

方程式解法ソフトによるダイナミックシミュレーション

今回と次回の2回にわたって、微分方程式を含んだ問題を取り上げる。化学工業において微分方程式によるモデル化を必要とする問題は数多いが、中でも装置やプラントの時間的な挙動を追跡する、いわゆるダイナミックシミュレーションはその代表格と言えよう。今回はその一つの例として、スチレンの重合反応器のシミュレーションをFORTRANのプログラムおよび方程式解法ソフトによって行った例を紹介する。

ダイナミックシミュレータ

具体的な問題を取り扱う前に、化学プロセスの動的な挙動を検討する場合の、一般的なアプローチについて述べておきたい。化学プロセスでは一般にその数式モデルは複雑で、かつ線形近似による扱いが困難であるため、常微分方程式を含む物理的な数式モデルを直接数値的に解く方法が採られる。シミュレーションの方法としては、問題ごとに直接プログラムを作成する方法を別とすれば、2つのアプローチがあると言える。

一つは化学プロセス用の汎用シミュレータを用いるもので、化学プロセスによく現われる各種の装置や制御機器のモデルおよび物性計算ルーチンがモジュールとして用意されており、これらのモジュールを組み合わせることによって、任意のプロセスを合成することができる。

もう一方のアプローチは方程式解法ソフトあるいは連続系シミュレーション言語を用いるもので、こちらの方は数式モデルをそのまま式の形で組み込むことにより、モデルを構成する。

表1に代表的なダイナミックシミュレータを、前者のモジュール型と後者の方程式型に分類して示してある。モジュール型は、厳密なモデル化が必要である大規模な

表1 ダイナミックシミュレータ¹⁾²⁾

分類	シミュレータ名	開発年	開発元
モジュール型	DYNSYS	1969	McMaster 大学
	DYSCO	1974	Michigan 大学
	DOPL	1975	千代田化工建設(株)
	DPS V2	1983	(株)日本科学技術研修所
	MODYS	1977	三井東圧化学(株)
方程式型	CSMP III	1972	IBM 社
	CSSL IV	1976	Nilsen Associates 社
	ACSL	1975	Mitchell and Gauthier Associates 社
	SLCS IV EQUATRAN-M (バージョン2)	1979 1986	富士通(株) 三井東圧化学(株)

問題でも比較的容易に取り扱える反面、フレキシビリティに乏しいきらいがあり、しばしば新しいモジュールを作成する必要を生じる。一方、方程式型はモデルを自由に表現できる反面、大規模な問題を扱うことは不可能ではないまでも、かなり困難であると言えよう。目的に応じて両者を使い分けたり、あるいは併用しているのが実情である。本稿では方程式型のシミュレーション言語SLCS IVと、本連載の中心である方程式解法ソフトEQUATRAN-Mについて詳しく述べよう。

連続重合反応のシミュレーション

モノマー（単量体）の重合によって高分子を生成する反応は、ポリエチレン、ポリプロピレン、ポリスチレン、ポリ塩化ビニルなど、おなじみのプラスチックの製造に

図1 スチレンの重合反応器

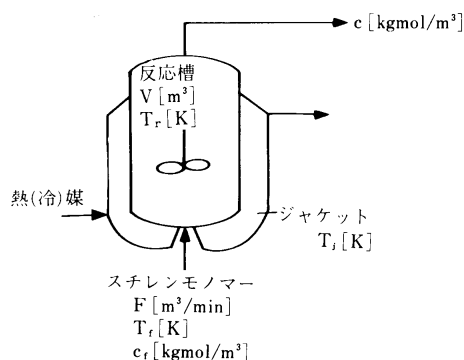


表2 重合反応シミュレーションの計算条件 (起動時)

反応槽容積 (V)	1 m ³
ジャケット伝熱面積 (A)	2 m ²
総括伝熱係数 (U)	2 kcal/m ² min °C
供給量 (F)	0.001 m ³ /min
供給モノマー濃度 (c _f)	7 kgmol/m ³
供給温度 (t _f)	20 °C
ジャケット温度 (t _j)	140 °C

欠かすことができない。これらの重合反応は、一般に大きな発熱を伴い、また温度の上昇によって反応速度が増大するので、適切な温度制御が行われないと温度が急上昇して、いわゆる暴走反応を起こしやすい。ここでは重合反応の例として、スチレンの重合反応を取り上げ、その特性を明らかにしてみよう。^{3) 4)}

