

数理モデルの構築と運用のための会話型計算システムの開発

— 環境拡散モデルを例として —

Development of a Interactive Numerator to make and use Simulation Models

— With an Illustration of an Environmental Diffusion Model —

槌田 博・加藤 龍夫

Hiroshi TSUCHIDA and Tatsuo KATOU

Synopsis

Complicated programing has been necessary to get a result of mathematical simulation models. But, NIFE (Numerator of Itemized Formula Expressions) is a new tool to make and use them on MS-DOS. NIFE extracts results of values, tables, and graphs from ordinary itemized formula expressions and vector-matrix operations, of which a simulation model is made. NIFE has performance of calculating, listing, searching, modifying, and saving them. And also, NIFE has external librarys like numerical analysis of differential equation and method of least squares. So, anybody may briefly operate a model with NIFE.

1. 緒 言

環境汚染の解析や予測に数理モデルを使う試みは、活発に研究が進められている分野である。スーパーコンピュータを駆使して複雑な地形での微気象まで考慮した汚染予測が試みられる一方で、近年急速に進歩したパーソナルコンピュータの能力で充分に実用的な結果が得られる数理モデルも少なくない。現在のパーソナルコンピュータの普及率はめざましく、誰でも容易にアクセスできる環境が整ったとみてよい。数理モデルによる環境汚染の把握を一部の研究者だけのものから、もっと広範囲の人が扱えるものに普及していく時期が来ているといえる。

環境汚染の数理モデルは、単位過程（物質移動、生物過程、物理過程など）の組合せでセル（コンパートメント）の物質収支を記述し、そのセルを組み合わせでシステム全体の様子を記述していくような階層構

造¹⁾を持つのが一般である。このようなシステムを説明するときに、全体像から順に細かい部分を説明していくトップダウンの記述方法がより判り易くなる。しかし、これを計算プログラムに直す過程で問題が起こる。従来の高級言語は、計算機に計算させる順序を記述するために設計されているために、論文中に現れるような箇条書にされた数式をそのまま計算することができない。計算するためには、下位の計算要素から求めるべき最上位の計算要素に至る計算順序の道筋をコンピュータープログラマーが指定しなければならない。これは、論理の順番を逆にしてボトムアップな記述に直すことに相当し、多くの努力が必要であった。ここに、計算の順序を意識しないで、計算機を利用できるようなツールの必要性が生まれてきた。オブジェクト指向プログラミングというのがその解決方法として提示されているものである。

数理モデルの結果は、常に実際の環境測定結果で検証されなければならない。検証の結果として不合理な結果が見いだされたときに、その原因がモデルのどこにあるのかを調べなければならない。しかし、従来的高级言語で数理モデルを記述してしまうと、計算途中のパラメータ（計算要素）の値を表示したり、計算モデルの一部分を仮に交換してみたりといった作業は容

* 横浜国立大学 環境科学研究センター 環境基礎工学研究室

Department of Environmental Engineering Science,
Institute of Environmental Science and Technology,
Yokohama National University
(1990年12月1日受領)

易ではない。ここに、即座に必要な値を操作できるような会話型のツールの必要性が生まれる。

数理モデルを記述する上での従来の計算環境での問題点を表1にまとめた。これらの問題を解決するために、新しくNIFE (Numerator of Itemized Formula Expressions)を開発した。

2. 計算環境 (NIFE)

2.1 可読性の高い記述方法

2.1.1 計算要素の名前

計算要素は、関数、変数、行列型関数、行列である。これらの名前は、英字、カタカナ、漢字、ギリシャ文字、ロシア文字、数字 (半角の数字だけは名前の1文字目に使用できない)、演算子ではない記号を組合せて任意に記述する。大文字小文字、全角文字半角文字は区別される。このことによって、計算要素のほとんどを学術論文中にあらわれる慣用的な名前で記述することができる。

2.1.2 数値の記述

NIFEの計算環境では、演算はすべて倍精度浮動小数点で行われる。このようにすることによって、従来の計算機言語で可読性を著しく損傷していた型宣言を省いた。入力される数値の記述方法は、整数 (123)、小数 (123.45)、指数表示 (123.45E-67) のいずれも許される。指数表示のEは10の累乗を示している。

