

フラッシュ蒸溜プロセスのシミュレーション・プログラム*

井上久仁子** 研野和人**

1. まえがき

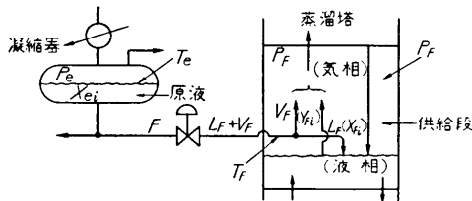
ある圧力の多成分よりなる原液を、より圧力の低い雰囲気フラッシュすると、圧力差があまり大きくないときは気相と液相にわかれるであろう。そして気相は原液よりも揮発しやすい成分をより多く含み、液相は揮発しにくい成分をより多く含んだものとなるので、多成分よりなる原液のある成分を分離することができる。これをフラッシュ蒸溜プロセスという。石油工業における蒸溜塔の原液供給段として、このフラッシュ蒸溜プロセスが含まれる。

本報告は、第1図に示すような一般的なフラッシュ蒸溜装置について取り扱っている。この報告のプロセスにおいては、次の二つの仮定をする。

(1) 気相の成分 i のモル分率 Y_i と液相のモル分率 X_i には、 $Y_i = K_i X_i$ の式が成立し、ここに K_i は圧力と温度のみの関数である。この仮定は石油工業において取り扱う成分については一般に成立する。

(2) 気相と液相の平衡関係は常に成立する。

フラッシュ・プロセスのシミュレーションの数式モデルは、C.D. Holland¹⁾の方法に従った。このプログラムを用いることにより、凝縮器(第1図参照)内の圧力と、フラッシュされる装置内の圧力および原液の組成があたえられた場合に、原液をフラッシュさせたフィードの温度、気相と液相の量および各相の各成分のモル分率を求めることができる。



第1図 フラッシュ蒸溜プロセス

2. 入力データと出力データ

第1表に入力に関する情報をまとめる。

第1表 入力データ

P_e :	原液のある凝縮器内の全圧
P_f :	フラッシュされる蒸溜塔の供給段の全圧
i :	原液の成分 (1~n)
F :	凝縮器より蒸溜塔に供給される原液の流量
X_{ei} :	凝縮器内原液の液相の成分 i のモル分率
$K_{ei}(T)$:	凝縮器内の圧力 P_e での成分 i の平衡係数
$K_{Fi}(T)$:	蒸溜塔供給段の圧力 P_f での成分 i の平衡係数
$h_i(T)$:	液相の成分 i の単位当たりのエンタルピー
$H_i(T)$:	圧力 P_f での気相の成分 i の単位当たりのエンタルピー
$[T_{U1}, T_{L1}]$:	K_{ei}, h_i は圧力 P_e における実験データから温度 T の多項式近似であたえられている。この近似式の成り立つ温度区間の下限 T_{L1} と上限 T_{U1} をいみする。
$[T_{U2}, T_{L2}]$:	K_{Fi}, h_i, H_i は圧力 P_f における実験データから温度 T の多項式近似であたえられている。この近似式の成り立つ区間の下限 T_{L2} と上限 T_{U2} をいみする。

第2表に求めるべき出力の情報をまとめる。

第2表 出力データ

T_e :	あたえられた組成 X_{ei} をもつ原液が圧力 P_e の凝縮器内で有する温度、ただし凝縮器は図示するように気相と液相とが存在し、かつ気液平衡が成立しているものとする。
H_e :	組成 X_{ei} をもつ原液が温度 T_e において有する単位量当たりのエンタルピー
Y_{ei} :	圧力 P_e の凝縮器内の液相と平衡関係にある気相の成分 i のモル分率
T_B :	組成 X_{ei} の液が圧力 P_f において有する沸点
T_D :	組成 X_{ei} の気体が圧力 P_f において有する露点。
T_F :	蒸溜塔供給段にフラッシュされたときの原料の温度。供給段では原料は気相と液相とに分かれ、その二相の間には、気液平衡が成立しているものとする。
X_{Fi} :	供給段における原料の液相の成分 i のモル分率
Y_{Fi} :	供給段における原料の気相の成分 i のモル分率
L_F :	供給段における原料の液相の流量
V_F :	供給段における原料の気相の流量
T_F :	V_F/F
l_{Fi} :	L_F 中の成分 i の流量
v_{Fi} :	V_F 中の成分 i の流量

