# 続連載「EQUATRAN-M」技術計算方程式解法言語（V・最終回）

# 質問に答えて

佐渡友　秀　夫*
宮　原　是　中**

EQUATRAN-Mの文法と機能について，前連載[1]と続連載[4]で主要なものは一通り説明してきた．できるだけケミカルエンジニアになじみのある例題を上げて解説するように心がけてきたつもりであるが，ともすれば一面的な説明になってしまわなかったかと心配している．

連載を終るにあたって，これまで読者諸氏からEQUATRAN-Mについてたくさんの質問が寄せられたので，まとめてお答えすることにした．回答は機能拡張版（バージョンII）に従っており，できるだけ連載の関連箇所を参照し［　］で示すようにしたのでEQUATRAN-Mの理解の一助と使用上のヒントになれば幸いである．

## 1．文法について

**【Q1】** BASICによるプログラミングとの違いは．

**【A】** BASICでもFORTRANでも「＝」は右辺の式の値を左辺の変数に代入する意味に使われるが，EQUATRAN-Mでは「＝」は左辺と右辺の式が等しいという数学的等号を意味する．これが基本的な違いで，これによりEQUATRAN-Mでは方程式の順序を気にせずそのまま書くだけで解が得られ，式を追加したり入力と出力の変数の関係を逆転するなどの変更にも柔軟に対処できるわけである．比較のためBASICのプログラムをEQUATRAN-Mで書いたのが［例題4.8］である．随分簡潔に表現できることが分かる．

**【Q2】** BASICのGOTO文とかIF文あるいはFOR文に対応する処理ができないか．

**【A】** EQUATRAN-Mでは式の順番は意味を持たないのでジャンプ（GOTO文）の必要はない．条件によって異なる式を用いたい場合には条件付きの式（WHEN文）を使えばよい．IF文もWHEN文で大体対応できる［例題2.2］.FOR文による繰り返しは配列変数を使って書くことができる［例題8.3，II.4，III.6］．

```
10  FOR I=1 TO N : X(I) = I : NEXT I
                  ⇩
10: X(1) = 1 ; X(2:N) = X(1:N-1) + 1
```

実行の繰り返しはREPEAT文が便利である［例題I.2，IV.3］．

```
10  FOR X=0 TO 1 STEP 0.1
20    Y = SQR( 1-X^2 ) : PRINT X,Y
30  NEXT X
              ⇩
10:    Y = SQRT( 1-X^2 )
11: REPEAT X [0,1] STEP 0.1
12: TREND Y
```

**【Q3】** 入力するデータを毎度キーインするのは面倒なので登録しておくことはできないか．

**【A】** BASICのDATA文を想定していると思われ

* Hideo Sadotomo　三井東圧化学(株) システム部　主席部員
** Koreatsu Miyahara　三井東圧化学(株) システム部　部長

るが，INPUT文で読み込みたいデータを保存する機能はデータテキストで実現できる［例題4.5］.

【Q4】サブルーチンは使えるか.

【A】いわゆるサブルーチンはないが，マクロによって同様なことができる．ひとかたまりの式をマクロとして登録しておき呼び出して使う［第2回5節，第5回3節，例題2.5，4.4他多数］．汎用のものを多種用意しておき複写して使うと便利である.

【Q5】配列の大きさをパラメタとして与えられるか

【A】配列はVAR文で2次元まで定義できる［第2回4節］．配列のサイズをシンボルで定義しておき，LOCAL文かGLOBAL文でその変数に大きさを指定すればよい［例題7.2．7.3，Ⅰ.3］.

```
LOCAL n=5
VAR A(n,n),B(n,n),I(n,n),b(n),x(n)
```

【Q6】行列演算はどのように書くのか.

【A】いま，変数を上のように定義したとき代表的な行列演算は次のように書ける.

・連立一次方程式　SUM(A＊x)＝b
　［例題7.3～7.6，Ⅰ.3］
・逆行列B　SUM(A＊B(,1))＝I(,1)
　（Ⅰは単位行列）　⋮
　SUM(A＊B(,n))＝I(,n)
・転置行列　A⁻と書く
　［例題Ⅰ.3］
・固有値　固有方程式を解く
　［例題4.7．7.8］

【Q7】線図から数値を読み取り計算に使いたいが.

【A】読み取った数値を数表として与え直線内挿して使うことができる（TABLE文）［例題2.3，7.1］．数表は2次元の（2変数で表される）表まで定義できるのでパラメトリックな線図にも対応できる［例題8.2，Ⅳ.5］．直線内挿ではなく階段状の関数としたいときにはTABLE文でSTEPと指定をすればよい．便覧などにはデータが線図や表として与えられていることが多く，これから近似式を作るのは結構手間がかかる．そんなときに便利である.

