方程式解法ソフトEQUATRAN-M/G

横山克己*

数値計算の簡易言語

EQUATRANは数値計算をプログラミングせずに簡便に行うソフトで，いわば「数値計算の簡易言語」です．もう少し正確にいうと，方程式を通常の表記でそのまま入力するだけで，直接数値解を得ることができます．

たとえば，図1を見てください．端的に理解してもらえるように，5元連立方程式をもってきました．このように方程式を入力したのち実行を指示すると，瞬時に図の下のような計算結果を得ることができるのです．

このようなソフトは「方程式解法ソフト」と呼ばれています．同種のソフトは国内では唯一ですが，米国ではTK！SolverやEureka：The Solver，MathCADなどがあり，すでに技術計算分野で一般的に利用されるまでになってきています．日本においても，技術者・研究者の生産性向上のために，このようなツール(道具)の活用がますます盛んになってくるのではないでしょうか．

```
1:     /*      5 元 連 立 方 程 式        */
2:
3:       A +    B -    C + 2*D +    E = 7
4:       A + 6*B + 2*C - 3*D         = 12
5:     2*A -    B + 5*C -    D + 3*E = -2
6:      -A - 2*B +    C         - 2*E = 3
7:       A +    B - 5*C +    D - 4*E = 1
8:
9:       OUTPUT  A, B, C, D, E


<<    計 算 結 果     >>
A                  =   -1.264706
B                  =   3.738754
C                  =   3.624567
D                  =   5.472318
E                  =   -2.794118
```

図1　5元連立方程式の計算

*三井東圧化学株式会社

EQUATRANには，現在EQUATRAN－Mと，その上位版であるEQUATRAN－Gがあります．EQUATRAN－Mは，1985年にパソコン用としてはじめて登場したもので，改良を重ねて今日に至っています．いわばEQUATRANの基本バージョンといえます．これに対してEQUATRAN－Gは，EQUATRAN－Mをもとに機能アップしたもので，1989年にワークステーション用として販売を開始しました．現在EQUATRANは，パソコンとワークステーションで利用可能です．

EQUATRANと他のシステムとの違い

EQUATRANは，蒸留計算ソフトのような単機能のアプリケーションソフトと，FORTRANやBASICのようなプログラミング言語との中間に位置付けられます．

【アプリケーションソフトとの違い】

アプリケーションソフトは，ある特定の計算をするように作られているため，いま計算しようとしている問題に合うプログラムがあれば，それを使うのがベストでしょう．しかし，ぴったりしたプログラムがあることはまれですし，似たものがあったとしても要求を満たさない場合は変更が利きません．これに対してEQUATRANは，問題の数式モデルさえ入力すれば答が得られるので，どんな問題でも使うことができます．さらに，問題の変更に柔軟に対応できるのです．

【プログラミング言語との違い】

一方，プログラミング言語とはどこが違うのか，図2で説明しましょう．これは，問題を解く場合のアプローチの違いを示しています．

右側がFORTRANやBASICの場合です．数式モデルを作成したら，まず計算の手続き，すなわ

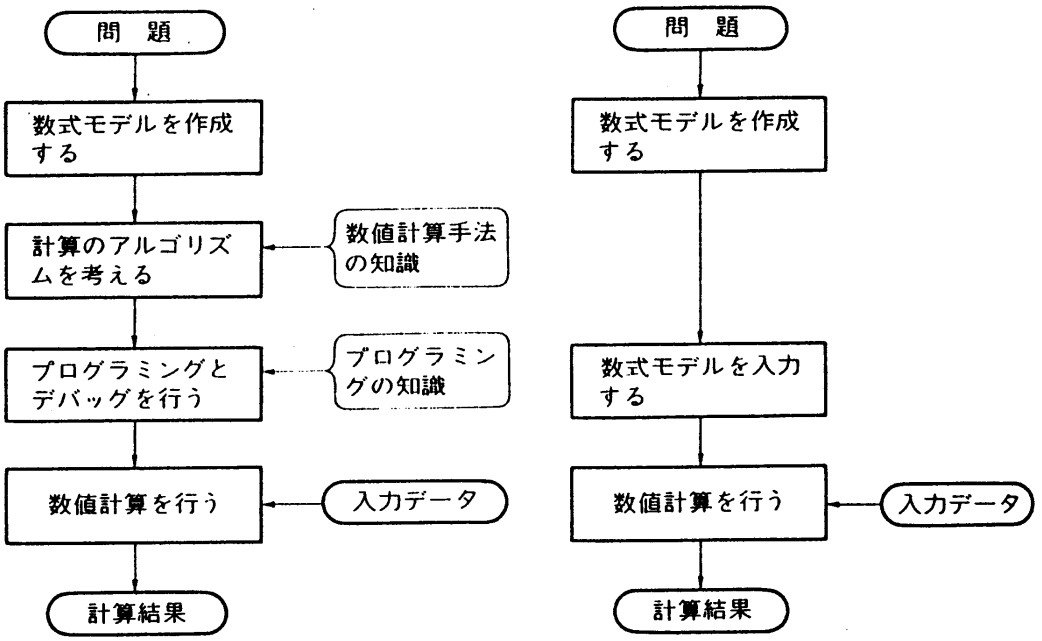

図2　問題解決の手続き

ちアルゴリズムを考えます．このとき，線形連立方程式や非線形方程式，常微分方程式などが含まれる場合は，それぞれの数値計算の手法(たとえば，常微分方程式を解くのであればルンゲ・クッタ法など)の知識が必要になります．次にそのアルゴリズムにしたがってFORTRANやBASICによりプログラムを作成し，正しく計算できるように誤りを直します(デバッグします)．今度はその言語の文法などプログラミングの知識が要求され，また多くの時間と労力が必要となります．