2.1.3 計算式の計算

数理モデルに含まれる要素の値がいつでも即座に取り出せる事を重視した。計算要素や数値を含んだ計算式を入力すると、直ちに計算が行われてその値が表示される。このとき定義式に単位の記述があれば、同時に単位も表示することで利用者の便宜を図っている。

2.1.4 関数の定義

最も基本になる計算要素が関数である。その定義の記述は、例えば次のようになる。

> C(t)=C0 exp(-k t) '汚染濃度[ppm]
まず、「C」が関数名であり、必要に応じて引数「(t)」

が記述される。等号「=」が関数の定義であることを示す。そのあとに計算式「C0 exp(-k t)」が記述される。計算式に於て、従来の高級言語では乗算の記号「*」が不可欠であったが、NIFEでは可読性を高めるため空白で区切れば乗算記号が省略できる。計算式の演算は関数定義の段階では行われないので、計算式中の計算要素 (C0, k, t) は、任意の位置で定義されていれば充分でありボトムアップである必要はない。以下必要に応じて、「'」で始まるコメントで定義の説明を記述する。コメント内に「[ppm)」のような単位の記述があれば、計算結果の表示の時に一緒に表示される。単位の表示は、利用者の理解を助け、誤解を防ぐための重要な機能である。

2.1.5 行列ベクトルの記述

行列やベクトルは、数理モデルの各コンパートメント毎の計算要素の記述や不規則な時間変化を伴う気象条件の記述には欠かせないものである。目的に合わせて、以下の数通りの定義方法が用意されている。

(1) 要素並びによる定義

$$A[3,3] = \{1, f(2,2), 3, 4, 5, 6, 7, 8, 9\}$$

(2) 添え字の計算による定義

$$B[3,3] = [i,j] i+j+f(i,j)$$

(3) 行列計算による定義

$$C[] = A + A \cdot B + \text{inv}(B)$$

(4) 関数 faa(x), fal(x)...fww(x)による定義

$$D(x)[alw,alw] = f$$

(1)から(3)の方法では、行列ベクトル定義の時点で行列要素の値を計算してしまうので、定義後に下位の計算要素 (この例では f(x,y)である) を修正しても行列要素の値は変化しない (計算時間の短縮のため)。このため、定義式の通りに再計算する「.matrix」命令が別に用意される。

(4)の方法は、関数型の行列を定義する方法である。この場合は行列Dを参照する度に改めて関数 f??(x) (?は alw のいずれか) を計算するので、この関数型行列の定義以後に下位の計算要素を修正したとき、D

表1 コンピューターによる代表的な計算環境

電子計算機言語	特徴	問題点
大型計算機 Fortran DDSL,SDS (DYNAMO)	高速、豊富な演算ライブラリー シミュレーション専用言語	予測式の変更に柔軟に対応できない。 操作が容易でない。予測式の構築後に有効。
パーソナルコンピューター BASIC Fortan,C 表計算	プログラム修正が容易 高速 修正が容易、再計算	ボトムアップな記述。見通しのわるいプログラム。 ボトムアップな記述。修正から実行までが煩雑。 変数の扱いが面倒。関数が引数を持たない。

の行列要素の値は自動的に変化する。

2.2 モデル構築支援

2.2.1 表示命令

表示関係の命令には、定義の一覧を表示する .list 命令、ある計算要素を使用している上位の計算要素を調べる .select 命令、ある計算要素の計算の中で値を参照されている下位の計算要素を調べる .reference 命令がある。例えば、

```
> C(t)=C0 exp(-k t) '汚染濃度 [ppm]
```

```
> k=k1+k2+k3 '減衰速度係数 [1/h]
```

である場合、Cは、kに対する上位の計算要素、k1,k2,k3は、kに対する下位の計算要素である。数理モデルが単純なら .list 命令だけで十分であるが、複雑になり定義の数が増えるにつれて .select 命令や .reference 命令で表示する定義を絞り込むことで思考を支援する。

.watch 命令は、重要な計算要素を指定して、その値をディスプレイ画面の上部に常時表示しておく。この機能により、下位の計算要素の修正や計算式の変更による計算要素の値の変化をシミュレートする。