表Ⅴ-1　組込み込み関数一覧表（バージョンⅡ）

| 関　数 | 数学的表記 | 「EQUATRAN-M」の表記 |
|---|---|---|
| 指数 | $e^x$ | exp(x) |
| 10のべき乗 | $10^x$ | exp10(x) |
| 平方 | $x^2$ | sqre(x) |
| 自然対数 | $\log_e(x)$ | loge(x) |
| 常用対数 | $\log_{10}(x)$ | log10(x) |
| 平方根 | $\sqrt{x}$ | sqrt(x) |
| 正弦 | $\sin x$ | sin(x) |
| 余弦 | $\cos x$ | cos(x) |
| 正接 | $\tan x$ | tan(x) |
| 逆正弦 | $\sin^{-1}x$ | asin(x) |
| 逆余弦 | $\cos^{-1}x$ | acos(x) |
| 逆正接 | $\tan^{-1}x$ | atan(x) |
| 双曲線正弦 | $\sinh x$ | sinh(x) |
| 双曲線余弦 | $\cosh x$ | cosh(x) |
| 双曲線正弦 | $\tanh x$ | tanh(x) |
| 双曲線逆正弦 | $\sinh^{-1}x$ | asinh(x) |
| 双曲線逆余弦 | $\cosh^{-1}x$ | acosh(x) |
| 双曲線逆正接 | $\tanh^{-1}x$ | atanh(x) |
| 配列変数の総和 | $\sum_{i=1}^{n} y_i$ | sum(y) |
| 配列変数の総積 | $\prod_{i=1}^{n} y_i$ | prod(y) |
| 最大の変数値 | —— | max(a,b,c,…) |
| 最小の変数値 | —— | min(a,b,c,…) |
| 配列の最大要素 | $\max_i y_i$ | maxof(y) |
| 配列の最小要素 | $\min_i y_i$ | minof(y) |
| 整数値に切り捨て | —— | int(x) |
| x/yの剰余 | —— | mod(x,y) |
| 符号の取り出し | —— | sign(x,y) |
| 論理値化 | —— | if(x) |
| 絶対値 | $\lvert x\rvert$ | abs(x) |

【Q8】整数の取り扱いはどうしたらよいか.

【A】組み込み関数として［表1-11］に記載されている24種に，バージョンⅡで整数値化や剰余などが追加され全部で29の関数が用意されている（表Ⅴ-1）．実数を整数にする（小数部を切り捨てる）にはINT関数を使えばできる.

```
/* 伝熱面積から管本数を計算する */
Ncal = A / ( _pi*dio*L )
Nact = INT( Ncal ) + 1
```

【Q9】複素数の演算をしたいのだが.

【A】複素数を直接扱うことはできないが，実部

と虚部とを別々の変数（またはベクトルの要素）として取り扱えば対応は可能である［例題8.1］.

## 2. 数値計算法について

### 2. 1 繰り返し計算法

**【Q10】** 繰り返し計算の手法は何か．収束性に問題はないか．

**【A】** 収束の確実性を改良した修正Newton法を採用している．どんな初期値でも必ず収束するというわけにはいかないが，妥当な変域と初期値を与えれば収束はかなり速い．なお，連立一次方程式は直接解くので繰り返し計算は行わない．

**【Q11】** 繰り返し変数はいくつまで可能か．

**【A】** 自動リセット(繰り返し変数を自動的に選定する機能)では50まで．RESET文により指定する場合には制限がない．

**【Q12】** どの変数が未収束か分かると便利だが．

**【A】** ダンプオプション［第2回6節，例題3.4］で繰り返し計算の途中経過を表示できるので，これを調べればどの変数が未収束か分かる．

**【Q13】** 収束しないときにはどんな対策が可能か．

**【A】** 未収束の状況に応じていくつかの対策が考えられ，次の項目をチェックしてみる．

①式が正しいか（変数名はどうか）

②繰り返し変数は適切に選ばれているか

③収束判定式は繰り返し変数の動きに対し適当な感度を示すか

④繰り返し変数の変域と初期値は妥当か

⑤繰り返し回数の不足か（→Q15）

収束性にとって数式モデルの善し悪しが決定的な要因である(第3節)．繰り返し変数として物理的な意味が明らかな変数（例えば，比率のような）を選べば適切な変域や初期値を与えやすい．初期値の適否は収束性に大きく影響する．Newton法では解の探索方向を決めるのに微係数を用いるが，この値は変域の幅に対する微小変化量から数値微分で算出する．したがって，変域はできるだけ絞られていることが望ましい．収束判定式の感度を調べるには，繰り返し変数に対するその式の左辺と右辺の差の関数形をグラフ機能を使って描いてみるとよい．

**【Q14】** 自動リセットとRESET文による指定との使い分けは．

**【A】** 繰り返し変数と収束判定式の組合せについて自動リセットでもかなりよいものを選んでくるがいつでもベストとは限らない．最善の選択には問題の現象的理解が必須で，技術者の知識そのものであるから，自動リセットの結果を参考にしてRESET文で指定することも大切．

**【Q15】** 計算の精度と繰り返し計算の回数は．

**【A】** 演数は倍精度（有効数字約16桁）で行い，収束の許容誤差は標準値として$10^{-8}$，繰り返し回数の標準値は100回である．これらの標準値はDEFAULTコマンドで再設定できるし，許容誤差はRESET文でも指定できる．

**【Q16】** 複数の解があるとき同時に求められるか．

**【A】** 最初に得られる解を1つだけ求める．現象を数式モデルで表したとき，数式の上で2つ以上の解が存在しても有意な解は1つだけのことが多い．強いて2つ以上の解を求めるときは，実行時に変域と初期値を与えることができるのでそれらをずらしてケーススタディすればよい．