3. 数式モデル

凝縮器内および蒸溜塔供給段においては気相と液相のモル分率の間には前述した仮定に従って平衡関係が

* The Simulation Program of the Flash Distillation Process, by Kuniko Inoue and Kazuto Togino

** 工業技術院 機械試験所

成立しているので、

$$Y_{ei} = K_{ei} X_{ei} \quad \text{および} \quad \sum_{i=1}^n Y_{ei} = 1 \quad (1.1)$$

$$Y_{Fi} = K_{Fi} X_{Fi} \quad \text{および} \quad \sum_{i=1}^n Y_{Fi} = 1 \quad (1.2)$$

の関係がある。

また物質収支

$$F X_{ei} = L_F X_{Fi} + V_F Y_{Fi} \quad (2)$$

の関係および、熱収支の関係

$$F H_e = L_F \sum_{i=1}^n h_i X_{Fi} + V_F \sum_{i=1}^n H_i Y_{Fi} \quad (3)$$

が成立しなければならない。この三つの式が本報告のシミュレーションにおける基本の数式モデルを導く。

次に第2図のフローチャートを参照して、シミュレーションの手順を説明する。

3.1 原液のエンタルピー

圧力 P_e の凝縮器内で、組成 X_{ei} の原液が平衡状

態にあるから、温度 T_e は (1) 式より導かれる次式

$$F_e(T) = \sum_{i=1}^n K_{ei} X_{ei} - 1 \quad (4)$$

を零にする値 T である。上式において K_{ei} は温度 T の近似多項式 $K_{ei} = \sum_{j=0}^m \alpha_{ij} T^j$ ($T_{L1} \leq T \leq T_{U1}$) であたえられているので、(4) 式は変数 T のみの関数となり、Newton の反復法を用いて T_e を算出する。反復過程において r 回目の近似値が $T^{(r)} < T_{L1}$ となるときは、

$$T^{(r)} = \frac{(T^{(r-1)} + T_{L1})}{2}$$

また r 回目の近似値が $T^{(r)} > T_{U1}$ であれば

$$T^{(r)} = \frac{(T^{(r-1)} + T_{U1})}{2}$$

なる近似値を r 回目の近似値とする *reguli-falsi* の方法を用いて、近似値が $T_{L1} \leq T^{(r)} \leq T_{U1}$ を満足するように補正する。

反復回数が所定以上に達したときは、根は得られなかったことをプリントして、次の入力データに対するシミュレーションに進む。

全圧 P_e における凝縮器内の温度 T_e がきまれば、エンタルピーの近似式 $h_i = \sum_{j=0}^m \beta_{ij} T^j$ ($T_{L1} \leq T \leq T_{U1}$) を用いて、これに $T = T_e$ を代入して蒸溜塔に供給される原液の単位量当たりのエンタルピー