2.2.2 制御命令

シミュレーションを繰り返したり、計算結果を一覧表にして表示したり、行列ベクトル演算を繰り返したりするために、.do, .if, .for, .while, .repeatなどの制御命令が用意されている。これらの命令によって様々なマクロ命令を手続きとして利用したり、外部の手続きファイルを読み込んで実行したりする。外部の手続きファイルはライブラリとしても整備されつつあり、現在、「最小自乗法」、「ルンゲ・クッタ法による常微分方程式の数値計算」などが手軽にNIFEから利用できるようになっている。

2.2.3 変数の代入

制御命令を有効に活用するためには変数の利用が不可欠である。NIFEでは、変数の代入には、代入記号「<」を用いる。従来の高級言語では、

$$N=N-1$$

という代入文が許されたが、NIFEでは再帰演算の定義になってしまう。代入記号をつかい、

$$N \langle N-1$$

の様に記述する。

2.2.4 グラフの描画

計算結果を折れ線グラフで示すために、.draw, .from, .toなどのグラフ描画命令がある。たとえば、

```
> .draw i,0,100,10i,F(10i)
```

という命令は、xを0から100まで変化したときのF

(x)という関数の値をグラフに描画する。グラフの表示位置の指定 (VXmax, VXmin, VYmax, VYmin), グラフの論理座標の指定 (Xmax, Xmin, Ymax, Ymin), 座標軸の描画 (.coord), 色の指定 (.color), グラフの削除 (.cls) など必要な命令が揃えてある。

グラフ上の1点に印を付ける .dot 命令は、実測値の表示や規制値の境界を示すのに用いられる。

また、描いたグラフ上の点をマウスでクリックしてその論理座標値を予約変数 (Xmouse, Ymouse) に取り込み、これらを表示したり計算したりすることで、グラフの結果の検討を深めることができる。例えばグラフから、最大汚染濃度を読み取ったり、基準値まで減衰するのに必要な時間を求めたりするときに有効な機能である。

2.2.5 計算の途中結果の監視

数理モデルが複雑になると、有り得ない結果が出力されて、その原因が判らなくなることがある。そのような場合には、.set trace on 命令を用いると計算途中の全ての計算要素の値を逐一表示するので、モデルの不適当な部分を発見しやすくなる。

2.3 モデルの修正支援

2.3.1 関数の修正

既に定義されている関数と同名の関数を定義すると、

```
> C(t)=CO exp(-(k+k3)t)+I
```

```
old: C(t)=CO exp(-k t) '汚染濃度 [ppm]
```

```
new: C(t)=CO exp(-(k+k3)t)+I '汚染濃度 [ppm]
```

の様にコメントを引き継ぎながら定義が入れ替わる。このように数理モデルの変更修正がいつでも容易に出来ることは必須要件である。

ある計算要素の修正によって、それより上位の計算要素の値も当然変化する。(ただし、行列の要素だけは .matrix 命令の後に変わるので注意が必要である。) この機能を使うことで、モデルや条件を様々に変化させたときの結果を随時シミュレートし互いに比較することができるのである。

既に定義されている計算要素の一部分だけを修正するのに、その定義の全体をもう一度初めから入力しないのは不便である。そこで .edit 命令を使い、指定した定義をディスプレイ上に呼び出しフルスクリーンモードで修正して再定義する。

2.3.2 行列の要素への代入

行列には気象データなどを入れることが多い。シミュレーションの都合で部分的に値を変更してみたい場合がしばしば発生する。その場合には、以下の方法が用

いられる。

(1) 全体の代入更新

```
A[] <A + 5 E[3,3]
```

行列ベクトルの計算をしてその結果をすべて代入する方法である。この例では、行列Aの対角成分に5が加えられる。代入記号「<」を使い、再帰的な行列計算式も許される。

(2) 要素を指定して代入

```
A[1,*] <{g(4)+g(1/4),45,0.25}
```

行列の行とか列の指定に「*」を使うことで、特定の列や行に代入する。[*,*]とすると全体であり、[0,0]であれば唯一つの要素を指定することになる。代入記号「<」の後ろは、行列要素の並びであり、「g(4)+g(1/4)」の様に計算式でもよい。計算式は即座に計算されて答えの値が代入される。

(3) .input 命令

```
.input A[1,0]
```

とすると、A[1,0]からA[2,2]まで順に値をひとつずつ会話モードで代入するようになる。NIFEが順番に

```
I A[1,0] {現在値} <{ }
```