**【Q17】** 高次代数方程式を解く場合も同様か．

**【A】** 高次代数方程式$f(x)$を解くには次のようなアイデアがある．

```
/* 高次代数方程式を解く */
    f = x^3 - 4*x^2 - 7*x + 10
    f / (x-x1) / (x-x2) = 0
  INPUT x1, x2
```

初めはx1とx2に適当に大きな数を与え（3行目の分母が0にならないように），得られた解を順次x1，x2に入力してやる（GOコマンドでケーススタディ）．重根がなければつぎつぎと解が得られる．

**【Q18】** 例外演算が発生したときは．

**【A】** 0での割算や平方根の引き数が負などの例外演算が起きたとき，影響を受けた計算値は計算結果の出力時に＊印を付して表示される．そして，例外演算の原因はVALUEコマンドを使って調べることができる．また，数値計算の過程をトレースオプションで逐次表示させるのも調査の有力な手段である［第9回4節］.

### 2. 2 微分方程式

**【Q19】** 数値積分手法は何か．

**【A】** 数値積分の手法は3種類用意されている．

①Euler 法
②固定きざみの Runge-Kutta法
③可変きざみの Runge-Kutta 法
で，数値積分を指示する INTEGRAL 文の BY 項で指定する［例題Ⅱ.5］．通常は②で十分であるが計算精度を確保したいときには③を使う．①を使えば精度は多少犠牲になるが計算時間を稼ぐことができる．

【Q20】定積分の計算はできるか．

【A】定積分は次の変形をすれば通常の常微分方程式と同じに扱える［例題Ⅰ.5］

$$y = \int_a^b f(x)\,dx$$
$$\Downarrow$$
$$dy/dx = f(x) \qquad a \leqq x \leqq b$$
$$x = a \text{ で } y = 0$$

【Q21】数値積分の途中で条件により計算を中断したり再開することはできるか．

【A】INTEGRAL 文の BREAK 項に中断条件を書いておけば，条件が成り立つと計算が中断される［例題Ⅱ.1］．また，ESC キーを押すことにより任意の時点で中断することもできる．ここで，INPUT 文で指定した変数の入力を要求してくるので，新たな数値をキーインすると計算が再開される．この機能を使えば計算結果を見ながらダイナミックシミュレーションを対話的に実行することができる［例題Ⅳ.5］．

```
/* 一次反応のシミュレーション */
  C' = -k0*exp(-E/(R*(T+273.15)))*C
  Cp*ro*V*T' = U*A*(Tj-T) - V*dH*C'
  cond = (T>=To)
INTEGRAL t[0,10] STEP 0.1 BREAK cond
INPUT Tj, To
```

【Q22】動特性の解析は伝達関数で表現されることが多いがこのような問題にも使えるか．

【A】伝達関数をそのまま記述することはできないが，ラプラス逆変換で簡単に微分方程式に直せるのでそれを解けばよい［例題Ⅳ.4］．

【Q23】制御器や遅れ要素はどう扱ったらよいか．

【A】微分方程式で表されるダイナミックシミュレーションのために微分関数，むだ時間やステップ関数などの７種の動的な組み込み関数が用意されている（表Ⅴ-2）．これを使えば PID コントローラを定義したり［例題Ⅳ.5］，ヒステリシス特性［例題Ⅳ.2］やむだ時間遅れ［例題Ⅳ.4］を表すことができ，たいていの動特性解析に間に合うと思う．

表Ⅴ-2　動的な組み込み関数一覧表

| 関　　数 | 「EQUATRAN-M」の表記 |
|---|---|
| 微分関数 | deriv(v,vs) |
| むだ時間 | delay(v,vs,d,N) |
| 前回値関数 | prev(v,vs) |
| ステップ関数 | step(v) |
| 同期 | sync(v) |
| 一様乱数 | randu(n) |
| 正規分布乱数 | gauss(n) |

【Q24】制御の問題では定常状態を基準とすることが多いが，定常状態は別途計算するのか．

【A】定常状態は時間による微分項が０の時の解である．したがって，プロセス方程式の積分初期値を設定する＃式を微分項が０の式（例：X' 0）で置き換えてやれば定常状態の計算ができるわけである．このとき INTEGRAL 文は殺しておく．求めた定常状態の解は OUTPUT 文でリザルトファイルに書いておき，元のプロセス方程式の初期条件として読み込めば所望の計算を進めることができる［例題Ⅱ.6］．

【Q25】繰り返し計算との組合せは可能か．

【A】微分方程式を繰り返し計算を含む代数方程式と連立して解くことができる［例題Ⅱ.5］．ただし，繰り返し計算のループの中に微分方程式を抱えることはできない．そのときは，積分結果を画面で見ながらケーススタディの機能を使って対話的に修正反復してやる［例題Ⅲ.5］．

