

数理計画のためのモデリングツールの開発

山下 浩

1. はじめに

オペレーションズ・リサーチの分野あるいは今回のテーマである最適化に限らず広く数理的な方法で問題解決を目指す分野において、対象の的確なモデリングは最も重要なテーマの一つである。私自身はもっぱら最適化のアルゴリズムの研究、あるいはそれを実際問題に応用するためのソフトウェアの作成にかねてより興味があった。私の会社では数理計画に限らず数理工学的手法を現実の問題解決に適用するためのソフトウェアを開発して実際の問題解決にあたる、あるいは、そのソフトウェアを広く利用してもらうということをビジネスにしている。この分野で数少ない日本製のソフトウェアであるが、何とか国内のみならず広く海外にも普及させたいと常日頃思っていて活動している。そのような中で実際のモデリングに関しては、具体的な対象を前にしてそのつど言わばアドホックに工夫を重ねるのがせいぜいであった。したがって、もちろんそれなりのノウハウの蓄積はあるものの、最適化のモデリングについて他人に講釈するほどの体系だった知識は持ち合わせていないのであるが、上記ソフトウェアを開発していく過程で否応なくモデリングのためのツールについて真剣に考えることが必要になった。以下で、若干の体験を記したい。

我々の開発したモデリングツールの具体的な内容は後半に譲って、前半では個人的な歴史を交えてツール開発に至る経緯を記したい。強調したいのは、このようなツールの実現・実用化に至るまでには様々な関連技術・周辺状況の成熟・進歩が不可欠ということである。この事情を理解してもらうために少し回り道をしながらか話を進めたい。

2. モデリング言語の必要性

筆者が最適化のアルゴリズムに興味を持って、自分なりのアルゴリズムを考えたり、そのプログラムを作ったりし始めた頃は、「数値計算はFORTRAN」の時代であった。非線形のモデルを計算機に教えるためには、関数の非線形部分を目的関数と制約条件の数だけ計算するプログラムを具体的に書いてやる必要があった。偏導関数が必要なアルゴリズムには上記の関数一つずつに対して変数の数だけ偏導関数を書かなくてはいけない。2階偏導関数が必要な場合にそれを間違えなくコーディングすることは超人的な能力を要する作業である。

線形計画問題は基本的に係数行列を何らかの形で定義すれば良いので、事情はもう少し簡単だった。MPSファイルという形式で係数データを定義してやれば基本的に問題定義はできたことになる。しかし、この場合も問題（モデル）全体の構造などが一見して分かるようにはなっていない。データファイルを見ただけでどんなモデルか理解できる人は稀であろう。

前述のような状況が（少なくとも私の周辺では）1990年頃まで続いていた。私の会社で出している数理計画パッケージNUOPTも版を重ね、内点法の台頭もあって大規模な非線形最適化問題が現実的ターゲットになってきたところなのに、数理計画のモデリングについては旧態依然という状況であった。何とかしたいと思っていた頃、海外の最適化のコンファレンスでAT & Tベル研究所のDavid Gayがモデリング言語AMPLについて発表するのを何回か聞く機会があった。最初は代数的モデリング言語（algebraic modeling language）という物珍しい名称に惹かれて聞いてみただけであったが、これはなかなか面白いテーマだということが分かってきた。調べてみると私が無知なだけで、GAMSというモデリング言語は1980年頃から提案されていたということだった。大規模問題を扱うために集合とその要素の概念を採用したり、モデ

やました ひろし
㈱数理システム
〒160-0022 新宿区新宿2-4-3

ルとデータを分離してモデルの汎用化を図るあたりが鍵ということも分かってきた。

面白いことに1980年頃提案されたGAMSはFORTRANそっくり、80年代後半に提案されたAMPLはCのフレーバーが一杯である。AMPLについてはベル研のグループによるもので、さらに設計者の一人にCの設計者のKernighanが加わっているので当たり前といえば当たり前だろうか。