これに比べ右側のEQUATRANの場合は，一番厄介な部分であるアルゴリズムを考える段階と，プログラミングおよびデバッグの段階を自動化してくれます．数式モデルを正しく入力しさえすればよいのです．したがって，ほしいときに答えが得られると同時に，各人が抱えている本来の専門の仕事に専念できるわけです．

EQUATRANの基本機能

まずEQUATRANの基本機能，すなわちEQUATRAN－Mの機能について説明し，その後にEQUATRAN－Gで拡張・強化された点を解説することにします．

【方程式解法機能・記述機能】

EQUATRAN－Mでは，

・線形連立方程式
・非線形連立方程式
・常微分方程式(高階または非線形を含む)
・最適化計算

を数値的に解くことができます．方程式は通常の数学的表記法により入力すればよく，変形したり解く順序に並び換える必要はありません．また，これらの複合した問題も取り扱えます．

線形・非線形連立方程式の場合は，図1に示したようにそのまま入力すればよいのです．特に非線形の場合は，一般に直接解くことができないので，繰り返し収束計算が必要になりますが，EQUATRAN－Mでは線形部分や非線形部分を判断して，それぞれの計算手法を自動的に組み込みます．なお，ユーザーが繰り返し収束計算の指定をすることも可能です．

常微分方程式では，微分項はアポストロフィ(')を微分記号として

$$\frac{dx}{dt} \rightarrow x', \quad \frac{d^2x}{dt^2} \rightarrow x''$$

と書き表せばよいので，高階の方程式でも扱うことができます．

最適化計算とは，評価式の値が最大あるいは最小となるように，1つあるいは2つ以上の独立変数の値を求める問題ですが，これも独立変数と評

価変数を指定するだけです.

このほか，数式モデルを簡潔に記述するために以下のような豊富な記述機能を持っています.

・配列変数……1次元および2次元の配列変数が扱えます.
・組み込み関数……対数，指数，三角関数など36個の関数が内蔵されています.
・数表……変数間の関係が図や表として与えられている場合に，数表として定義し，方程式の中で関数のように利用できます.
・条件付きの式……条件によって場合分けされるような式を表現できます.
・パラメータ・マクロ……数式モデルをモジュール化するなど，記述に融通性を持たせることができます.

【グラフ作成機能】

片対数・両対数グラフ・スプライン曲線による補間，直線，2次・3次曲線近似など，科学技術グラフに必要な機能を豊富に備えているほか，各種の編集機能があるので完成度の高いグラフが作成できます. 画面に表示されたグラフはそのままプリンタにハードコピー可能です.

計算結果を即座にグラフ化して評価できるこの機能は，強力な方程式解法機能と相まって相乗効果をもたらします.

EQUATRAN－Gの拡張機能

EQUATRAN－Gでさらに追加・強化された機能を説明します.

【大規模問題への対応】

EQUATRAN－Mでは，一度に解ける方程式の数が数百程度に制限されています(パーソナル版では100). たとえば，化学プロセスの物質収支・熱収支計算では，[成分数×ストリーム数]オーダーの方程式が必要で，この制約に引っ掛かることがあります. これは，MS－DOSで利用できるメモリが640KBに制限されているからで，最近はEMSなどさまざまな工夫がされていますが，抜本的な解決になっていません. EQUATRAN－Mも残念ながらこの制約を受けることになります.

これに対してEQUATRAN－Gでは，メモリの制限は事実上ありませんので，方程式の数が1000を越えるような，かなり大規模な問題でも取り扱えます.

【ユーザー関数機能】

EQUATRANで記述したソーステキストをユーザー関数として定義して利用することができます.

この利点は2つあります. 1つは，大規模な問題をいくつかのモジュールに分割して作成でき，ソーステキストの見通しをよくします. また，汎用性のあるユーザー関数はライブラリにして共用することができます.

もう1つは，メインからユーザー関数を，あるいはユーザー関数からユーザー関数を呼び出すことができますので，繰り返し計算，最適化計算および積分計算を多重化できます. EQUATRAN－Mでは，積分計算のループは1つのみで多重化はできませんし，積分計算の外側で繰り返し計算や最適化計算はできません. これに対してEQUATRAN－Gでは，積分計算を多重化，すなわち多重積分や，積分計算の結果によって常微分方程式の初期条件を修正し，繰り返し収束計算をするような2点境界値問題，さらに積分計算の結果をみてパラメータを最小2乗法により決定するような問題などが扱えます.

【数値計算手法の強化】

非線形最小2乗法の手法としてマルカート法が組み込まれていますので，比較的利用頻度が高い最小2乗問題が高速に精度よく計算できます.