反応器は図1に示すように、ジャケットによって冷却された流通型の反応槽を有し、槽内は完全混合とみなしてよい。スチレンモノマーの重合反応は、一次反応と考えられ、その反応速度 r [kg mol/m³ · min] は次式で与えられる。

$$r = k \cdot c \quad (1)$$

c は槽内のスチレンモノマー濃度 [kg mol/m³] であり、反応速度定数 k [1/min] は、次のアレニウス式によって与えられている。³⁾

$$k = k_0 e^{-E/RT_r} \quad (2)$$

ここで、 k_0 (頻度因子) = 1.39×10^9 1/min, E (活性化エネルギー) = 2.13×10^4 kcal/kg mol, R (ガス定数) = 1.987 kcal/kg mol · K であり、 T_r は反応槽の温度 [K] である。

つぎに反応器におけるモノマーの物質収支から、 t を時間 [min] として、

$$V \frac{dc}{dt} = F(c_f - c) - V_r \quad (3)$$

が、さらに熱収支から、

$$V\rho c_p \frac{dT_r}{dt} = F\rho \int_{T_r}^{T_f} c_p dT + Q_R - Q_j \quad (4)$$

が得られる。ここで V は反応槽容積 [m³]、 F は供給量 [m³/min]、 c_f は供給モノマー濃度 [kg mol/m³]、 T_f は供給温度 [K]、 ρ は反応液密度 (= 900 kg/m³)、 c_p は反応液の比熱 [kcal/kg · K] である。反応による発熱量 Q_R [kcal/min] とジャケットによる冷却量 Q_j [kcal/min] は、重合反応の反応熱を $-ΔH$ [kcal/kg mol]、またジャケットの冷却面積を A [m²]、総括伝熱係数を U [kcal/m² min °C]、ジャケットの温度を T_j [K] として、それぞれ、

$$Q_R = -ΔH \cdot r \cdot V \quad (5)$$

$$Q_j = U \cdot A (T_r - T_j) \quad (6)$$

で与えられる。ここに $-ΔH = 17600$ kcal/kg mol であり、比熱 c_p は温度 T [K] の関数として次式で与えられている。³⁾

$$\left. \begin{aligned} c_p &= a + b \cdot T \\ a &= 0.088, \quad b = 0.103 \times 10^{-2} \end{aligned} \right\} \quad (7)$$

シミュレーションは起動時を想定し、槽内に供給液を満たしたのち、少量のモノマーを流通させた状態で、ジャケットに一定温度の熱媒を供給して昇温し、反応開始の状態を観察するものとする。すなわち、微分方程式の初期条件として次式を与える。

$$\left. \begin{aligned} T_r(t=0) &= T_f \\ c(t=0) &= c_f \end{aligned} \right\} \quad (8)$$

図2 可変きざみ型 Runge-Kutta 公式⁵⁾

$$\begin{cases} k_1 = hf(x_0, y_0) \\ k_2 = hf(x_0 + h/2, y_0 + k_1/2) \\ k_3 = hf(x_0 + h/2, y_0 + k_2/2) \\ k_4 = hf(x_0 + h, y_0 - k_1 + 2k_2) \\ k_5 = hf(x_0 + h, y_0 + k_3) \\ y_1 = y_0 + (1/6)(k_1 + 4k_2 + k_4) \\ y_2 = y_0 + (1/6)(k_1 + 2k_2 + 2k_3 + k_5) \\ T = |y_1 - y_2| \end{cases}$$

注) $\frac{dy}{dx} = f(x, y)$ の積分において T を y_2 の打切り誤差の推定値として用いる。h は積分きざみ。

その他の計算条件は表2に示した。なお、シミュレーションの結果は反応槽の温度の他に、反応の進行度を示す反応率 x [-] を次式によって計算して表示する。

$$x = (c_f - c) / c_f \quad (9)$$

また、入力および表示する温度の単位は [°C] とする。

可変きざみ Runge-Kutta 公式によるプログラム

最初に、本例題を Runge-Kutta 公式による FORTRAN プログラムで解いてみよう。実際に計算してみれば分かるが、この例では反応槽の温度が上昇するに従い、反応速度が急速に増大し、ある時点で暴走反応が生ずる。この際、系の時定数が大幅に変化するので、積分きざみのとり方が難しい。きざみを十分小さくとらないと暴走反応前後が正しく計算できない一方、小さいきざみでは低温域でむだな計算に多くの時間を費すことになってしまう。このような場合、積分打切り誤差の評価を行いながら、きざみを調整することのできる可変きざみ型の公式がたいへん有効である。