$$H_e = \sum_{i=1}^n h_i X_{ei} \quad (5)$$

が算出される。

3.2 全圧 P_F における組成 X_{ei} の沸点と露点

圧力 P_F で組成 X_{ei} の液体の沸点 T_B は (1.1) 式と、 $X_{Fi} = X_{ei}$ の関係を用いてえられる次式

$$F_B(T) = \sum_{i=1}^n K_{Fi} X_{ei} - 1 \quad (6)$$

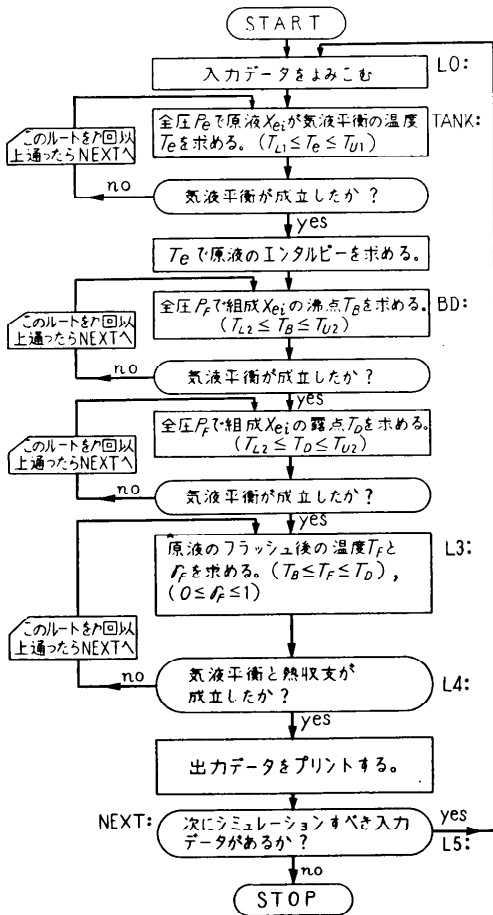
を零にする T の値として求められる。

一方圧力 P_F で組成 X_{ei} の気体の露点 T_D は (1.2) 式と、 $Y_{Fi} = X_{ei}$ の関係を用いてえられる次式

$$F_D(T) = \sum_{i=1}^n \frac{X_{ei}}{K_{Fi}} - 1 \quad (7)$$

を零にする T の値として求められる。ただし上式中 K_{Fi} は、 $K_{Fi} = \sum_{j=0}^m \alpha'_{ij} T^j$ ($T_{L2} \leq T \leq T_{U2}$) なる近似多項式であたえられる。

T_B および T_D は (6) 式および (7) 式にそれぞれ Newton の反復法を用いて計算されるが、その過程で r 回目の近似値が、 $T^{(r)} < T_{L2}$ または、 $T^{(r)} > T_{U2}$ になったときは、3.1 節で述べたように *reguli-falsi* を用いて $T_{L2} \leq T^{(r)} \leq T_{U2}$ を満足するように補正する。そして反復回数が所定以上に達したときは、根は



第2図 フローチャート

得られなかったことをプリントして次のデータに対するシミュレーションに進む。 T_B , T_D を求めた意味は次節において説明される。

3.3. T_F と γ_F

T_F と γ_F は次にのべるように物質収支と熱収支との関係を同時にみたしてはいけない。

i) 凝縮器より供給される原料と、蒸溜塔供給段にフラッシュされた供給段の原料との間には、各成分について物質収支(2)式が成立する。供給段に供給される気相 V_F の成分 i のモル分率 Y_{Fi} については、(2)式に $X_{Fi} = \frac{Y_{Fi}}{K_{Fi}}$ および $\frac{V_F}{F} = \gamma_F$, $\frac{L_F}{F} = 1 - \gamma_F$ の関係を代入して次式

$$Y_{Fi} = \frac{X_{ei} K_{Fi}}{(K_{Fi} - 1)\gamma_F + 1} \quad (8)$$

がえられるであろう。

蒸溜塔にフラッシュされたときの原料の温度 T_F と、 γ_F は(8)式および(1.2)式より導かれる次式

$$F_0(T, \gamma) = \sum_{i=1}^n \frac{X_{ei} K_{Fi}}{(K_{Fi} - 1)\gamma + 1} - 1 \quad (9)$$

を零にする T と γ の値でなくてはならない。同時に次の熱収支をみたす必要がある。

ii) フラッシュされた供給段における原料の気相のエンタルピーは、(2)式の両辺に H_i をかけた次式として求められる。

$$V_F \sum_{i=1}^n H_i Y_{Fi} = F \sum_{i=1}^n H_i X_{ei} - L_F \sum_{i=1}^n H_i X_{Fi} \quad (10)$$

上式において H_i は温度 T の近似多項式 $H_i = \sum_{j=0}^{m_i} \beta'_{ij} T^j$ ($T_{L2} \leq T \leq T_{U2}$) であたえられている。熱収支の関係式(3)式の右辺第2項に(10)式を代入すると、

$$FH_e = F \sum_{i=1}^n H_i X_{ei} - L_F \sum_{i=1}^n (H_i - h_i) X_{Fi} \quad (11)$$