3. 高速自動微分

話を非線形モデルの偏導関数の計算に戻す。これを機械に教えることが実際の応用においてバリエーションになっていることは先に述べたとおりである。この流れの中での技術革新は高速自動微分の技術であろう。私がこの方法の存在を初めて知ったのは、1980年代初めの数値解析シンポジウムの会場であった。このシンポジウムは毎年1回3日間の日程で一つの会場に出席者全員が泊り込んで数値解析の様々な分野の発表を行うという楽しい集まりである。この会場で伊理先生から立ち話でこんな新しい技術があるということを知ることができた。私の最初の反応は「これで準ニュートン法が消えてニュートン法が広く使われるようになるかもしれない」というものだった（私は準ニュートン法の無用とも思える複雑さと、延々と続く更なる複雑化を目指したとしか思えない研究があまり好きではなかった）。ニュートン法が広く使われるようになるかもしれないという私の予想は確かに当たったが、この技術が広く使われるようになるにはかなり時間がかかった。

この後、伊理先生と共同研究者たちの研究発表や論文を参考にして、私も少しかじってみた。この技術の研究者のインプリメンテーションは私の知る限りすべてFORTRANなどで書かれた関数計算プログラムのソースコードから関数と偏導関数を計算するコードを自動生成するというものだった。そこで、私もバイトの学生にこの技術を説明して同様なことをする簡単なプログラムを試験的に作ってもらった。ところが、ほんの数行の関数計算プログラムから相当大きな導関数計算プログラムが生成されてしまうことを報告されて、その当時（1985年ごろ）の計算機環境ではあまり実用にならないと思い、早々に開発を止めてしまった。ただ、最適化に限らず非線形モデルの実用的利用にこの技術が不可欠であろうと思い、伊理先生のグループの研究には常に興味を持っていた。そのうち、海外でも広く高速自動微分法が研究されるようになった。

4. モデリング言語の構想と実現

話を再度モデリング言語に戻す。もう一つ1980年代後半に私が興味を持っていたのはいわゆるオブジェクト指向言語である。今では当たり前の技術であるが、当時は新しかった。Smalltalkの統一的世界の美しさに惚れ込んで私なりに随分勉強した。社内で勉強会のグループを作り新しい言語を設計するという、今考えればかなり無謀な企てを数年間続けたこともあった。その流れの中で当時現れたC++に興味を持ったのは自然な成り行きであった。この言語はプログラミング言語としてはあまりスマートとは言えず、色々問題もありそうだったが、設計者のStroustrupの教科書を読むと色々な可能性が見えてきた。Stroustrup自身は明示的に書いてはいないが、次で述べるような利用法を念頭に言語のある種の機能を設計したのは（少なくとも私には）明らかに思えた。

私の構想は、次で述べるSIMPLEにおいて数理計画のためのモデリング言語と高速自動微分の両方をC++の機能で実現しようというものだった。たしか1994年のゴールデンウィークの休日中に10ページ程度のモデリング言語の構想を書き上げた。細かい部分は別にして、現在のSIMPLEの機能はほとんどそのメモに書かれていたように思う。そのメモにあってまだ実現されていない機能もある。

その後、社内のチームで細かい設計とインプリメンテーションを行った。優秀な開発チームの頑張りによって1994年の末には α バージョンが動き始めた。外部に発表したのは1994年から95年にかけてOR学会の研究部会が最初だったが、その集まりの一つに伊理先生と久保田先生が出席され、我々の結果を詳しく聞いて頂けたのは幸いだった。伊理先生に最初に自動微分法のことを伺ってから10年以上経っていたが、私としては遅れていた宿題を提出した気分だった。また、数理計画のモデリング言語で高速自動微分法を実装したのは我々が最初だったと理解している。

5. モデリング言語SIMPLEの特徴

モデリング言語はプログラミング言語ではないとはいえ、計算機にある種の情報を教える手段である。プログラミング言語との重要な違いは、後者が計算機の実行する手続きを指定するのに対して、モデリング言語は単に対象としている数理計画モデルの内容を伝えるという点にある。多くの場合数理計画モデルは数式