可変きざみ型の積分公式は多く提案されているが、ここでは、4次の古典的 Runge-Kutta 公式に3次の Kutta 公式を組み合わせて、打切り誤差の推定を行う公式(図2)を使用する。⁵⁾ この公式では、5回の関数値の計算で4次公式の打切り誤差の推定が行える。

リスト1に本公式のためのサブルーチン RKV のリストを示す。このサブルーチンを用いれば、例題のための

リスト1 可変きざみ Runge-Kutta 法の FORTRAN サブルーチン "RKV"

```

SUBROUTINE RKV( NV, INDEX, X, STEP, Y, F, W, ERROR, AERROR )
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION Y(1),F(1),W(1)
C
C 4TH ORDER RUNGE-KUTTA METHOD WITH ERROR EVALUATION
C BY G. OGUCHI(1986)
C
C NV ... NUMBER OF DEPENDENT VARIABLES [INPUT]
C INDEX ... CONTROL [INPUT/OUTPUT]
C SET TO 0 ON INITIAL CALL ONLY
C RETURNS STAGE OF R. K. METHOD
C
C X ... INDEPENDENT VARIABLE [INPUT(INITIAL)/OUTPUT]
C STEP ... STEP OF INTEGRATION [INPUT(INITIAL)/OUTPUT]
C Y ... DEPENDENT VARIABLES [INPUT(INITIAL)/OUTPUT]
C F ... DERIVATIVES [INPUT]
C W ... WORK AREA [OUTPUT]
C
C RESERVE (NV+1)*4 DOUBLE WORDS
C
C ERROR ... RELATIVE ERROR ALLOEANCE [INPUT]
C AERROR ... ABSOLUTE ERROR ALLOEANCE [INPUT]
C OK WHEN ABS(Y-Y') < ABS(Y)*RERROR+AERROR FOR ALL Y
C
C
C IW1 = 4
C IW2 = IW1 + NV
C IW3 = IW2 + NV
C IW4 = IW3 + NV
C IF( INDEX.EQ.0 ) THEN
C W(2) = STEP
C W(3) = X
C INDEX = 1
C ENDIF
C
C GOTO ( 100, 200, 300, 400, 500 ), INDEX
100 CONTINUE
C H = STEP*0.500
C W(1) = X
C IF( X.GE. W(3)-STEP*1.0D-3 ) W(3) = W(3) + W(2)
C DO 110 I=1,NV
C W(IW1+I) = Y(I)
C W(IW2+I) = F(I)
C Y(I) = Y(I) + H*F(I)
C W(IW3+I) = F(I)
C W(IW4+I) = -F(I)
110 CONTINUE
C X = X + H
C GOTO 600
200 CONTINUE
C H = STEP*0.500
C DO 210 I=1,NV
C W(IW2+I) = W(IW2+I) + 2.0D0*F(I)
C Y(I) = W(IW1+I) + H*F(I)
C W(IW3+I) = W(IW3+I) + 4.0D0*F(I)
C W(IW4+I) = W(IW4+I) + 2.0D0*F(I)
210 CONTINUE
C GOTO 600
300 CONTINUE
C DO 310 I=1,NV
C W(IW2+I) = W(IW2+I) + 2.0D0*F(I)
C Y(I) = W(IW1+I) + STEP*F(I)
310 CONTINUE
C X = W(1) + STEP
C GOTO 600
400 CONTINUE
C DO 410 I=1,NV
C W(IW2+I) = W(IW2+I) + F(I)
C Y(I) = W(IW1+I) + STEP*W(IW4+I)
410 CONTINUE
C GOTO 600
500 CONTINUE
C H = STEP/6.000
C ERRMAX = 0.000
C DO 510 I=1,NV
C Y(I) = W(IW1+I) + H*W(IW2+I)
C YS = W(IW1+I) + H*(W(IW3+I) + F(I))
C EPS = DABS(Y(I))-RERROR + AERROR
C ERROR = DABS( YS - Y(I) )/EPS
C ERRMAX = DMAX1( ERROR, ERRMAX )
510 CONTINUE
C IF( ERRMAX.GT.1.0D0 ) THEN
C X = W(1)
C DO 520 I=1,NV
C Y(I) = W(IW1+I)
520 CONTINUE
C STEP = STEP*0.500
C ENDIF
C IF( ERRMAX.LT.0.03D0 .AND. STEP.LT.W(2) ) THEN
C STEP = STEP*2.000
C ENDIF
C IF( X.LT.W(3)-W(2)*1.0D-6 .AND. X*STEP*1.01D0.GE.W(3) )
C STEP = W(3) - X
600 INDEX = INDEX + 1
C IF( INDEX.EQ.0 ) INDEX = 1
C RETURN
C END
    
```