がえられるであろう。液相における各成分の単位量あたりのエンタルピーは、ここでは圧力に関係がないと仮定して、3.1節で用いた $h_i = \sum_{j=0}^{m_i} \beta_{ij} T^j$ で与えられる。したがって h_i は ($T_{L1} \leq T \leq T_{U1}$) と ($T_{L2} \leq T \leq T_{U2}$) の両区間で成立することが必要である。(11)式に $\frac{V_F}{F} = \gamma_F$, $X_{Fi} = \frac{X_{ei}}{(K_{Fi} - 1)\gamma_F + 1}$ の関係を代入し、さらに(9)式を収束の許容誤差を同値にするため、その両辺を H_e で割れば

$$F_1(T, \gamma) = \left\{ \sum_{i=1}^n H_i X_{ei} - (1 - \gamma) \sum_{i=1}^n (H_i - h_i) \cdot \frac{X_{ei}}{(K_{Fi} - 1)\gamma + 1} \right\} \frac{1}{H_e} - 1 \quad (12)$$

を得るであろう。 T_F と γ_F は上式を零にする T と γ の値でなくてはならない。

上述した(9)式と(12)式を同時に零にする T と γ の値が T_F と γ_F である。これを求めるには Newton Raphson の反復法を用いる。上の2式より次の連立方程式がえられる。

$$\left. \begin{aligned} F_0 + \frac{\partial F_0}{\partial \gamma} \Delta \gamma + \frac{\partial F_0}{\partial T} \Delta T &= 0 \\ F_1 + \frac{\partial F_1}{\partial \gamma} \Delta \gamma + \frac{\partial F_1}{\partial T} \Delta T &= 0 \end{aligned} \right\} \quad (13)$$

ここに

$$\frac{\partial F_0}{\partial \gamma} = \sum_{i=1}^n \frac{X_{ei} K_{Fi} (1 - K_{Fi})}{(K_{Fi} - 1)\gamma + 1}^2$$

$$\frac{\partial F_0}{\partial T} = (1 - \gamma) \sum_{i=1}^n \frac{X_{ei} \left(\frac{dK_{Fi}}{dT} \right)}{\{(K_{Fi} - 1)\gamma + 1\}^2}$$

$$\frac{\partial F_1}{\partial \gamma} = \frac{1}{H_e} \sum_{i=1}^n \frac{\lambda_i X_{ei} K_{Fi}}{\{(K_{Fi} - 1)\gamma + 1\}^2}$$

ここで $\lambda_i = H_i - h_i$

$$\frac{\partial F_1}{\partial T} = \frac{1}{H_e} \left\{ \sum_{i=1}^n X_{ei} \frac{dH_i}{dT} - (1 - \gamma) \cdot \sum_{i=1}^n \frac{\left(\frac{d\lambda_i}{dT} \right) X_{ei}}{(K_{Fi} - 1)\gamma + 1} - \gamma \frac{\partial F_0}{\partial T} \right\}$$

である。

上式を解いて補正項 $\Delta \gamma$, ΔT をうることができるので、 r 回目の近似値は次式からえられる。

$$\left. \begin{aligned} T^{(r)} &= T^{(r-1)} + \Delta T^{(r-1)} \\ \gamma^{(r)} &= \gamma^{(r-1)} + \Delta \gamma^{(r-1)} \end{aligned} \right\}$$

ただし(9), (12)式はフラッシュされた供給段での原料は気相と液相とに分れ、この二相間では気液平衡関係が存在することを仮定して導かれた式である。したがって(9), (12)式は区間 $0 \leq \gamma_F \leq 1$, $T_B \leq T_F \leq T_D$ において成立する。

上述の理由から r 回目の近似値が、これらの範囲外になったときは、3.1節で述べたように reguli-falsi を用いて、 $T_B \leq T^{(r)} \leq T_D$, $0 \leq \gamma^{(r)} \leq 1$ を満足するように r 回目の近似値を補正する。