で表現されている。したがって、我々の方針はなるべく数式に忠実な言語にしたいということだった。その点で既存のモデリング言語には不満があった。

次に SIMPLE の設計方針の主なものを簡条書きにしてみる。

- (1) システムの要素の数学的関係が自然な形で記述できる。
- (2) 大規模モデルを簡便に記述できる。
- (3) モジュール化・階層化記述ができる。
- (4) システムの解析法に関してユーザが意識する必要がない。
- (5) 新しい解析プログラム (ソルバ) とのリンクが簡単にできる。
- (6) システムに対する細かい調整が簡単に指定できる。

これらの中で(1), (2), (3)あたりが最も重要なポイントであろうか。大規模モデルの記述は他のモデリング言語でも採用している集合とその要素が利用できれば良い。その他の点に関しては色々な工夫が必要であった。

ここで、インプリメンテーションの話に戻るが、このような言語の実装は専用のインタプリタを作るのが普通であろう。この方法でも特に問題はなかったのだが、高速自動微分の実装、数式記述、任意の前処理・後処理の手続きが書けるといふ点から C++ を利用することにした。C++ には演算子のオーバーディングという機能があって、すべての演算子の動作を指定することができる。この機能を使うと、ユーザがどんなに複雑な数式を書いてもある意味で自動的に高速自動微分に必要な計算グラフを作成することができる。計算グラフだけではなく、モデルの様々な情報を演算子経由でシステムに教えることが可能になる。基本的には、モデルを記述したプログラムを 1 度実行するだけで、すべての情報が演算子経由で SIMPLE システムに蓄えられる。その後、SIMPLE システムは最適化のソルバプログラムと情報のやり取りをしながら、最適化を実行することになる。

6. SIMPLE で書かれたモデルの断片例

簡単に SIMPLE を使ったモデルの雰囲気を変えたい。変数やパラメタの指定は省く。数式を見れば見当が付くであろう。以降の断片はすべて C++ のプログラムである。例えば、

$$-0.5 < \sin(x) + \cos(y) \leq 0.5$$

という不等式制約条件があったとすると、これは

$$-0.5 \leq \sin(x) + \cos(y) \leq 0.5;$$

と書けばよい。sin(x), cos(y) の中身も SIMPLE が定義しているが、演算子+が実行されたときに sin(x)+cos(y) という情報が計算グラフとして内部に蓄えられる。不等号の演算子<=もそれが実行されることによって、上記の不等式制約条件の存在がシステムの知るところとなる。上記のプログラムを一度だけ実行すればシステムに必要な情報が伝わることになる。

さて、x とか y という少数の変数ではなく、多数の変数を扱う必要があるときには、それらは大抵グループ分けされてそれぞれのグループの変数は添え字で区別されることが多い。添え字 i が集合 S の要素、すなわち $i \in S$ となるとき

$$-0.5 \leq \sin(x_i) + \cos(y_i) \leq 0.5$$

という不等式条件が書かれると、任意の $i \in S$ に対して、この条件が存在すると解釈するのが普通であろう。すなわち、i の個数だけ上記条件が存在する。SIMPLE で

$$-0.5 < \sin(x[i]) + \cos(y[i]) <= 0.5;$$

と書くと、集合 S 内のすべての i に対して上記条件が成立すると解釈される。プログラミング言語ではこのような場合 i に関するループで表現するが、SIMPLE では一行の式で済むことに注意されたい。

上記の不等式がある添え字 $i \in T$ に対してのみ成り立つとき

$$-0.5 \leq \sin(x_i) + \cos(y_i) \leq 0.5, i \in T$$

のように書くことが多いが、SIMPLE でも同様な書き方ができる。

$$-0.5 < \sin(x[i]) + \cos(y[i]) <= 0.5, i < T;$$

ASCII 文字列には集合の要素を表す \in に対応する適当な文字がないので、< を使っている。また、上の書き方で肝心なのは、(コンマ) が C++ において演算子として扱われていることである。コンマに対応する演算子関数を定義することによってシステム内部で適切な処理がなされる。