また，常微分方程式の解法として2つの陰解法(後方オイラー法，台形公式)が追加されています. 今まで積分きざみをかなり細かくとらなければいけなかったような問題(stiffな問題)でも高速に安定して解けます.

【不連続現象の記述】

連続系のシミュレーションで，任意の条件の発生時に，定数の値や被積分変数の値を設定しなおす機能が追加されましたので，ボールの衝突問題などの記述ができます. 例題で取り上げた回分蒸留計算で，タンク切替え操作の記述に使用しています.

応 用 分 野

EQUATRANは現在約1500セットが使われており，その応用分野は化学，機械，電気，制御などの各工学分野はもちろんのこと，医学，薬学，経

済学などにまで及んでいます．数式モデルを用いる分野であればどんなところでも適用することができます．このように汎用的なツールですが，特に化学工業分野では適応例が数多くあり，たとえば以下のような問題です．

・物質収支・熱収支計算
・気液平衡計算，化学平衡計算
・制御系の解析と設計
・連続系の動的シミュレーション
・反応速度の検討
・実験データの回帰式決定とグラフ化

具体的には，文献[1~3]に多く例題が紹介されています．

例　題（回分蒸留計算）

ここではEQUATRAN－Gを用いた適用事例として，回分蒸留の詳細モデルの計算を取り上げます．なお，回分蒸留については数式モデルを含めて文献[4]にくわしい解説があります．

この問題は，多数の常微分方程式を解く非定常問題で，同時に繰り返し収束計算が組み合わされており，ワークステーションクラスの計算機パワーを必要とします．

> メタノール，エタノールおよび水の3成分混合物から，20段（コンデンサ，リボイラを含む）の蒸留塔で回分蒸留によりメタノールとエタノールを回収したい．仕込み量は20[kgmol]，組成（モル分率）は0.3, 0.2, 0.5とする．操作条件は，全還流運転後の定常状態を出発状態として，還流比10で運転し，5.5時間後にタンク切替えを行うものとする．

【数式モデル】

図3のような段数Nの蒸留塔で，コンデンサ，第j段およびリボイラでの全物質収支，成分物質収支，熱収支を考えます．なお，リボイラ以外の液ホールドアップU_j($j=1, 2,\cdots, N-1$)は一定とみなします．また，ここで使用する記号は表1を参照ください．

・コンデンサ（全縮器）まわり

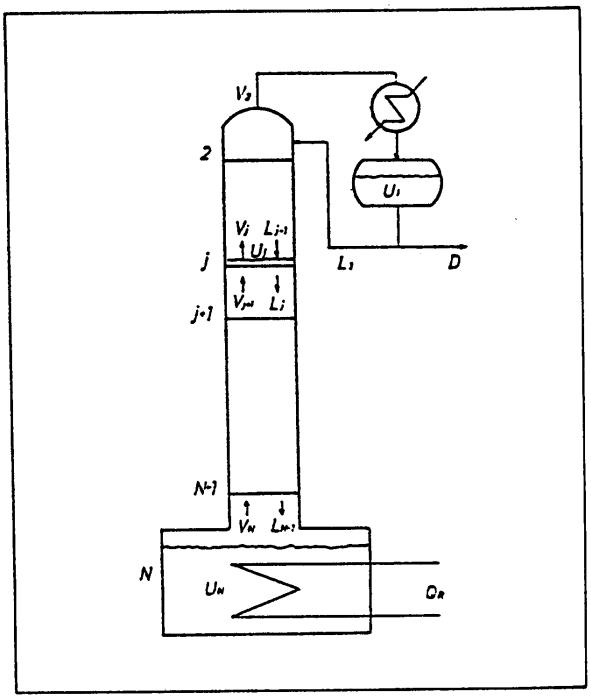

図3　回分蒸留塔

$$-D-L_1+V_2=0 \tag{1}$$

$$U_1\frac{dx_{1i}}{dt}=-(D+L_1)x_{1i}+V_2y_{2i} \tag{2}$$

$$U_1\frac{dh_1}{dt}=-(D+L_1)h_1+V_2H_2-Q_C \tag{3}$$

・第j段まわり（$j=2, 3,\cdots,N-1$）

$$L_{j-1}-V_j-L_j+V_{j+1}=0 \tag{4}$$

$$U_j\frac{dx_{ji}}{dt}=L_{j-1}x_{j-1,i}-V_jy_{ji}-L_jx_{ji}+V_{j+1}y_{j+1,i} \tag{5}$$

$$U_j\frac{dh_j}{dt}=L_{j-1}h_{j-1}-V_jH_j-L_jh_j-V_{j+1}H_{j+1} \tag{6}$$

・リボイラまわり

$$\frac{dU_N}{dt}=L_{N-1}-V_N \tag{7}$$

$$U_N\frac{dx_{Ni}}{dt}+x_{Ni}\frac{dU_N}{dt}=L_{N-1}x_{N-1,i}-V_Ny_{Ni} \tag{8}$$