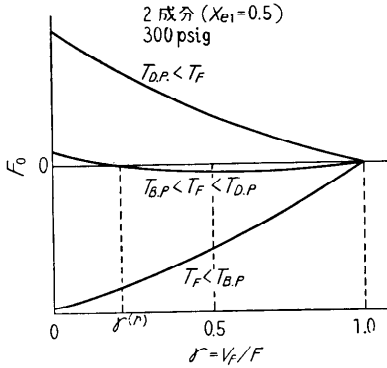
また Newton Raphson の反復回数が所定以上に達したときは、根はえられなかったことをプリントして、次の入力データに対するシミュレーションへ進む。(9), (12)式が、同時に許容誤差内に収束したならば、出力データをプリントし、さらに次の入力データに対するシミュレーションへ進む。

3.4 F_e , F_B , F_D , F_0 , F_1 の性質

気液平衡係数 K_i が、圧力 P をパラメータとして、温度 T の単調増加関数であるような成分を対象とす

る。フラッシュ・プロセスにおいては、上述した F_e , F_B , F_D , F_0 , F_1 は、ただ一つの根を有することが簡単に証明できる。一般に成分が炭水化物のときには、 K_i は単調増加関数となる。

次に F_0 はただ一組の根 (γ_F , T_F) しか有しないことを簡単にのべておく。



第3図 温度と γ の関係

$0 \leq \gamma \leq 1$ で、 $T_B \leq T \leq T_D$ の区間内では、

$$[F_0(T, \gamma)]_{\gamma=0} > 0$$

$$[F_0(T, \gamma)]_{\gamma=1} = 0$$

$$[F_0'(T, \gamma)]_{\gamma=1} > 0$$

$$[F_0''(T, \gamma)] > 0$$

また $T_B > T$ なる T に対しては、

$$[F_0(T, \gamma)]_{\gamma=0} < 0$$

$$[F_0(T, \gamma)]_{\gamma=1} = 0$$

$$[F_0'(T, \gamma)]_{\gamma=1} > 0$$

$$[F_0''(T, \gamma)] > 0$$

また $T_D < T$ なる T に対しては、

$$[F_0(T, \gamma)]_{\gamma=0} > 0$$

$$[F_0(T, \gamma)]_{\gamma=1} = 0$$

$$[F_0'(T, \gamma)] < 0$$

$$[F_0''(T, \gamma)] > 0$$

となり第3図に示したような関係である。上式において

$$F_0' = \frac{\partial F_0}{\partial T}, \quad F_0'' = \frac{\partial^2 F_0}{\partial T^2}$$

である。

4. プログラム上の注意

(1) ISSP ALGOL²⁾ を用いて FACOM 202 で実際にシミュレーションを行なったため、若干の標準手続きについて説明する。

i) : 入力 of procedure の一例としては、

```
READREAL 6 (6, F, E, L1, U1, L2,
U2); は
READREAL(F); READREAL(E); REA-
DREAL(L1); READREAL(U1); REA-
DREAL(L2); READREAL(U2); のこと
である。
```

ii) : 出力 of procedure の一例としては、

```
printreal(5, F); は
仮数部5桁で印刷せよということである。
```

iii) : sense(2, L5); は、

制御卓のスイッチの No. 2 が jump になっているときは、この statement は go to L5 と同値になり、ignore のときは dummy statement となる。

(2) スwitch の No. 2 を ignore にすると、 K_{ei} , K_{Fi} , h_i , H_i を印刷する。

(3) データテープの作り方 (パンチの順序にのべる)。

integer: 成分数, 平衡係数の近似式の次数, エンタルピーの近似式の次数, Newton の反復法の反復回数の上限, reguli-falsi の反復回数の上限, 次の入力データをよみこむか, 計算を停止するか判定 (0以外は停止). 以上6個.
real: 原料の単位時間当たりの流量, 許容誤差, L1, U1, L2, U2. 以上6個.

array: K_{ei} , K_{Fi} , h_i , H_i の係数 (各成分ごとに高次の係数から). 圧力 P_e における原液の組成.