のように尋ねてくるので、利用者は {} 内に代入したい値もしくはその計算式を記入する。利用者が改行キーを押すと、次のA[1,1]に移り、最後のA[2,2]まで入力が続く。

2.3.3 定義名の変更

小さい数値モデルから、徐々に修正と追加を繰り返して大きな数値モデルを構築していく過程で、計算要素の名前をまとめて変更する必要があることがある。.rename 命令により、指定した計算要素の名前を別な名前にすべての定義について書き換えることができる。

2.3.4 定義の削除と全削除

不要になった定義は、.del 命令で削除する。.new は、全ての定義を削除する。

2.3.5 定義の保存と読み込み

記憶している定義は、.save 命令でディスクファイルに書き出して、保存する。これは、.do 命令によって後日読み込んで使用できるので、長期間にわたって数値モデルの計算や修正を繰り返すことができる。さらに、これによって書き出されたファイルの書式はMS-DOSの標準テキストフォーマットであるので、エディタやワープロを初め様々なユーティリティで加工・再編集・利用することができる。関数の可読性が極めて高いので、その数値モデルの解説を随所に書き

加えるだけで報告書になる。この報告書は、そのままNIFEに読み込ませて値の計算をさせることが可能である。報告書の中で、プロンプト「>」が先頭に付いている行だけをNIFEが理解し、その他の行は説明と見なされ、読み飛ばされるようになっているからである。

2.3.6 結果の保存

NIFE実行中の計算結果を、.set data to 命令でディスクファイルに書き出すように指定する。これもMS-DOS標準テキストフォーマットなので、他のプログラムで容易に再利用される。全ての実行過程は、.set log to 命令でディスクファイルに記録されるので、あとで修正の過程や計算値を検討するときに使う。

2.4 初心者から熟練者まで対応

2.4.1 動作環境

日本国内で最も普及しているパーソナルコンピュータ日本電気PC9801とその互換機のMS-DOS上で動作する。また、NIFE自身は、ANSI準拠のC言語で記述されているので、他機種への移植は困難ではない。

2.4.2 アシスト

NIFEの既設定値では、ディスプレイの最上段の2行には、利用者が使おうとしている命令についての解説が常に表示されている。利用者は、この行を見ながら命令に必要なパラメータを記入していくことができる。さらに、その下5行にはフルスクリーンモードで入力しているカーソルなどの制御キーについての一覧表が示される。カーソル制御キーは、カーソルキーのほかに標準的なCTRL+ダイヤモンドキーでも操作する。熟練者は、これらのアシスト2行とカーソル制御一覧5行を、.set assist 命令で非表示にすることができる。

2.4.3 ヘルプ機構

.help 命令で、NIFEについての様々な情報を表示する。命令の解説やNIFEの使用例などを見ることができる。また、ヘルプ情報の書かれているファイルもMS-DOS標準テキストフォーマットであるので、必要に応じてヘルプ情報を書き換えたり、ヘルプファイルを切り換えたりすることで様々な要求に対応している。

このヘルプ機構はハイパーリンクのようなページ制御機能を持ち、利用者は関連項目の解説を縦横無尽に参照していくことができる。

2.4.4 ファンクションキー

PC9801のキーボードにある10個のファンクションキーとSHIFTキー、GRPHキーを組み合わせるこ

とにより、N I F Eのほとんどの制御命令がファンクションキーから入力でき、制御命令のメニューの役割を果たしている。

3. 計算例

煙突から排出された汚染物質の濃度の予測式を例に用いて、N I F Eの機能を解説する。高さH0[m]の煙突から排出された汚染物質が、観測地点O(x,y,z)でどのくらいの濃度になるかを予測する最も一般的で簡単なモデルにプルームモデル^{2,3)}がある。これに基づいて計算してみよう。

表2に煙突の有効高さHeを計算するために必要な式をN I F Eへの入力書式で記載した。このように論文や教科書に書かれている数式をほとんど変更せずにそのままN I F Eに登録定義できる。煙突の諸元は、仮想のものを定義した。表2の定義式を登録すると次のようにHeを求めることができる。

> He

He=65.1972 [m]