$$U_N\frac{dh_N}{dt}+h_N\frac{dU_N}{dt}=L_{N-1}h_{N-1}-V_NH_N+Q_R \tag{9}$$

気液平衡からx_{ji}とy_{ji}の関係は，気液平衡比K_{ji}により

$$y_{ji}=K_{ji}x_{ji} \tag{10}$$

K_{ji}およびエンタルピh_j, H_jは圧力P_j, 温度T_j, 組

成x_{ji}およびy_{ji}の関数で

$$K_{ji}=K(P_j, T_j, x_{ji}, y_{ji}) \quad (11)$$

$$h_j=h(P_j, T_j, x_{ji}) \quad (12)$$

$$H_j=H(P_j, T_j, y_{ji}) \quad (13)$$

と表現でき，またT_jは次式を満足するように沸点計算により求められます．

$$\sum_{i=0}^{n} y_{ji}=1 \quad (14)$$

以上の方程式を連立されて解くことにより，回分蒸留の非定常解が得られます．

【ソーステキスト】

数式モデルから作成したソーステキストを図4に示します．ここでは，細かな文法の説明は省き，大まかな意味の説明にとどめておきます．

3行目のINCLUDE文は他のファイル(ここではb_func. eqs)をここに挿入することを指定するもので，このファイルには後で説明する気液平衡比やエンタルピを計算するユーザー関数，そして沸点計算をするユーザー関数が含まれています．