【Q26】数値積分過程で発散するときの対策は．

【A】微係数が大きな系では数値解が指数関数的に大きくなってしまうことがある．誤差が拡大伝播していくためである．このときは
①INTEGRAL 文で STEP を小さくする
②可変きざみの Runge-Kutta 法を使う
とよい［例題Ⅱ.5，Ⅲ.1］

【Q27】境界値問題は扱えるか．

【A】初期値問題を対象としているので直接は扱

えないが，積分の終端で与えられていると同数の条件を初期条件として仮定すれば初期値問題に帰着できる［例題Ⅱ.3，Ⅲ.5］．このとき，仮定した初期値について修正反復が必要になるので境界条件が満足されるまで対話的に実行することになる．あるいは，微分方程式を差分近似して解く方法もある［例題８.3］．

【Q28】偏微分方程式はどうか．

【A】偏微分を差分近似することにより連立一次方程式に還元して解くのが一般的である［例題９.3，Ⅱ.4］．また，差分近似と常微分方程式を組み合わせて解くこともできる［例題Ⅱ.4別解，Ⅲ.6］．差分近似の方法は陽公式，Crank-Nicolson法など各種提案されている．差分法では方程式の規模が大きくなること，きざみの大きさの与え方によっては計算が不安定になることがあるので注意が必要である．

【Q29】常微分方程式のパラメタ決定問題は．

【A】反応速度式のパラメタを求める場合などに現れ，最適化手法との組合せになる．数値計算の最も難しい問題の一つで，EQUATRAN-Mでも今のところ直接解くことはできない．ケーススタディの機能を使ってパラメタを摂動させるのがよかろう．

### 2.3 最適化計算

【Q30】最適化手法は何か．

【A】１変数のときは２次方程式近似法，多変数のときはBoxのcomplex法を改良した方法で，制約条件を扱えるのが特徴である．他の方法もそうであるが，変数領域が凸であることと目的関数が単峰であることを前提にしている．

【Q31】独立変数はいくつまでか．

【A】10変数まで可能である．しかし，目的関数の特性と計算時間の浪費を考えると５変数が実用上の限界であろう［例題５.2］．

【Q32】うまく最適解が求まらないのだが．

【A】多変数の最適化では実行可能解（feasible solution）から出発することが必要である．凸領域と単峰性についても要確認．単峰性を調べるには初期値をいろいろ変えて同じ解に達するかどうかみるのも手である．

【Q33】非線形最小二乗法は扱えるか．

【A】線形最小二乗法では正規方程式を連立一次方程式として解く問題に帰着される［例題７.4］が，非線形の場合には残差二乗和を最小にする最適化問題として解くことができる［例題７.7］．なお，混合系では非線形なパラメタを独立変数とし，線形なものについて正規方程式を解くのが効率的である［例題７.5］．

【Q34】線形計画法の問題は解けるか．

【A】いわゆる線形計画法の手法(LP)では扱えないが，一般の最適化問題として解くことはできる．二次計画法(QP)についても同様である．

## 3．問題の定式化について

【Q35】扱える問題の大きさは

【A】パソコンのメモリー量に依存し，640KB装備の場合で数百～2000変数までの問題が扱える．この巾は個々の数式の複雑さに依存する．

【Q36】方程式と変数の数は一致しなければならないか

【A】数学的な意味からも一致することが必要条件である．解ける部分だけの解を求めるとか多過ぎる式を最適化問題として扱うという考え方もある．しかし，モデルの間違いや変数名の誤記を見過ごす危険を考えると，物理的に無意味な部分解や最適解を求めるのはナンセンスであるという立場をとっている．

【Q37】方程式と変数の数の一致を調べるには

【A】最もよくある数の不一致の原因は変数名の書き間違いである．このチェックを助けるためコンパイルオプションとして診断メッセージが用意されている［表９-８］．

```
/* 向流熱交換器の設計 */
  P = (t2-t1)/(T1-t1)
  R = g*c/(G*C) ; R =(T1-T2)/(t2-t1)
  UA/(g*c)=loge((1-P)/(1-P*R))/(R-1)
INPUT  G,C,g,c,U,A,T1,t1
OUTPUT t1,t2,T1,T2
```

上の例は例題４.1の解（表４-２）のリストの一部を書き写したもので，これに対する診断メッセージを表Ⅴ-３に示す．「(2)定義されていない変数」とはVAR文で定義していないもの．「(4)一度しか使われていない変数」とは方程式や入出力文に一度しか現れない変数のことで，間違いの可能性があるものである．ここでは，

表V-3　診断メッセージ

```
[  診断メッセージ  ]
(1) 式と変数の数
      式 ......................   4 (E)
      全変数 ..................  13 (V)
      積分変数 ................   0 (Vs)
      入力変数 ................   8 (Vi)
      使われていない変数 ......   0 (Vn)
(2) 定義されていない変数
      P             t2            t1            T1
      R             g             c             G
      C             T2            UA            U
      A
(4) 一度しか使われていない変数
      UA            U             A
(5) 式と変数の数の一致
      (E)-{(V)-(Vs)-(Vi)-(Vn)} =   -1  ...  式が少なすぎます
```