利用者の求めに応じて、どんな計算要素の定義や値でも即座に表示する。いくつかの計算要素の値をまとめて表示するには、次のような複文で指定する。

> He;Hm;Ht;H0

He=65.1972 [m] Hm=4.89511 [m]

Ht=10.302 [m] H0=50 [m]

実際の高さよりも有効高さが何%高くなっているかを計算し、単位を[%]と表示するには、

> (He-H0)/H0*100[%]

(He-H0)/H0*100[%]=30.3943 [%]

このように、任意の数式を書くときN I F Eはたちどころにその計算値を表示する。利用者はこの機能を使って、数理モデルの各計算要素が現在どのような値になっているかを簡単に調べ、数理モデルの検討を続けられる。

同じ名前でもう一度定義すると、古い定義が新しい定義に置き換わる。数理モデルの修正は、変更のある関数をこの機能で新しい定義に書き換えればよい。また、計算条件を変更することで、さまざまなシミュレーションを行える。例えば、排気温度Tgを、

> Tg=600

old: Tg=500 '排気温度 [K]

new: Tg=600 '排気温度 [K]

とすると、古い定義の500[K]が削除されて、600[K]になる。このとき、Tgに付いていたコメントが自動的に引き継がれる。これに伴って、その上位の計算要素の値も同時に変化する。例えば、煙突の有効高さHeは、

> He

He=70.3654 [m]

のように値を表示してみると、先の約65mよりも高くなっていることが判る。このシミュレーション機能は、数理モデルや入力条件を徐々に変化させたときの応答を見るための重要な手段である。

つぎに、表3に観測点O(x,y,z)での汚染濃度をStatistical (Gaussian) Theoryで計算するための

表2 煙突の有効高さを求める計算式

煙突有効高さ		
> He=H0+(Hm+Ht)K		'煙突の有効高さ [m]
> Hm=4.77/(1+0.43U/V)*sqrt(Q0 V/U)		'運動量の効果 [m]
> Ht=6.37g Q0/U^3*(Tg-T0)/T0*(2log(J)+2/J-2)		'温度と密度の効果 [m]
> J=U^2/sqrt(Q0 V)*(0.43sqrt(T0/g/G)-0.28V/g*T0/(Tg-T0))+1		
煙突の仕様		
> Q0=1	'排出量 [Nm ³ /s]	
> V=5	'排出速度 [m/s]	
> Tg=500	'排気温度 [K]	
> K=1	'修正係数	
> H0=50	'煙突の高さ [m]	
> C0=1000000	'排出口濃度 [ng/Nm ³]	
> Q=Q0*C0	'汚染質の排出速度 [ng/s]	
気象条件		
> T0=300	'気温 [K]	
> U=3	'平均風速 [m/s]	
> g=9.8	'重力加速度 [m/s ²]	
> G=0.016	'大気温度勾配 [K/m]	