7～25行目は使用している変数の定義であり，27行目からは3つのマクロが定義されています．

```
 1:  /*  回分蒸留計算  */
 2:
 3:  INCLUDE b_func
 4:
 5:  LOCAL  N=20   /* 段数 */
 6:
 7:  GLOBAL VAR  ..
 8:       L(N)     "液流量                [kgmol/h]"
 9:      ,V(N)     "蒸気流量              [kgmol/h]"
10:      ,U(N)     "液ホールドアップ      [kgmol]"
11:      ,x(N,3)   "液モル分率                [-]"
12:      ,y(N,3)   "蒸気モル分率              [-]"
13:      ,T(N)     "温度                    [℃]"
14:      ,P(N)     "圧力                  [mmHg]"
15:      ,h(N)     "液エンタルピ    [kcal/kgmol]"
16:      ,H(N)     "蒸気エンタルピ  [kcal/kgmol]"
17:      ,QC       "コンデンサ除熱量    [kcal/h]"
18:      ,QR       "リボイラ加熱量      [kcal/h]"
19:      ,D        "留出量              [kgmol/h]"
20:
21:  VAR  sDx(3)
22:      ,pr       "製品留出量            [kgmol]"
23:      ,x_pr(3)  "製品液モル分率            [-]"
24:      ,R        "還流比                    [-]"
25:      ,t        "時間                      [h]"
26:
27:  MACRO COND    /* コンデンサまわり */
28:     -D - L(1) + V(2) = 0
29:     U(1)*x'(1) = -(D+L(1))*x(1) + V(2)*y(2)
30:     U(1)*DERIV(h(1),0)
31:       = -(D+L(1))*h(1) + V(2)*H(2) -QC
32:     BUBPT( P(1), x(1), y(1), T(1) )
33:     H_val( T(1), x(1), y(1), h(1), H(1) )
34:  END COND
35:
36:  MACRO TRAY    /* 第 j 段まわり */
37:     L(j_1) - V(j) - L(j) + V(j1) = 0
38:     U(j)*x'(j) = L(j_1)*x(j_1) - V(j)*y(j) ..
39:                  - L(j)*x(j) + V(j1)*y(j1)
40:     U(j)*DERIV(h(j),0) = L(j_1)*h(j_1) - V(j)*H(j) ..
41:                  - L(j)*h(j) + V(j1)*H(j1)
42:     BUBPT( P(j), x(j), y(j), T(j) )
43:     H_val( T(j), x(j), y(j), h(j), H(j) )
44:  END TRAY
45:
46:  MACRO BOTM    /* リボイラまわり */
47:     U'(n) = L(n_1) - V(n)
48:     U(n)*x'(n) + x(n)*U'(n)
49:       = L(n_1)*x(n_1) - V(n)*y(n)
50:     U(n)*DERIV(h(n),0) + h(n)*U'(n)
51:       = L(n_1)*h(n_1) - V(n)*H(n) + QR
52:     BUBPT( P(n), x(n), y(n), T(n) )
53:     H_val( T(n), x(n), y(n), h(n), H(n) )
54:  END BOTM
55:
```

```
56:  CALL COND( )
57:  CALL TRAY( j= 2, j_1= 1, j1= 3 )
58:  CALL TRAY( j= 3, j_1= 2, j1= 4 )
59:  CALL TRAY( j= 4, j_1= 3, j1= 5 )
60:  CALL TRAY( j= 5, j_1= 4, j1= 6 )
61:  CALL TRAY( j= 6, j_1= 5, j1= 7 )
62:  CALL TRAY( j= 7, j_1= 6, j1= 8 )
63:  CALL TRAY( j= 8, j_1= 7, j1= 9 )
64:  CALL TRAY( j= 9, j_1= 8, j1=10 )
65:  CALL TRAY( j=10, j_1= 9, j1=11 )
66:  CALL TRAY( j=11, j_1=10, j1=12 )
67:  CALL TRAY( j=12, j_1=11, j1=13 )
68:  CALL TRAY( j=13, j_1=12, j1=14 )
69:  CALL TRAY( j=14, j_1=13, j1=15 )
70:  CALL TRAY( j=15, j_1=14, j1=16 )
71:  CALL TRAY( j=16, j_1=15, j1=17 )
72:  CALL TRAY( j=17, j_1=16, j1=18 )
73:  CALL TRAY( j=18, j_1=17, j1=19 )
74:  CALL TRAY( j=19, j_1=18, j1=20 )
75:  CALL BOTM( n=N, n_1=N-1 )
76:
77:     R = L(1)/D
78:     U1 = U(1);    U(2:N-1) = Uj
79:     V(1) = 0;     L(N) = 0
80:
81:  /* 操作条件 */
82:     QR = 1e5
83:     U1 = 2;  Uj = 0.1
84:     P = ( 760, 765, 770, 775, 780,
85:           785, 790, 795, 800, 805,
86:           810, 815, 820, 825, 830,
87:           835, 840, 845, 850, 855 )
88:     R = 10
89:     x # ( 0.99867, 0.00133, 0.00000 ) ..
90:         ( 0.99823, 0.00177, 0.00000 ) ..
91:         ( 0.99765, 0.00235, 0.00000 ) ..
92:         ( 0.99686, 0.00313, 0.00001 ) ..
93:         ( 0.99580, 0.00418, 0.00002 ) ..
94:         ( 0.99435, 0.00561, 0.00004 ) ..
95:         ( 0.99235, 0.00756, 0.00009 ) ..
96:         ( 0.98954, 0.01026, 0.00020 ) ..
97:         ( 0.98552, 0.01406, 0.00042 ) ..
98:         ( 0.97956, 0.01952, 0.00092 ) ..
99:         ( 0.97044, 0.02758, 0.00198 ) ..
100:        ( 0.95589, 0.03989, 0.00422 ) ..
101:        ( 0.93177, 0.05932, 0.00891 ) ..
102:        ( 0.89088, 0.09071, 0.01841 ) ..
103:        ( 0.82305, 0.14032, 0.03663 ) ..
104:        ( 0.72051, 0.21079, 0.06870 ) ..
105:        ( 0.58830, 0.29172, 0.11998 ) ..
106:        ( 0.44490, 0.35651, 0.19859 ) ..
107:        ( 0.30580, 0.36376, 0.33044 ) ..
108:        ( 0.16129, 0.23405, 0.60466 )
109:     U(N) # 17.4
110:
111:  /* 製品 */
112:     sDx' = D*x(1);  sDx # 0
113:     pr = SUM( sDx )
114:     x_pr = sDx / pr   WHEN pr >  0  ..
115:          = 0          WHEN pr <= 0
116:
117:  /* タンク切替え */
118:  VAR  tank(2)     "各タンクの製品留出量  [kgmol]"
119:      ,x_tank(2,3) "各タンクの製品液モル分率  [-]"
120:      ,t_tank(2)   "タンク切替え時刻          [h]"
121:
122:     t_tank = ( 5.5, 10 )
123:     tank # 0;  x_tank # 0
124:
125:  CHANGE ( tank(1)#pr, x_tank(1)#x_pr, sDx#0 )  ..
126:                                    ON (t>=t_tank(1))
127:  CHANGE ( tank(2)#pr, x_tank(2)#x_pr, sDx#0 )  ..
128:                                    ON (t>=t_tank(2))
129:
130:  /* 積分計算の指定 */
131:  INTEGRAL  t[0,10] STEP 0.0005 BY RKV
132:
133:  /* 出力の指定 */
134:  TREND     D, U(N), T(1), T(N), x(1,1), x(1,2),
135:            pr, x_pr STEP 1
136:  OUTPUT    U(N), tank, x_tank
137:  OUTPUT1   t, x(1,1), x(1,2), T(1), T(N) STEP 0.05
```

図4　回分蒸留計算のソーステキスト

マクロはまとまった方程式群を定義しておき，後から繰り返し呼び出して記述を簡略化します．3つのマクロには，それぞれコンデンサ，第j段そしてリボイラまわりについて，収支式((1)式～(9)式)の記述と，沸点計算BUBPTおよびエンタルピ計算H　valのユーザー関数の呼び出しが含まれ

ます．なお，dh/dtの項は，微分値を計算する組み込み関数DERIVを使用して計算しています．

56行目～75行目でそれらのマクロの呼び出しが行われています．

82～109行目には，操作条件を記述しています．被積分変数であるxおよび$U(N)$には初期条件を指定してあり，xには後で説明する全還流定常状態の計算結果から得られた値を使用しています．

118行目～128行目ではタンクの切替え操作を記述しています．5.5時間でタンク切り替えを指示しており，それまでを第1タンクに，それからさらに10時間までを第2タンクに留出するとして，tankとx_tankに留出量と液モル分率を保持しています．

131行目は積分計算の指定で，134行目以降は出力関係の指定です．

【ユーザー関数】

先のINCLUDE文が読み込んでいたソーステキストを**図5**に示しました．これには(11)～(13)式に相当する気液平衡比K_{ji}およびエンタルピh_j, H_jの計算のユーザー関数，そして沸点計算のユーザー関数が含まれます．