UAとU，Aの3つが指摘されているが，実はUAはU＊Aの書き誤りであった．このように変数名が間違っていないかを調べ，さらに方程式が正しいかどうかもう一度見直すことが大切である．

なお，プロセスの物質・熱収支計算ではストリームに沿って各機器毎に「入り口条件が与えられたときに出口条件が決定できるか」を順次調べてみるとよい［第5回2.3節］．

**【Q38】** 方程式と変数の過不足を防ぐには

**【A】** よく見かける間違いに値が0の変数の指定を忘れることがある．例えば，プロセスの物質収支計算ではストリームを成分数の大きさをもつ配列変数で表す［例題2.4，5.4，6.1］が，原料中に存在しない成分については指定を省略してしまう等である．値が0ということも大変重要な情報なのである［例題6.1，6.5］．また，収支計算で全ての機器回りの収支式と全体の収支式を書くと，それらの間には従属な関係があり方程式は1つ余分である．これも注意が必要．

**【Q39】** 従属な方程式があるときは

```
/* 三元連立一次方程式 */
   2*x1 + 4*x2 -   x3 =  5
     x1 -   x2 + 2*x3 =  5
   3*x1 + 2*x2 +   x3 = 10
> RUN⏎
            ⇩
 [ 計算結果 ]
 x1           = *0
 x2           = *0
 x3           = *0
E211 演算不能エラーが発生しています
 > VALUE⏎
            ⇩
  変数名 > x1
  x1          = *0
線形方程式の従属エラーが起きています
```

**【A】** 計算を実行した結果方程式間に線形従属関係がある場合には，［計算結果］表示のとき“演算不能”というメッセージが出され，同時に関連する変数の前に＊が付けられる．そこで，VALUEコマンドを使ってこの変数を表示させると“従属エラー”が発生したことが分かる．

**【Q40】** 非線形方程式でも従属エラーが起きるか

**【A】** 初期値によっては起きることがある．次の例はGaussの積分公式の分点 $u_i$ と係数 $K_i$ を求めるものである[2)]．

```
/* Gauss積分の分点と係数の計算 */
   K1       +K2       +K3       = 1
   K1*u1    +K2*u2    +K3*u3    = 0
   K1*u1^2  +K2*u2^2  +K3*u3^2  = 1/3
e1:K1*u1^3  +K2*u2^3  +K3*u3^3  = 0
e2:K1*u1^4  +K2*u2^4  +K3*u3^4  = 1/5
e3:K1*u1^5  +K2*u2^5  +K3*u3^5  = 0
  RESET u1# 1 [-1,1] BY e1
  RESET u2# 0 [-1,1] BY e2
  RESET u3#-1 [-1,1] BY e3
```

u1，u2，u3は［−1，1］に根をもつが，たまたま同じ初期値を与えると第2，4，6式は $K_i$ につき同一の関係になってしまう．前問と同様にVALUEコマンドで調べればすぐに状況は判明する．なお，上のリストのようにそれぞれに異なる初期値を与えれば正確に収束する．

**【Q41】** 数式モデル作成上で注意すべき点は

**【A】** あらゆる問題について言えることであるが，必要以上に精密なモデルを構築しても無駄なことが多い．まず最初はできるだけ簡略化されたモデルを作ることが大切である．その意味からも線形化モデルを用いることを勧めたい．このような線形化の工夫は［例題2.4，5.4，6.5］のプロセス物質収支計算[3)]，［例題4.1〜4.6］の熱交換器の計算[4)]や［例題6.3］のフラッシュ計算にも見られ，安定して解が得られるモデルとなっている．

**【Q42】** 収束性の良いモデルを作るには

**【A】** 非線形な方程式を含むモデルでは収束性の良さが重要である．Q13でも述べたように，繰り返し変数の選び方には多少コツがある．非線形なパラメタを繰り返し変数に選ぶことによってモデルを線形化するのがよい［第5回2.4節］．例によって説明しよう．N成分からなるプロセスのストリーム①が2つのストリーム

②，③に分流する場合を考える．

```
[1]  strm(1) = strm(2) + strm(3)
     strm(1,1:N-1)/SUM(strm(1)) ..
     = strm(2,1:N-1)/SUM(strm(2))
     SUM(strm(1))/SUM(strm(2)) = b

[2]  strm(1) = strm(2) + strm(3)
     strm(2) = strm(1)*b
```

〔1〕は2つのストリームの組成が等しいことと分流比がbであるという条件を表している．strm(1)やbの値が他の条件から繰り返し計算で決められるときには各成分について非線形な式ができてしまう．［2］も全く同じことを表しているが，表記が簡単であるだけでなくbを仮定すれば線形な式となり収束性がよい［例題6.1］．また，うまい変数変換により収束しやすくしている例として［例題3.3］があり，状態方程式を解くのに体積$V$でなく圧縮係数$z$を繰り返し変数に選んでいる．さらに，収束判定式によっても収束性は左右される．繰り返し変数と密接に関連し，かつその変化に対してなるべく線形に応答する式がよい．これについて［例題3.5］に具体例が述べてある．