7. 数式によるモデル記述

SIMPLE によるモデル記述には多くの機能があり、相当自由度の高いモデル記述が可能である。また、集合の概念を活用することによって簡潔なモデル記述から大規模なモデルを生成することが可能であり、多くの実際的な問題が解かれている。例えば、ある生産工場全体の最適化モデルでは様々な物理的・化学的關係を

記述した数多くの部分モデルからなり全モデルが1万ステップにもなる非線形モデルが実際に記述され最適化されている。モデリングツールを提供している側の我々が驚くような使い方が現場でなされているのである。

さて、モデリング言語は機能が豊富で実用的とは言っても、プログラミング言語であるのでその使用にはやはりバリエーションがある。私がかねてから実現したいと思っていたモデリングツールは、教科書などに書いてある数式モデルがそのまま実行されて最適解を得られるような環境である。当初、SIMPLEを構想したときにそのような環境に利用できるように設計したと言っても良い。すなわち、数式を何らかの方法によって自動的にSIMPLEに変換できることを想定した。

数式を入力するツールは既存のものが数多く存在する。それらの中からWindowsのWORDの中で動くMathTypeを採用した。1998年末にMathTypeを開発しているDesign Science社にこういうことをやりたいのだが、環境は整っているかという質問をメールでしたところ「現在開発中で近い将来リリースする」という返事が返ってきた。2002年ごろに調べたところ、どうも環境が整ったらしいということで、2003年から開発を開始して2004年にMP.docというソフトが完成した。その方法は、MathTypeのメニュー経由で入力した数式モデルをMathTypeの組み込み機能によってTeXに変換してもらい、その後TeXで書かれたモデルをSIMPLEに変換するという仕組みである。

最適化を実行した答えも数式の形でWORDドキュメントの中に書くことができる。したがって、WORDドキュメントの中で対話的にモデルを変更するとその答えが自動的にドキュメントの中に現れる。その意味で、WORDがライブドキュメントとなる。現在EXCELの表やグラフとも連携すべく開発中である。この特集が刊行される頃には完成していると期待している。先に例として挙げた数式や次のような複雑なモ

デル（フィッティングの問題）もそのまま実行することができる。

problem

Set $Samples, T$

Variable σ, γ, β

Index $i \in Samples, j \in T$

Variable e_i

Parameter x_i, y_i, rp

Expression A, b_j

$A := 10\beta - 12\gamma^2 - 18$

$b_0 := -\sigma^2(4\beta - 3\gamma^2)/A$

$b_1 := -\sigma\gamma(\beta + 3)/A$

$b_2 := -(2\beta - 3\gamma^2 - 6)/A$

$x_i := x_i - rp$

minimize $\sum_i e_i^2$

subject to

$$e_i = \left(\frac{1}{2b_2}\right) \log|b_2x_i^2 + b_1x_i + b_0| - \left(\frac{2b_1 + (b_1/b_2)}{(\sqrt{4b_0b_2 - b_1^2})}\right) \tan^{-1}\left(\frac{2b_2x_i + b_1}{(\sqrt{4b_0b_2 - b_1^2})}\right) - y_i$$

8. おわりに

我々の開発しているモデリングツールは、当面は数理計画のためのモデリングを主たるターゲットにしているが、最終的には数理的なモデリング全般やシミュレーションなども視野に入れて行きたいと思っている。実験的には有限要素法を利用した解析、常微分方程式系の解析などに利用したこともあるが、これらの点についての完成はまだ遠い道のりであろうか…。

参考文献

- [1] 山下, 田辺, 逸見: “数理計画のためのモデリング言語 SIMPLE I 概要”, OR 学会 1997 年春季研究発表会アブストラクト集.
- [2] 高橋, 山下: “数式による記述で数理計画問題を解く試み”, OR 学会 2003 年秋季研究発表会アブストラクト集.