気液平衡比を計算するユーザー関数K_val(3行目～26行目)では，Kは

$$K=\frac{p}{P}\gamma \qquad (15)$$

により求めます．ここで蒸気圧pはAntoine式で，活量係数γはWilson式で計算しています．

エンタルピを計算するユーザー関数H_val(28行目～48行目)では，各成分の組成の2次式で近似した式を用いて計算しています．

そして，沸点計算をするユーザー関数がBUBPT(50行目～61行目)です．

【計算結果】

出力結果を**図6**に示しました．はじめの部分はトレンド出力で，時間あたりの留出量，塔底の残量，塔内温度，留出組成，製品の量や組成を，時間経過にしたがって出力してあります．

計算結果は計算終了時(10時間後)の塔底の残量と，各タンクの製品留出量とその組成を示しています．これから，第1タンクには約89%のメタノールが，第2タンクには約62%のエタノールがそれぞれおよそ5.8kgmol，4.4kgmol回収できることが分かります．

```
1:  /*  回分蒸留計算用ユーザー関数  */
2:
3:  FUNCTION  K_val( P, T, x; K )
4:       VAR  P          "圧力                          [mmHg]"
5:           ,T          "温度                           [℃]"
6:           ,x(3)       "液モル分率                     [-]"
7:           ,K(3)       "気液平衡比（K値）              [-]"
8:           ,p(3)       "蒸気圧                        [mmHg]"
9:           ,g(3)       "活量係数（γ）                  [-]"
10:      VAR  a(3) = ( 7.87863, 8.04494, 7.96681 )
11:          ,b(3) = ( 1473.11, 1554.30, 1668.21 )
12:          ,c(3) = ( 230.00,  222.65,   228.00 )    ,
13:                  "Antoine定数                          "
14:      VAR  lam(3,3) = ( 1,        2.30756, 0.43045 ),
15:                      ( 0.20397, 1,        0.21618 ),
16:                      ( 0.94934, 0.79133, 1       ),
17:                      "Wilson式の定数  Λij          "
18:
19:      K = p / P * g
20:   /* Antoine式 */
21:      LOG10( p ) = a - b/(T+c)
22:   /* Wilson式 */
23:  eq: LOGE( g ) = -LOGE(SUM(x*lam)) + 1    ,;
24:                   - SUM((x*lam~)/SUM(x*lam))
25:      RESET  g#1[0,10]  BY eq
26:  END
27:
28:  FUNCTION  H_val( TC, x, y; h, H )
29:       VAR  x(3)       "液モル分率                     [-]"
30:           ,y(3)       "蒸気モル分率                   [-]"
31:           ,TC         "温度                           [℃]"
32:           ,h          "液エンタルピ     [kcal/kgmol]"
33:           ,H          "蒸気エンタルピ   [kcal/kgmol]"
34:           ,hi(3)
35:           ,Hi(3)
36:       VAR  a(3)=(-9.2643E+02,-1.4605E+03,-3.2698E+03)
37:           ,b(3)=( 1.4706E+00, 2.5901E+00, 1.7297E+01)
38:           ,c(3)=( 3.8075E-02, 5.0548E-02, 4.2021E-03)
39:       VAR  A(3)=( 9.6570E+03, 1.0311E+04, 1.0707E+04)
40:           ,B(3)=( 5.1470E+00, 7.6532E+00, 6.9025E+00)
41:           ,C(3)=( 9.4386E-03, 1.6791E-02, 1.6256E-03)
42:
43:       h  = SUM( hi*x )
44:       hi = a + b*T + c*T~2
45:       H  = SUM( Hi*y )
46:       Hi = A + B*T + C*T~2
47:       T  = TC + 273.15
48:  END
49:
50:  FUNCTION BUBPT( P, x; y, T )
51:       VAR  P          "圧力                          [mmHg]"
52:           ,x(3)       "液モル分率                     [-]"
53:           ,y(3)       "蒸気モル分率                   [-]"
54:           ,T          "温度                           [℃]"
55:           ,K(3)       "気液平衡比（K値）              [-]"
56:
57:       y = K*x
58:       K_val( P, T, x, K )
59:  eq: SUM( y ) = 1
60:       RESET T # 80[50,110] BY eq
61:  END
```

図5　回分蒸留計算用ユーザー関係

また，**図7**は留出液中のメタノールとエタノールの組成と，塔頂と塔底の温度の時間変化をグラフ化したものです．

【全還流定常状態の計算】

全還流運転での定常状態を計算するソーステキストを**図8**に示しておきます．図4のソーステキストをもとに微係数の項を0と置き(70行目)，各段のホールドアップと仕込み液との収支式を追加する(74～78行目)など，一部修正することで作成が可能です．なお，簡単のために熱収支を省略してあります．