## 4．グラフ化について

**【Q43】** 関数の形を描くには

**【A】** 2つの方法がある．

①REPEAT文で独立変数の範囲と計算のきざみ(STEP項)を指定する．REPEAT文は独立変数をきざみ分づつ増分しながら計算を繰り返す．同時に，ファイルの出力を指示するOUTPUT1文（またはOUTPUT2文）のSTEP項にもグラフを描くきざみを指定し，結果をグラフモードに渡す［例題Ⅰ.1］

②独立変数に対するデータをデータテキスト［例題9.1］として用意し，実行結果をグラフ化する．

通常は①が便利．

**【Q44】** 一価関数でない場合は

**【A】** データ間の線を引くにはいくつかの方法が用意されている．円や楕円のように変数に対して2つ以上の関数値がある場合にも，媒介変数などを導入して線を描く順に（独立変数，関数）のデータの組が計算できれば，3次のスプライン曲線Ⅱを指定することにより滑らかな曲線を描くことができる．さらに，位相面軌跡のように螺旋状の曲線も描ける［例題Ⅳ.2］．

**【Q45】** 関数の等高線は描けるか

**【A】** 等高線は2つの独立変数に対する関数形を描くものである．鳥かん図のような立体図は本グラフ機能の範囲外であるが，関数値をパラメタとして与え，2つの変数間の関係が順次計算できれば前問の機能を使って描ける．

**【Q46】** 常微分方程式をケーススタディし，結果をグラフ化するには

**【A】** 特定の変数の値を種々変えて常微分方程式を解き，その結果を同一のグラフ上に並べて描くには2つの方法がある．

①変数をケーススタディの数だけのベクトルとして定義し，方程式を連立させた形で解きグラフ化する［例題Ⅰ.4］．

②特定の変数または初期値を入力し，値を変えた計算結果を別ファイルに保存しグラフの重ね書きをする［例題Ⅱ.1，Ⅱ.2，Ⅳ.2］．

**【Q47】** 線は5本までしか描けないか

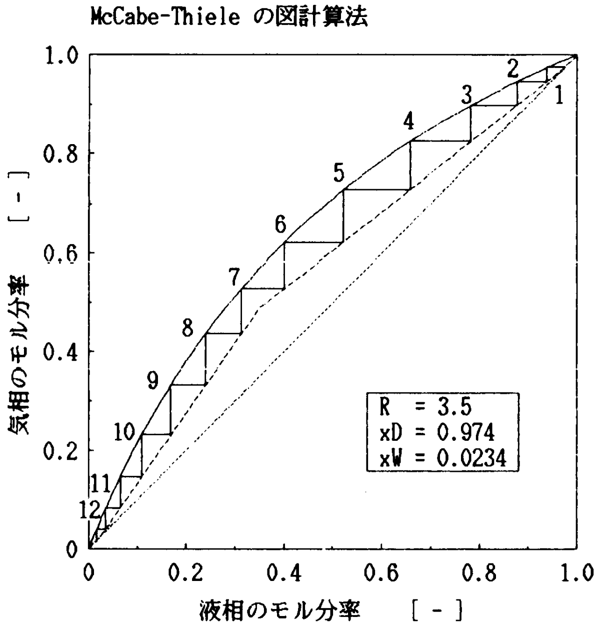
**【A】** 通常1つの独立変数に対していくつかの関数のグラフを描いたり，1つの関数についてパラメトリックに線を引くこともできる．しかし，同時には5本しか線が描けない．それ以上の線を引くときには重ね書きをする（MERGEコマンドを使う）か2枚のグラフにすればよい．［例題Ⅳ.1］の計算結果のグラフは重ね書きで6本の線を描いている．また，［例題Ⅲ.5，Ⅳ.5］では2枚のグラフとして表示している．

**【Q48】** データをプロットし近似線を引くには

**【A】** 測定データの組を与え最小二乗法で近似した線を引くには，線の種類として「1次近似線」，「2次近似線」，「3次近似線」の中から選んで指定すればよい．データ点のプロットと近似線が描かれる．さらに，近似式の係数も表示しようとすれば［例題Ⅰ.3］のようになる．

**【Q49】** 2成分蒸留の階段作図のようなグラフは描けるか

**【A】** 多少工夫を要するが図Ⅴ-1に示すようなグラフが描ける．この計算は逐次段毎に進めていくので，PREV関数（表Ⅴ-2）を用いて前段の計算値を取り出し，段数分だけREPEAT文




図V-1 2成分蒸留の図計算法

で繰り返してやる．リストと詳細な説明は表V-6の文献〈7〉を参照願いたい．

【Q50】ビジネスグラフは描けるか

【A】EQUATRAN-Mは技術計算を目的としているので，棒グラフや円グラフのようないわゆるビジネスグラフを描くことはできない．

## 5．EQUATRAN-Mの機能について

【Q51】どんな言語で書かれているか

【A】言語仕様とか計算速度などを勘案して主にC言語で，一部はアセンブラで書かれている．

【Q52】内部構成はどうなっているか

【A】旧バージョンの構成については［第9回1節］で説明した．バージョンⅡで大きく変わったのはエディタ部，コンパイル部，実行（ゴー）部に新たにグラフ部が追加されたこととコンパイル部で微分方程式を処理できるようになったことである．詳細については表V-6〈6〉のP.11およびP.162を参照願いたい．