(4) シミュレーション完了によって得られた L_F , V_F およびエンタルピーは、フラッシュプロセスが付け加されている蒸溜塔の静特性のシミュレーション・プログラム²⁾ の入力データとなる。

(5) Newton の反復法の許容誤差および Newton Raphson の許容誤差については、対象とする多成分系により異なる。筆者等が試みたエチレン、エタン2成分系においては、(4), (6), (7) および (9), (12) 式ともに 10^{-6} の誤差で、イテレーション回数2回から4回で収束している。

5. プログラム

Flash Simulation :

```
begin integer i, n, m, I, N, M3, Mr, R1, R2,
R3, C;
```

```

L 0: READINTEGER 6 (6, i, n, m, M3, Mr, C);
begin
  real F, E, L1, L2, U1, U2, He, T, G, W1,
    W2, W3, W4, W5, W6, W7, G0, T0,
    TB, TD, dT;
  array Ae, A[1:i, 0:n], B1, B2[1:i, 0:m],
    K, dK, h, dh, H, dH, Xe, X, Y, W[1:
    i];
  procedure Polyno (a, j, b, P, dP);
    integer j; Boolean b;
    array a, P, dP;
    begin
      for I:=1 step 1 until i do
        begin W1:=W2:=a[I, j];
          for N:=j-1 step-1 until 0 do
            begin W1:=W1×T+a[I, N];
              if b∧N≠0 then W2:=W2×T+W1
                end;
              P[I]:=W1; if b then dP[I]:=W2
            end
          end Polyno;
    real procedure Sum (P, I);
      integer I; real P;
      begin real S;
        S:=0;
        for I:=1 step 1 until i do S:=S+P;
        Sum:=S
      end Sum;
    procedure Test (1, u, x, x0);
      real 1, u, x, x0;
      begin
        if 1>x then
          begin R1:=R1+1; if R1>Mr then
              go to NEXT;
              X:=(1+x0)/2; go to L10
            end;
          if u<x then
            begin R2:=R2+1; if R2>Mr then go to
                NEXT;
                X:=(u+x0)/2
              end;
            L10: end Test;
    procedure Root (1, u, a, b2);
      real 1, u; array a; Boolean b2;
      begin real F0;
        T:=(1+u)/2; R1:=R2:=R3:=0;
      L11: R3:=R3+1; if R3>M3 then go to
          NEXT;
          Polyno (a, n, true, K, dK);
          F0:=(if b2 then Sum (Xe[I]×K[I], I)
            else Sum(Xe[I]/K[I], I))-1;
          if abs(F0)≥E then
            begin W1:=(if b2 then Sum(Xe[I]×dK
              [I], I) else Sum(-Xe[I]×dK
                [I]/(K[I]×K[I]), I));
              if abs (W1)<10-10 then
                begin crlf; printstring('F'=0'); go to
                    NEXT
                  end;
                T0:=T; T:=T-F0/W1; Test (1, u,
                  T, T0); sense(2, L12); crlf; printstring
                    ('T='); printreal (3, T);
            L12: go to L11
          end
        end Root;
      procedure printarray (X, ST, N);
        integer N; string ST; array X;
        begin integer I, K;
          crlf; printstring(ST);
          for I:=1 step 1 until N do
            begin crlf; for K:=0 step 1 until N do
                printreal (10, X[I, K])
              end
            end printarray;
      procedure printv (ST1, X, ST2, Y, ST3, Z);
        integer Z; string ST1, ST2, ST3;
        real X, Y;
        begin crlf; printstring (ST1); printreal (5,
          X); printstring (ST2); printreal (5, Y);
          printstring (ST3); printinteger (2, Z)
        end printv;
      crlf; printstring ('Flash calculation');
      READREAL 6 (6, F, E, L1, U1, L2, U2);
      READARRAY 5 (5, Ae, A, B1, B2, Xe); crlf
      printstring ('Feed='); printreal (5, F);
      printstring ('error'); printreal (2E); crlf;
      printstring ('Xe');
      for I:=1 step 1 until i do