メインプログラムは図3のように作ればよい。RKVの使用方法は、このメインプログラムとリスト内の注釈によって明らかと思う。積分きざみは最初に与えた値(例では1min)から変化していくが、最初の値の倍数にあたる時点は必ず計算されるようになっている。

図3の計算結果で分かるように、経過時間が160minぐらいまでは比較的安定して温度が上昇しているが、その後急激に温度が327°Cまで上昇し、暴走反応に到って

図3 FORTRANによる連続重合反応のシミュレーション
(メインプログラムと計算結果)

```

Microsoft FORTRAN 使用

C MAIN FOR STYRENE POLYMERIZATION PROCESS SIMULATION
  IMPLICIT REAL*(A-H,O-Z)
  DIMENSION DY(2), Y(2), W(12)
  DATA V /1.000 /, A /2.000 /, U /2.000 /
  1 , RHO/900.DO /, DH /-17.603 /, AKO/1.39D9 /
  2 , E /2.13D4 /, R /1.987D0 /
  3 , SA, SB / 0.088D0, 0.103D-2 /
  DATA RERROR, AERROR/ 1.0D-4, 1.0D-4 /
C GET INPUT DATA
  WRITE(=,101)
  101 FORMAT(' F,CF,TFC,TJC = ')
  READ(=,*) F,CF,TFC,TJC
  WRITE(=,600) F,CF,TFC,TJC
  TJ = TJC * 273.15D0
  TF = TFC * 273.15D0
  INDEX = 0.D0
  TIME = 0.D0
  Y(1) = CF
  Y(2) = TF
  STEP = 1.0D0
  PTIME = 0.D0
  ETIME = 300.D0
C
DO 80 LOOP=1,9999
  AK = AKO*EXP(-E/R/Y(2))
  SR = AK*Y(1)
  DY(1) = F/V*(CF-Y(1))-SR
  QJ = U*A*(Y(2)-TJ)
  QR = -DH*V*SR
  CP = SA-SB*Y(2)
  CPDT = SA*(TF-Y(2))-SB/2.0D0*(TF**2-Y(2)**2)
  DY(2) = (F*CPDT+(QR-QJ)/RHO)/(CF*V)
C
  IF (INDEX.LE.1 .AND. TIME.GE.PTIME) THEN
    PTIME = PTIME + 10.0D0
    X = (CF-Y(1))/CF
    TE = Y(2)-273.15D0
    WRITE(=,801) TIME, X, TE, Y(1)
  ENDIF
  IF (INDEX.EQ.1 .AND. TIME.GT.ETIME) GOTO 81
C
  CALL RKV( 2, INDEX, TIME, STEP, Y, DY, V, RERROR, AERROR )
C
80 CONTINUE
81 STOP
600 FORMAT(/// ' INPUT DATA' /15X ' F =',F10.3,5X ' CF =',F10.3
1 /15X ' TFC =',F10.3,5X ' TJC =',F10.3/
601 FORMAT(' TIME=',F10.3, ' X=',F10.6, ' T=',F10.3
1 ' C=',F10.4)
  END

INPUT DATA
      F = .001      CF = 7.000
      TFC = 20.000  TJC = 140.000

TIME= .000 X= .0000000 T= 20.000 C= 7.0000
TIME= 10.000 X= .0000004 T= 32.661 C= 7.0000
TIME= 20.000 X= .0000021 T= 43.557 C= 6.9999
TIME= 30.000 X= .0000068 T= 53.013 C= 6.9995
TIME= 40.000 X= .000193 T= 61.285 C= 6.9987
TIME= 50.000 X= .000422 T= 68.575 C= 6.9970
TIME= 60.000 X= .000888 T= 75.057 C= 6.9939
TIME= 70.000 X= .001631 T= 80.878 C= 6.9886
TIME= 80.000 X= .002851 T= 86.169 C= 6.9800
TIME= 90.000 X= .004703 T= 91.053 C= 6.9671
TIME= 100.000 X= .007401 T= 95.842 C= 6.9482
TIME= 110.000 X= .011224 T= 100.054 C= 6.9214
TIME= 120.000 X= .016541 T= 104.414 C= 6.8842
TIME= 130.000 X= .023885 T= 108.879 C= 6.8328
TIME= 140.000 X= .034084 T= 113.663 C= 6.7614
TIME= 150.000 X= .048589 T= 119.122 C= 6.6599
TIME= 160.000 X= .070380 T= 125.969 C= 6.5073
TIME= 170.000 X= .107496 T= 136.127 C= 6.2475
TIME= 180.000 X= .202193 T= 159.788 C= 5.5846
TIME= 190.000 X= .999954 T= 326.700 C= .0003
TIME= 200.000 X= .999941 T= 314.834 C= .0004
TIME= 210.000 X= .999915 T= 303.022 C= .0006
TIME= 220.000 X= .999879 T= 293.048 C= .0008
TIME= 230.000 X= .999832 T= 283.091 C= .0012
TIME= 240.000 X= .999767 T= 273.731 C= .0016
TIME= 250.000 X= .999680 T= 264.946 C= .0022
TIME= 260.000 X= .999566 T= 256.713 C= .0030
TIME= 270.000 X= .999418 T= 249.007 C= .0041
TIME= 280.000 X= .999227 T= 241.803 C= .0054
TIME= 290.000 X= .998988 T= 235.078 C= .0071
TIME= 300.000 X= .998692 T= 228.805 C= .0092