【Q53】入力の要求や計算結果の表示の際に変数に説明文とかコメントを出すことはできないか

【A】確かに前連載の例題の解では変数名しか表示されず分かりにくい面があった．バージョンⅡでは，変数定義（VAR文）のとき変数に内容とか単位などの説明項を定義してやれば，それが入力要求や計算結果表示のときにそのまま出されるように改善された［例題Ⅰ.2他］．

【Q54】結果の表示形式をユーザが指定できないか

【A】結果の表示形式（FORMAT）は固定されているが，表示する変数の順序はOUTPUT文で指定できるし．行列の行と列を入れ換えて表示することもできる（転置行列を指定）．

【Q55】計算結果をBASICのプログラムに渡せないか

【A】MS-DOSの下で作動するBASICプログラムにOUTPUT 1文またはOUTPUT 2文で作ったリザルトファイルを介して計算結果を渡すことができる．また，BASICプログラムで作ったデータをEQUATRAN-Mのデータテキストとして読み込むこともできる．それは，EQUATRAN-Mのファイル形式がMS-DOSのそれで書かれているからである．このことについて［第9回5節］に説明されているが，バージョンⅡではより簡便にできるよう改善された．

【Q56】ケーススタディをやりたいのだが

【A】いままでもケーススタディのことが出てきたがここでまとめておこう．設計変数やパラメタをいろいろ変えて計算することをケーススタディと言っているが，その方法には3つある．

①GOコマンドを使い対話形式で前回の入力値を一部変えながら実行する［第9回1節，例題Ⅲ.5］．

②変えたいデータの組をデータテキストに用意して一挙に計算を実行する［第9回2節，例

表V-4 例題の計算時間(2) 単位［s］

| ステップ / 例題 | コンパイル | ゴー | グラフ |
|---|---|---|---|
| 5.1 | 4 | 2 | — |
| 5.2 | 6 | 28 | — |
| 6.5 | 55 | 28(12)* | — |
| Ⅰ.1 | 4 | 2 | 30 |
| Ⅰ.2 | 4 | 25(5)* | 28 |
| Ⅰ.4 | 4 | 8 | 27 |
| Ⅱ.2 | 5 | 34 | 22+23** |
| Ⅱ.5 | 31 | 290 | 34 |
| Ⅲ.4 | 7 | 11 | 31 |
| Ⅲ.6 | 23 | 75 | 28+31** |
| Ⅳ.5 | 12 | 115 | 24+24** |

［IBM5540数値演算プロセッサ付き］

*（ ）内は［計算結果］表示を省略した場合

**グラフを重ね書きまたは2枚描いている

題4.5，9.1］．

③変数の値を等間隔で変えるにはREPEAT文で繰り返す［例題Ⅰ.1，Ⅰ.2，Q43］．

場合により使い分ければよい．

**【Q57】** 計算時間はどのくらいか

**【A】** 代表的な例題について実測した結果を表V-4に示した．この値は機種によってもまた数値演算プロセッサの有無によっても異なるがおよその目安となろう．［表6-12］にも実測値を示してあるが，バージョンⅡでは実行速度面での改善もある．なお，数値演算プロセッサを付加すると実行速度は最高10倍位期待できる．

**【Q58】** BASICプログラムより速いか

**【A】** 直接比較はできないが実行速度はEQUATRAN-Mの方が速い．要は計算を実行している時間よりプログラミングで悩んでいる時間の方が断然長いということで，問題の解答を得るまでの時間はプログラミング不用のEQUATRAN-Mの方が圧倒的に短い［例題4.8］．

**【Q59】** Toy problemしか解けないか

**【A】** 例題に掲げた問題は比較的小さなものなので，この程度の問題しか解けないかという質問だと思う．解説の分かりやすさと紙面の制約を配慮して，小型ながら実用的な例題をできるだけ数多く載せたつもりである．実際には変数の数が数百の問題とか繰り返し変数が数十の問題を解いているので十分実用に耐えると言えよう．

**【Q60】** 数式処理との違いは

**【A】** 数式処理は記号演算を使って行列演算，方程式の求解，常微分方程式の求解，積分などができ，結果は式の形で得られる（解析解）．ただし，一次方程式以外の連立方程式は一般的には解けない．EQUATRAN-Mは式の形での演算はできないが単一の方程式はもちろん連立型の一次方程式，非線形方程式，常微分方程式やそれらの混合形のいずれでも数値的に扱うことができる（数値解）．理論式を導出するときなどには数式処理が役に立ち，工学的に要求される数値解を求めるにはEQUATRAN-Mが便利である．