表3 煙突の風下での汚染濃度の計算式^{2,3)}

(Statistical (Gaussian) Theory)

```

汚染濃度
> Co(x,y,z)=if(x-0.01,0,Cx)           'O点(x,y,z)での汚染濃度 [ng/Nm3]
> Cx=Q/(2π σy σz U) * Kcy(Kcz1+Kcz2) ' (x,y,z)での濃度 [ng/Nm3]
> Kcy=exp(-0.5 y^2/σy^2)              ' Y軸の広がり
> Kcz1=exp(-0.5 (z-He)^2/σz^2)        ' Z軸の広がり (直接)
> Kcz2=exp(-0.5 (z+He)^2/σz^2)        ' Z軸の広がり (地面の反射)
    
```

```

大気の安定度
> =
> = 大気の安定度
> = =====
> = 風速 日射量 夜間
> = [m/s] 強 並 弱 本雲り 雲り 晴れ
> = -----
> = <2 A A-B B D - -
> = 2-3 A-B B C D E F
> = 3-4 B B-C C D D E
> = 4-6 C C-D D D D D
> = >6 D D D D D D
> = =====
> A=1 '大気の状態=強不安定
> B=2 '大気の状態=並不安定
> C=3 '大気の状態=やや不安定
> D=4 '大気の状態=中立
> E=5 '大気の状態=やや安定
> F=6 '大気の状態=並安定
    
```

```

排煙の広がり
> CONDITION=B '*** 大気の状態 ***
> σz=if(5000-σz1,5000,σz1) '煙の鉛直方向の広がり [m]
> σz1=σza (x/1000)^σzb
> σza=Za[ lookup(Za[* ,0],LE,x), CONDITION]
> σzb=Zb[ lookup(Zb[* ,0],LE,x), CONDITION]
> σy=0.46511628x * tan(0.017453293(σyc-σyd*log(x/1000))) '水平方向広がり [m]
> σyc=Ycd[CONDITION,1]
> σyd=Ycd[CONDITION,2]
    
```

```

排煙の広がり計算するための数表
> = x, A, B, C, D, E, F,
> Za[19,7]={ 0, 158.080, 90.673, 61.141, 34.459, 23.331, 15.209, &
& 150, 170.220, 90.673, 61.141, 34.459, 23.331, 15.209, &
.
. (途中省略)
.
& 40000, 453.850, 109.300, 61.141, 44.053, 47.618, 27.074, &
& 60000, 453.850, 109.300, 61.141, 44.053, 47.618, 34.219 }

> = x, A, B, C, D, E, F,
> Zb[19,7]={ 0, 1.05420, 0.93193, 0.91465, 0.86974, 0.81956, 0.81558, &
& 150, 1.09320, 0.93193, 0.91465, 0.86974, 0.81956, 0.81558, &
.
. (途中省略)
.
& 40000, 2.11660, 1.09710, 0.91465, 0.29592, 0.29592, 0.27436, &
& 60000, 2.11660, 1.09710, 0.91465, 0.29592, 0.29592, 0.21716 }

> Ycd[7,3]={ 0, 0, 0,
& A, 24.1670, 2.5334, &
& B, 18.3330, 1.8096, &
& C, 12.5000, 1.0857, &
& D, 8.3330, 0.72382, &
& E, 6.2500, 0.54287, &
& F, 4.1667, 0.36191 }
    
```

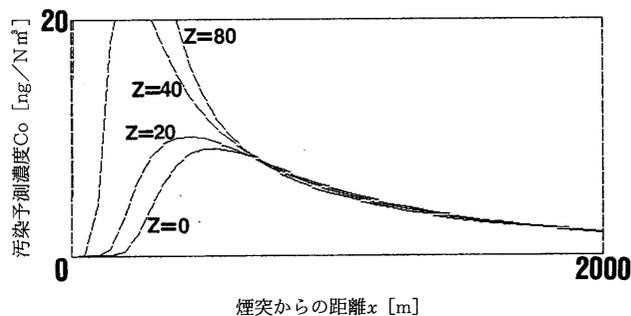


図1 測定高度 (Z) についての汚染予測濃度

表4 汚染濃度の距離別一覧表を作る命令と計算結果

```

> .set echo off
> x <0[m]; y <0[m]; z <0[m]; .repeat 20; x <x+100; Co; σz; σy
x=0 [m]   y=0 [m]   z=0 [m]
---- 繰り返し:20回 ----
x=100 [m]   Co=1.43608e-007 [ng/Nm3]   σz=10.6059 [m]   σy=19.2655 [m]
x=200 [m]   Co=0.343013 [ng/Nm3]       σz=20.2342 [m]   σy=36.1662 [m]
x=300 [m]   Co=4.02241 [ng/Nm3]        σz=29.5251 [m]   σy=52.2025 [m]
x=400 [m]   Co=7.71169 [ng/Nm3]        σz=38.6034 [m]   σy=67.6827 [m]
x=500 [m]   Co=9.01619 [ng/Nm3]        σz=47.5269 [m]   σy=82.7522 [m]
x=600 [m]   Co=8.85438 [ng/Nm3]        σz=56.3288 [m]   σy=97.4959 [m]
x=700 [m]   Co=8.11475 [ng/Nm3]        σz=65.031 [m]    σy=111.97 [m]
.
. (途中省略)
.
x=1900 [m]   Co=2.15108 [ng/Nm3]       σz=164.914 [m]   σy=273.076 [m]
x=2000 [m]   Co=1.97571 [ng/Nm3]       σz=172.988 [m]   σy=285.798 [m]