```

る。図6はその結果をグラフ表示したものであるが、これをみるとその状況がよくわかる。なお、このように暴走してしまうと、反応液の性状が大幅に変化してしまうので、その後のシミュレーション結果は実際の事象を正しく表現していない。実際のプロセスでは、このように暴走する前にジャケット温度を下げる、モノマーのフィード量を増加させるなどの操作を行い、反応温度を制御して安定した運転状態に導いている。

化学工業における数値計算

SLCS IV⁶⁾は最初にも述べたように、方程式型の連続系シミュレーション言語であり、汎用大型コンピュータのバッチ制御下での一括処理およびTSS制御下での会話処理が可能となっている。この種の言語としてはCSMPが有名であるが、SLCS IVは習得しやすい演算子表現、多重積分に代表される1つ以上の独立変数の取り扱いなどの改良がなされており、機械、電気、化学、制御など広い分野で使用されている。

SLCS IVによる解法

さて、本例題をSLCS IVで解いてみよう。簡単なため、ここではバッチ制御下での一括処理による方法を示す。図4にSLCS IVで記述したソーステキストのリストと、表形式出力による計算結果の一部を示す。このリスト中で第1カラムに*が書いてある行は注釈である。5~25行目は構造文といい、モデルの方程式を記述する。書き方はFORTRANの算術代入文と同様であるが、微分項は13行、14行のように演算子表現ができる。5行目のINITIAL文、10行目のDYNAMIC文、25行目のTERMINAL文は翻訳制御文であり、それぞれイニシャルデビュジョン、ダイナミックデビュジョン、ターミナルデビュジョンの開始を示している。

イニシャルデビュジョンはシミュレーションの開始時刻に一度だけ実行され、初期条件を求める場合に使われる。ダイナミックデビュジョンは、シミュレーションの開始時刻から終了時刻まで積分間隔ごとに実行される。ターミナルデビュジョンは、シミュレーションの終了時刻に一度だけ実行される。イニシャルおよびダイナミックデビュジョンでは指定しない限り、計算順序が正しくなるように自動的に順序づけられる(ソートセクションと呼ばれる)。

27~32行目はデータ文であり、モデルで使用する定数(CONST文)、パラメータ(PARAM文)、初期条件などに値を設定するために使用される。

34~35行目は実行制御文であり、シミュレーションの時間に関する値の設定(TIMER文)、積分手法の指定(METHOD文)、積分計算の許容誤差の設定などを与える。

37~41 行目は出力制御文であり、シミュレーションの計算結果を種々な形式で出力する指定を行う。表形式出力 (TITLE 文, PRINT 文), グラフ形式出力 (LABEL 文, OUTPUT 文), ディスプレイ出力, ファイル出力などが指定できる。