**【Q61】** 化学工学便覧のソフトとどう違うのか

**【A】** パソコンによる化学工学便覧情報システム（PRODECE）は化学工学の代表的な計算法をプログラム化したものである．物性定数との連結や図化出力に工夫がなされており，使用目的に合うプログラムがあれば便利である．しかし，その一部を変更したいときや適当なプログラムがないときはお手上げである．一方，EQUATRAN-Mでは方程式や定数をユーザーが与えなければならないが，それさえ分かっていればたいていの問題に対応でき，Q1に述べたように追加・変更が自在で結果のグラフ表示もきれいにできる．本連載では読者がケミカルエンジニアであることから，例題を化学工学便覧から多数引用しているので誤解されている面があるかもしれないが，EQUATRAN-Mの適用範囲は化学工学の問題に限らずあらゆる工学分野の問題にも及ぶものである［例題8.1，8.5］

**【Q62】** 他に類似のパソコンソフトはないか

**【A】** はっきり言ってこれだけ広いスパンの機能を持ったパソコンソフトは世界的にもない．しかし，部分的に類似のソフトはいくつかある．広義の代数方程式解法ソフトとしてはTK！SOLVER[5]とEUREKA THE SOLVER[6]がある．最適化の機能を特徴としたものにGINO/PC[7]がある．また，常微分方程式の解法ソフトとして

表V-5　方程式解法パソコンソフト

| ソフト名 | エディタ（＊） | 行列表現 | 代数方程式 | 常微分方程式 | 最適化問題 | グラフ（＊＊） | 問題の規模 | 開　発　元 |
|---|---|---|---|---|---|---|---|---|
| EQUATRAN-M | s | ○ | ○ | ○ | ○ | C/G | 1000 | 三井東圧化学㈱ |
| TK!SOLVER | l | △ | ○ | — | — | C | 100 | Software Arts Inc. |
| EUREKA | s | — | ○ | — | ○ | C | 20 | Borland Inc. |
| GINO/PC | l | — | ○ | — | ○ | — | 50 | Chicago大学 |
| ACSL-PC＊＊＊ | — | △ | — | ○ | — | G | 100 | MGA Icn. |

＊内蔵のもので，sは画面エディタ，lは行エディタ

＊＊Cはキャラクタグラフ，Gはグラフィックグラフ

＊＊＊エディタは内蔵せず．FORTRANソースプログラムを発生する．

表V-6　EQUATRAN-M文献一覧(1986年5月以降)

〈1〉小口，佐渡友，林田，横山，大村，工東，須藤：化学工業における数値計算，information, Vol.5，No.8，p.54, No.9，p.62, No.10, p.105, No.11, p.46, No.12, p.82 (1986)，Vol.6，No.1，p.112 (1987)
〈2〉小口，宮原：オペレーションズ・リサーチ，Vol.31, p.705 (1986)
〈3〉佐渡友：プロセス設計・プラント運転解析への方程式解法言語の適用，「プラントオペレーション工学」(化学工学協会編)，p.53, 海文堂 (1987)
〈4〉宮原，佐渡友，須藤，小口：続連載「EQUATRAN-M」技術計算用方程式解法言語，ケミカルエンジニヤリング, Vol.32, p.813, p.886 (1987)，Vol.33, p.97, p.405 (1988)
〈5〉宮原，山田：技術計算用簡易言語と化学工業への適用, MOL, Vol.25, No.4，p.83 (1987)
〈6〉三井東圧EQM研究会：「EQUATRAN-M入門」，省エネルギーセンター (1987)
〈7〉小口：分離技術，投稿中

はACSL-PC[8)]が知られている．これらのソフト間の機能比較をしたのが表V-5である．文献5，6)にEQUATRAN-M (旧バージョン) との詳細な比較がされているので参考になろう．

**【Q63】** EQUATRAN-Mの参考文献は

**【A】** 前連載の最後にその時点 (1986年4月) までに発表された文献一覧が載せてある [表9-12]．その後発表された文献を表V-6に掲げておく．特に，「EQUATRAN-M入門」はいろいろな工学分野の例題を豊富に取り入れ，構成も読みやすく配慮されているので絶好の入門書と言えよう．

## おわりに

前連載第1回から3年近く経つが，日本でも方程式解法言語の分野はかなり一般的になってきた．類似ソフトが目にとまるようになってきたのはその表れであり，EQUATRAN-M自身も随分大勢の方に活用されている．これを使用した研究が学会で報告されたり，企業での利用成果が聞かれるまでになった．もし，「EQUATRAN-M文化」なるものが生まれ育つのにこの連載記事が役立つとすれば，筆者らの喜びこれに過ぎるものはないと念じつつ本連載を終える．

**参考文献**

1) 宮原他：ケミカルエンジニヤリング，Vol.30, p.559, p.631, p.705, p.798, p.868 (1985), Vol.31, p.86, p.161, p.245, p.317 (1986).
2) 平田監訳：「化学技術者のための応用数学」，p.38, 丸善.
3) 矢木，西村：「化学プロセス工学」，丸善 (1969).
4) D.Q.Kern：「Process heat transfer」，p.144, Mc Graw-Hill (1950).
5) 宮原，林田，須藤：分離技術, Vol.15, p.281 (1985).
6) 伊東：分離技術，Vol.17, p.409 (1987).
7) 青沼：オペレーションズ・リサーチ，Vol.31, p.49 (1986).
8) 中西：システムと制御，Vol.29, p.706 (1985).