率的に行える。本システムで可視化している情報としては次のようなものがある。

1. 解くべき変数、位置定義：変数がどんな物理量を現し、メッシュのどこに定義されるかの情報を可視化する。
2. メッシュ・境界定義：使われるメッシュ及び境界条件が定義される場所の可視化。
3. プロセス中のコメント：プロセスの流れの理解に役に立つようなコメントが自動的に生成され、可視化される。
4. 生成されるプログラムの流れ図の可視化：本システムでは自動生成される計算プログラムを可視化する。ユーザが理解しやすいように PAD 図を用いて可視化している。
5. プログラム生成の各過程における数式結果の自然な表現による可視化。

6. プログラム生成のステアリング

数値シミュレーションでは、一般にはそれぞれの問題に敵した解法が存在する。問題の性質によって適用するべき解法が異なる。また同じ解法を使って計算しても同じ結果が得られると限らない。数値計算を行うので、同じ計算式であっても計算順番が異なれば結果が違ってくる場合がある。従ってよいシミュレーション支援環境では、ユーザに様々な解法を自由自在に選択できる機能や細かい計算操作の変更できる機能をもつべきである。本システムではこういったようなことを可能にする。使いたい数値解法を選択でき、プロセスの細かい部分での変更ができる。

7. 実行例

実行例の画面が下に示す。図の問題では解く変数が T_i と T_e の2つある。最初は各解くべき変数に関する情報：物理量定義、メッシュ上の位置定義、偏微分方程式・境界条件などが可視化される。さらには偏微分方程式・境界条件が差分化など数式処理される過程とその途中結果を可視化している。また自動生成される計算プログラムの PAD 図による可視化も行われる。

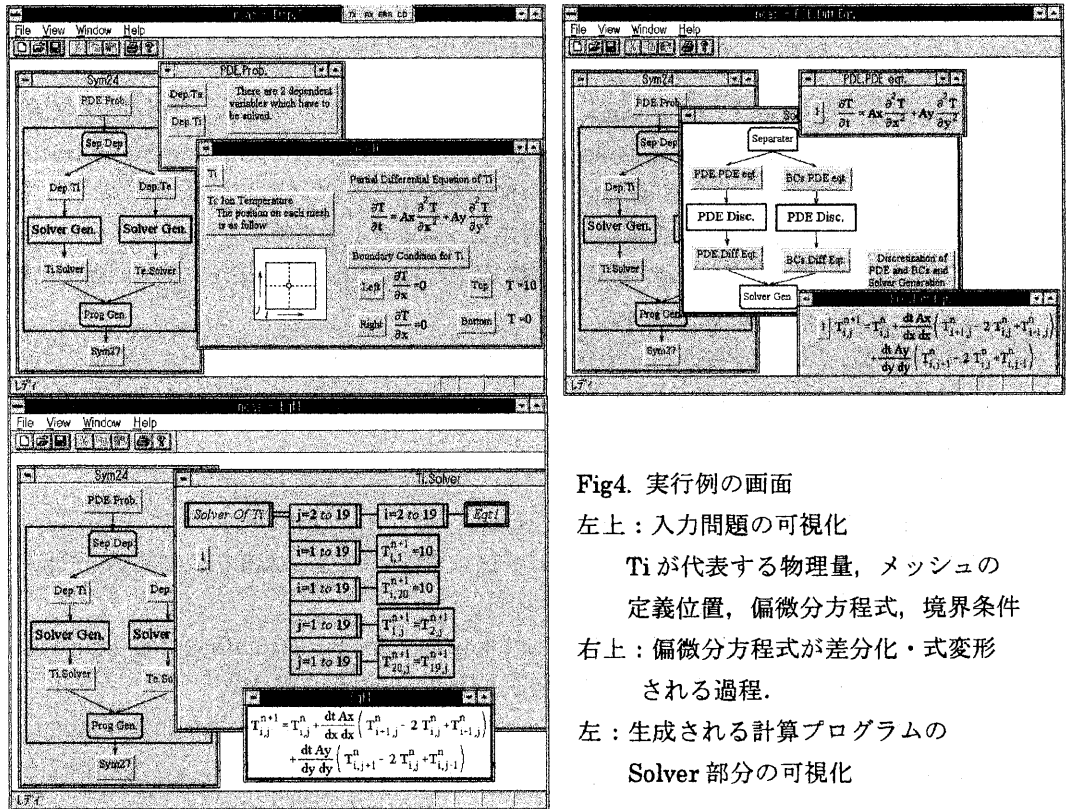


Fig4. 実行例の画面

左上：入力問題の可視化

T_i が代表する物理量，メッシュの定義位置，偏微分方程式，境界条件
 右上：偏微分方程式が差分化・式変形される過程。

左：生成される計算プログラムの Solver 部分の可視化

8. まとめ

従来の数値シミュレーションシステムでは、シミュレーションの正当性の論理的検証やシミュレーションプログラムの理解が非常に困難であった。本研究で開発された数値シミュレーション支援環境システムに可視化機能を追加した。数値計算プログラムができるまでの生成過程を効率的に可視化することによってシミュレーション正当性の論理的検証に大きく役立つ。

シミュレーションの論理的検証を行うために、偏微分方程式や境界条件をどのように差分・式変形され、プログラム生成にどう使われるかを細かく可視化する。又グループ化して可視化する。

数値シミュレーションプログラムの理解には、生成されるプログラムの流れ図をPAD図を用いて可視化する。ユーザが計算のアルゴリズムをPAD図を理解することにより簡単にプログラムが理解できる。

システムは完全に自動化するのではなく、プログラム生成過程が細かく分離し、ユーザがそれぞれの過程に変更を加えたり、追加処理したりできる。これにより柔軟な数値シミュレーションが行えるようになる。

参考文献

- [1] S.Kawata, K.Iijima, C.Boonmee and Y.Manabe: Computer-assisted scientific-computation / simulation software-development system -including a visualization system-, IFIP Transaction A-48, pp.145-153 (1994).
- [2] C.Boonmee, 真鍋保彦, 渋井俊昭, 釣谷浩之, 森正孝, 川田重夫: 数値シミュレーションプログラム生成支援に関する研究—差分法プログラム生成支援—, 情報処理学会研究報告, HPC-53, pp.1-8, (1994).
- [3] G.O. Cook, JR, ALPAL: A Program to Generate Physics Simulation Codes From Natural Descriptions. International Journal of Modern Physics C Vol.1, No.1, 1990, pp.1-51.
- [4] S.Weerawarana, E.N.Houstis, J.R.Rice, A.C.Catlin, C.L.Crabill, C.C.Chui, and S.Markus: DDELAB: AN OBJECT-ORIENTED FRAMEWORK FOR BUILDING PROBLEM SOLVING ENVIRONMENTS FOR PDE BASED APPLICATION., Purdue University Technical Report CSD TR-94-021, 1994.
- [5] A.F.Cardenas and W.J.Karplus: PDEL-A language for Partial Differential Equations, Communication of the ACM, Vol.13, No.3, March 1970, pp.185-191.
- [6] Y.Umetani: The Visual Diagnosis of The Numerical Calculations for PDE Problems. Purdue University Technical Report CSD-TR-92-082, CAPO Report CER-92-38, October 1992.