```

```

begin crlf; printinteger (2, I); printreal (3,
Xe[I]) end; crlf printstring ('L1, U1, L2,
U2'); crlf; for W1:=L1, U1, L2, U2 do
printreal (3, W1); crlf; sense (2, TANK);
printarray (Ae, 'Ae', n); printarray (A, 'A,'
n); printarray (B1, 'B1', m); printarray
(B2, 'B2', m);
TANK: Root (L1, U1, Ae, true); Polyno
(B1, m, false, h, dh);
He:=Sum(Xe[I]×h[I], I); printv ('T of
Tank=', T, 'Enthalpy/unit', He, 'ite-
ration no=', R3); sense (2, BD);
crlf; printstring ('K');
for I:=1 step 1 until i do
begin crlf; printinteger (2, I); printreal
(10, K[I])
end The Enthalpy of Feed;
BD: crlf; printstring('Tb.p. & Td.p. calcu-
lation');
Root (L1, U1, A, true); TB:=T;
Root (L2, U2, A, false); TD:=T;
printv ('Tb=', TB, 'Td=', TD, '', R3);
T:=(TB+TD)/2; R1:=R2:=R3:=0;
G:=0;
L3: R3:=R3+1; if R3>M3 then go to
NEXT;
Polyno (A, n, true, K, dK); Polyno (B1, m,
true, h, dh); Polyno (B2, m, true, H, dH);
W1:=W2:=0;
for I:=1 step 1 until i do
begin W[I]:=(K[I]-1)×G+1;
X[I]:=Xe[I]/W[I];
Y[I]:=X[I]×K[I];
W1:=W1+Y[I];
W2:=W2+H[I]×Xe[I]-(1-G)
×(H[I]-h[I])×Xe[I]/W[I]
end;
W1:=W1-1; W2:=W2/He-1;
if abs(W1)≥E∧abs(W2)≥E then
begin W3:=W4:=W5:=W6:=W7:=0;
for I:=1 step 1 until i do
begin
W3:=W3+Y[I]×(1-K[I])/W[I];
W4:=W4+X[I]×dK[I]/W[I];
W5:=W5+(H[I]-h[I]×Y[I]/W[I];
W6:=W6+Xe[I]×dH[I];
W7:=W7+(dH[I]-dh[I])
×Xe[I]/W[I]
end;
W4:=(1-G)×W4; W5:=W5/He;
W6:=(W6-(1-G)×W7-G
×W4)/He;

```

```

W7:=W3×W6-W4×W5;
if abs (W7)<10-10∧ abs (W5)<10
-10 then go to NEXT; T0:=T;
dT:=(W1×W5-W2×W3)/W7;
T:=T+dT;
Test(TB, TD, T, T0); G0:=G;
G:=G-(W1+W4×dT)/W3;
Test(0.0, 1.0, G, G0); sense(3, L4);
printv('T=', T, 'G=', G, 'iteration
no', R3);
printv('F0=', W1, 'F1=', W2, '',
R3);
L4: go to L3
end Simulation;
printv('T=', T, 'G=', G, 'iteration
no', R3);
W1:=F×G, W2:=F×(1-G);
printv('L=', W2, 'V=', W1, '', R3);
crlf;
printstring('Fxi Li Vi Xi Yi');
for I:=1 step 1 until i do
begin
W3:=X[I]; W4:=Y[I];
W5:=W1×W4; W6:=W2×W3;
W7:=F×Xe[I];
printinteger (2, I);
for W1:=W7, W6, W5, W4, W3 do
printreal (5, W1)
end output data;
sense (2, L5); crlf; printstring ('K h
H');
for I:=1 step 1 until i do
begin crlf; printinteger (2, I);
for W1:=K[I], h[I], H[I] do
printreal (5, W1);
end;
go to L5;
NEXT: crlf; printstring ('out of the
interval');
L5: if C=0 then go to L0
end Print
end Flash Simulation

```

参考文献

- 1) C.D. Holland: Multicomponent distillation. Prentice-Hall, 1963 pp. 14~40
- 2) 研野和人他: 蒸溜塔の静特性の計数型電子計算機によるシミュレーション, 機械試験所所報, 17巻3号, 1963, pp. 63~70
- 3) 井上謙蔵他: ISSP ALGOL のコンパイラ, 情報処理, 第5巻, 第1号, 1963
(昭和39年4月14日受付)