```

表5 汚染濃度のグラフを描く

```

> .coord 0
> Xmin=0[m]; Xmax=2000[m]
> Ymin=0[ng/m3]; Ymax=20[ng/m3]
> draw> .draw i,Start/dx,End/dx; i*dx, Co (i*dx,y,z)
> Start <0; End <2000; dx <50; y <0;
> z <0; .do draw
> z <20; .do draw
> z <40; .do draw
> z <60; .do draw

```

定義式をN I F Eの書式で示す。Coの定義に現れるif()関数は、第一引数の正負で第二、第三引数の値を選ぶ関数でN I F Eに予め組み込まれている。この関数によって、煙突の風上での汚染濃度をゼロにした。排気ガスの拡散の様子には大気安定度は重要な要素になる。どの安定度を選択すべきかは、このモデルの重要な内容であるので選択の基準をコメントとして記述しておく。「=」で始まる文、つまり関数名がない関数定義文は、コメントの定義文と見なされ、計算式の説明などを関数定義と同様に登録して置くようにする。排煙の広がりや計算する部分では、大気安定度に応じて数表を読み取る位置が異なるので、その位置をCONDITIONという値で指し示すようにした。表3では、大気の状態をとりあえずCONDITION=Bとしておく。定義中に現れるlookup(Za[*],LE,x)は、数表Zaを検索してZa[i,0]がx以下であるようなiを値として返す。つまり、行列を数表とみなして該当項目を検索する関数で、これもN I F Eの組み込み関数である。この様な関数を用いることで、大気の状態をCONDITIONに定義するだけで、必要な数表を参照して計算を進められる。数表の記述では、継続行の印&を用いて複数の行にまたがって記述することで行と列の対応を分かりやすくした。

表2、表3のように定義式が登録されれば、任意の空間座標の観測点O(x,y,z)での汚染濃度を求めることが出来る。例えば、風下200m高さ0mでは、
> Co(200,0,0)

Co(200,0,0)=0.343013 [ng/Nm3]

さらに、表4の様に示すように、.repeat命令と変数への代入を組み合わせることで、煙突からの風下距離xに対する汚染予測濃度Coなどの一覧表を作るのも容易である。また、表5に示した命令によって、汚染予測濃度のグラフが描かれる。画面上に表示されたグラフに説明のための文字を加えたものが、図1である。

このように利用者は、任意の計算要素の定義や値を参照したり、計算式に修正を加えたり、一覧表を作成したり、グラフを描いたりして、構築しようとする数理モデルの結果を検討する。問題点や改良出来ることがあれば、更にモデルの計算式や条件を変更して計算を試みる。これを繰り返すことで、高度な数理モデルを構築することができるであろう。

4. 結語

環境科学の分野で数理モデルを必要とする度合は、これから更に高くなると考えられる。ひとつは環境アセスメントや未来予測のために、もうひとつは汚染機構の説明と汚染対策の指針を得るためである。これまで提案されてきた数理モデルの多くでは、普通の数学物理化学の書式から逸脱した電子計算機の為の言語によって記述され、大型電子計算機といういわばブラックボックスの中からいきなり「地球が温暖化する」とか「皮膚ガンが増加する」などの結論だけが出てくることになっている。

しかし、環境を守る原則で政策上の方針を導くための数理モデルは、できるだけ多くの人が理解し操作できるように提示される必要がある。そのためには、通常の数学や物理化学の書式で関数を積み上げて数理モデルを構築できること、数理モデルの妥当性について何回でも条件を変えながら検討できること、パーソナルコンピュータで計算操作が簡単にできることなどの要件が求められる。

ここで開発したN I F Eは、上記の要件を満たした数理モデルの構築支援ツールとして広範な用途に利用できることが判った。

現在、N I F Eの基本プログラムはシェアウェアとして広く公開されているので、誰でも試用することができる。日本電気PC9801とエプソンPC286/386のMS-DOS上で動作する。最も簡単で早い入手方法は、商業コンピュータ通信のNIFTY-Serveに接続し、F GALAP というフォーラムのデータライブラリ7番(情報管理・統計ソフト)で入手する方法である。

参考文献

- 1) 鈴木基之：環境科学におけるモデル化の役割、水質汚濁研究9(10), pp.2-7 (1986)
- 2) Handbook Of Air Pollution Technology, pp.871-877 (1984).
- 3) 矢野雄幸, 佐藤弘三：拡散方程式入門, pp.118-119 (1978).