最後に

回分蒸留の事例のように，EQUATRANでダイ

```
/*  回分蒸留計算  */
```

t	1:D	2:U(20)	3:T(1)	4:T(20)	5:x(1,1)
	6:x(1,2)	7:pr	8:x_pr(1)	9:x_pr(2)	0:x_pr(3)
0	1.064335	17.40000	64.75907	81.38861	0.9986700
	0.001330000	0	0	0	0
1.000000	1.061898	16.33776	64.77497	82.29585	0.9961854
	0.003802997	1.062243	0.9976784	0.002317293	4.35383e-06
2.000000	1.061208	15.27615	64.80633	83.36030	0.9914403
	0.008489785	2.123849	0.9959107	0.004071588	1.76828e-05
3.000000	1.057415	14.21620	64.97046	84.44120	0.9691383
	0.02966545	3.183799	0.9918958	0.007971351	0.0001328706
4.000000	1.036062	13.16784	66.31138	85.59500	0.8296291
	0.1541915	4.232162	0.9725152	0.02572114	0.001763693
5.000000	1.010686	12.14490	69.02055	87.07113	0.6165918
	0.3400621	5.255095	0.9239525	0.06887179	0.007175671
6.000000	0.9899622	11.14749	71.73536	89.56920	0.4317928
	0.4996092	0.4974903	0.4747349	0.4626457	0.06261939
7.000000	0.9654206	10.16869	73.88294	95.87235	0.2898923
	0.6211420	1.476285	0.3969614	0.5294301	0.07360848
8.000000	0.9747997	9.202245	75.39373	103.3304	0.1884270
	0.7064410	2.442732	0.3333392	0.5837461	0.08291470
9.000000	0.9657794	8.231236	76.43172	103.3330	0.1224847
	0.7077700	3.413741	0.2818691	0.6232939	0.09483702
10.00000	0.9427573	7.276730	78.18345	103.3330	0.08004535
	0.4852181	0	0	0	0

```
<<   計算結果   >>
U(20)            =  7.27673          : 液ホールドアップ     [kgmol]
tank             =                   : 各タンクの製品留出量    [kgmol]
  1) 5.755023          2) 4.368247
x_tank           =                   : 各タンクの製品液モル分率    [-]
     (  1)               (  2)               (  3)
  1) 0.892968          0.09613929          0.01089275
  2) 0.2421981         0.6182322           0.1395697
```

図6　回分蒸留計算の計算結果

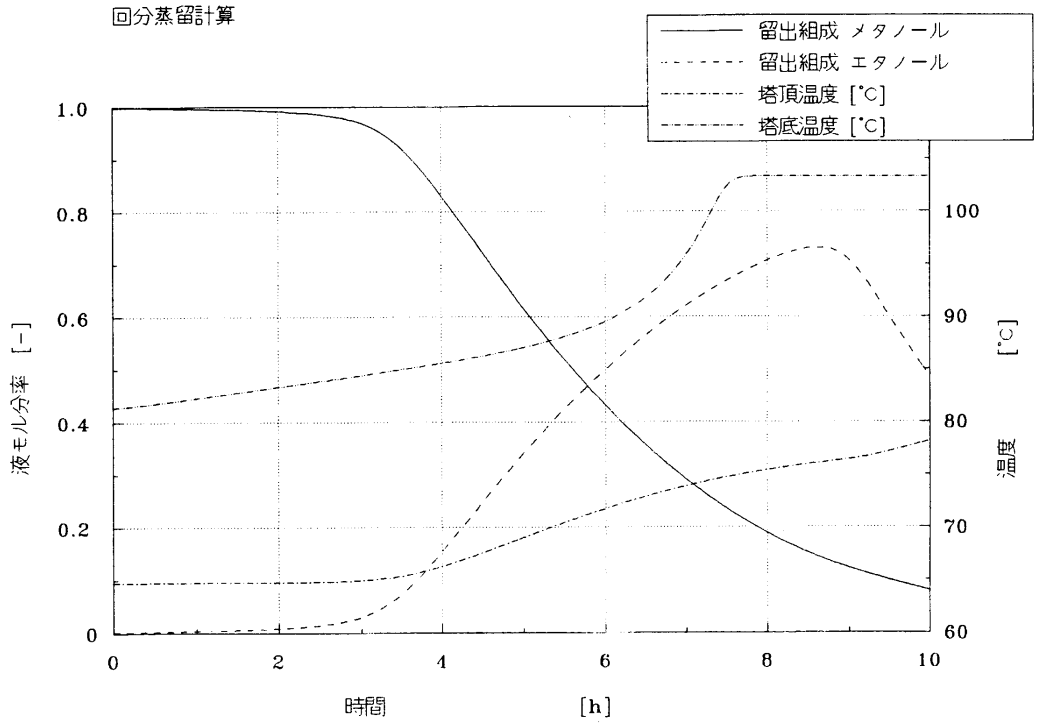

図7　回分蒸留計算の計算結果(グラフ化したもの)

```
/*  回分蒸留計算（全還流定常状態）  */

INCLUDE b_func

LOCAL  N=20   /* 段数 */

GLOBAL VAR  ..
      L(N)    "液流量                [kgmol/h]"
     ,V(N)    "蒸気流量              [kgmol/h]"
     ,U(N)    "液ホールドアップ       [kgmol]"
     ,x(N,3)  "液モル分率                [-]"
     ,y(N,3)  "蒸気モル分率              [-]"
     ,T(N)    "温度                    [℃]"
     ,P(N)    "圧力                   [mmHg]"
     ,QR      "リボイラ加熱量        [kcal/h]"
     ,D       "留出量                [kgmol/h]"

MACRO COND    /* コンデンサまわり */
   BUBPT( P(1), x(1), y(1), T(1) )
END COND

MACRO TRAY    /* 第ｊ段まわり */
e1:U(j)*x'(j) = L(j_1)*x(j_1) - V(j)*y(j) ..
                - L(j)*x(j) + V(j1)*y(j1)
   BUBPT( P(j), x(j), y(j), T(j) )
   RESET x(j,1)#0.6[0,1]   BY e1(1) MAXLOOP 200
   RESET x(j,2)#0.2[0,0.5] BY e1(2) MAXLOOP 200
END TRAY

MACRO BOTM    /* リボイラまわり */
e2:U(n)*x'(n) + x(n)*U'(n)
     = L(n_1)*x(n_1) - V(n)*y(n)
   BUBPT( P(n), x(n), y(n), T(n) )
   RESET x(n,1)#0.2[0,0.5] BY e2(1) MAXLOOP 200
   RESET x(n,2)#0.2[0,0.5] BY e2(2) MAXLOOP 200
END BOTM

CALL COND( )
CALL TRAY( j= 2, j_1= 1, j1= 3 )
CALL TRAY( j= 3, j_1= 2, j1= 4 )
CALL TRAY( j= 4, j_1= 3, j1= 5 )
CALL TRAY( j= 5, j_1= 4, j1= 6 )
CALL TRAY( j= 6, j_1= 5, j1= 7 )
CALL TRAY( j= 7, j_1= 6, j1= 8 )
CALL TRAY( j= 8, j_1= 7, j1= 9 )
CALL TRAY( j= 9, j_1= 8, j1=10 )
CALL TRAY( j=10, j_1= 9, j1=11 )
CALL TRAY( j=11, j_1=10, j1=12 )
CALL TRAY( j=12, j_1=11, j1=13 )
CALL TRAY( j=13, j_1=12, j1=14 )
CALL TRAY( j=14, j_1=13, j1=15 )
CALL TRAY( j=15, j_1=14, j1=16 )
CALL TRAY( j=16, j_1=15, j1=17 )
CALL TRAY( j=17, j_1=16, j1=18 )
CALL TRAY( j=18, j_1=17, j1=19 )
CALL TRAY( j=19, j_1=18, j1=20 )
CALL BOTM( n=N, n_1=N-1 )

   D = 0
   U1 = U(1);   U(2:N-1) = Uj;
   V = 1;       L = 1

/* 操作条件 */
   QR = 1e5
   U1 = 2;  Uj = 0.1
   P = ( 760, 765, 770, 775, 780,
         785, 790, 795, 800, 805,
         810, 815, 820, 825, 830,
         835, 840, 845, 850, 855 )
   x' = 0;  U'(N) = 0
   U(N) = 17.4

/* 仕込み液 */
VAR  xF(3)   "仕込み液モル分率          [-]"
   xF = ( 0.3, 0.2, 0.5 )
   SUM( U ) = F
   SUM( U*x(,1) ) = F*xF(1)
   SUM( U*x(,2) ) = F*xF(2)
   SUM( x(N) ) = 1

OUTPUT    x, y, T
```

図8　回分蒸留計算(全還流定常状態)のソーステキスト

ナミックシミュレーションが手軽にできるわけですが，最近はこの技術が応用されてプロセス制御コンピュータなどの実時間システムに組み込まれたり，トレーニングシミュレータの基盤技術として使われたりなど，その活躍の場をますます広げています．

参考文献

1) 宮原ほか：連載「EQUATRAN-M技術計算用連立方程式解法言語」，ケミカルエンジニヤリング，8月号(1985)～4月号(1986)
2) 宮原ほか：続連載「EQUATRAN-M技術計算用連立方程式解法言語」，ケミカルエンジニヤリング，10月号(1987)～2月号(1988)
3) 三井東圧EQM研究会：「パソコンのための方程式解法ソフト EQUATRAN-M入門」，省エネルギーセンター，(1987)
4) 佐渡友，宮原：「レビュー 回分蒸留」，分離技術，Vol.20，No.6，P.15 (1990)

＜問い合わせ先＞
三井東圧化学㈱システム部イコートラン係
〒100 東京都千代田区霞が関3-2-5(霞が関ビル)
☎ 03-3592-4190

使用記号

L	：液流量	[kg mol/h]
V	：蒸気流量	[kg mol/h]
U	：液ホールドアップ	[kg mol]
x	：液モル分率	[−]
y	：蒸気モル分率	[−]
T	：温度	[℃]
P	：圧力	[mmHg]
h	：液エンタルピ	[kcal/kg mol]
H	：蒸気エンタルピ	[kcal/kg mol]
Q_C	：コンデンサ除熱量	[*kcal/h*]
Q_R	：リボイラ加熱量	[kcal/h]
D	：留出量	[kg mol/h]
R	：還流比	[−]
t	：時間	[h]

添字

i　：成分
j　：段数