43 行目の END 文は実行制御文の一つであり, ケーススタディのための区切を示す。この後に他のケーススタディのデータを書くことにより, 一回のジョブで何回ものケーススタディを実行できる。44 行目の ;; はソーステキストの最後を示すもので, 必ず書かなければならない。

SLCS IV の特徴

SLCS IV は連続系シミュレーション言語として必要な機能をほぼすべて満足しているが, 特に独立変数を 1 つ以上取り扱える機能は強力である。この機能により, Volterra 問題として有名な 2 種生物系の補食問題のような積分方程式を含む微分方程式のシミュレーションも容易に解くことができる。また, SLCS IV は FORTRAN を基本にしているのので, FORTRAN プログラムを容易に組み込むことができ, 多様な処理に対応できるようになっている。

このように SLCS IV は優れた言語ではあるが, 現実のシミュレーションを行う上でもの足りない部分もある。ひとつは初期値計算に関するもので, 他のひとつはダイナミックシミュレーションにおける条件変更に関するものである。初期値計算の問題は, たとえば微分項 = 0 としたときに成立する代数方程式の解法の問題であり, SLCS IV ではサポートされていないので, ユーザが他の方法でこの初期値計算を行っておかなければならない。条件変更に関しては時間的な条件での変更は容易に記述できるが, 変数の組み合わせによる複雑な条件変更の記述は難しいことがある。化学工業におけるダイナミックシミュレーションを行う上で, この 2 つの問題は重要であり, 次項で説明する EQUATRAN-M がその解決法を与えてくれている。

EQUATRAN-M による解

次に方程式解法ソフト EQUATRAN-M を使ってこの例題を解いてみよう。問題の記述方法は前回までに紹介

図 4 スチレンの重合反応器のシミュレーション (SLCS IV のソースリストと計算結果の一部)

```

***** PS REACTOR SIMULATION ***** ( 1)
***** ( 2)
***** ( 3)
***** ( 4)
***** ( 5)
INITIAL ( 6)
TF = TFC + 273.15 ( 7)
TR = TF ( 8)
C = CF ( 9)
***** (10)
DYNAM (11)
RR = K*C (12)
K = KO*EXP(-E/R/TR) (13)
C* = (F*(CF-C) - V*RR)/V (14)
TR* = (F*RHO*ICPDT + QR - QJ)/(V*RHO*CP) (15)
QJ = U*A*(TR-TJ) (16)
QR = -DH*V*RR (17)
CP = AA + BB*TR (18)
ICPDT = AA*(TF-TR) + BB/2*(TF**2 - TR**2) (19)
***** (20)
X = (CF-C)/CF (21)
TJ = TJC + 273.15 (22)
TRC = TR - 273.15 (23)
TF = TFC + 273.15 (24)
***** (25)
TERMINAL (26)
***** (27)
CONST V = 1.0 , A = 2.0 , U = 2.0 (28)
CONST RHO = 900.0 , DH = -17600.0 (29)
CONST KO = 1.39E9 , E = 2.13E4 , R = 1.987 (30)
CONST AA = 0.088 , BB = 0.103E-2 (31)
***** (32)
PARAM F = 0.001 , CF = 7.0 , TFC = 20.0 , TJC = 140.0 (33)
***** (34)
TIMER FINTIM= 300.0 , DELT=1.0 , PRDFL= 10.0 , OUTDEL = 5.0 (35)
METHOD RKS (36)
***** (37)
TITLE PS REACTOR SIMULATION PRINT-OUTPUT-TITLE (38)
PRINT X, TRC, C (39)
***** (40)
LABEL PS REACTOR SIMULATION GRAPH-OUTPUT-TITLE (41)
OUTPUT X=( 0.0, 1.1), TRC=( 0.0, 400) (42)
***** (43)
END (44)
;; (45)

```



```

PS REACTOR SIMULATION PRINT-OUTPUT-TITLE
TIME X TRC C
0.0 0.0 2.000000+01 7.000000+00
1.000000+01 4.38991D-06 3.26612D+01 6.99997D+00
2.000000+01 2.07464D-05 4.35569D+01 6.99985D+00
3.000000+01 6.82406D-05 5.30135D+01 6.99952D+00
4.000000+01 1.82713D-04 6.12845D+01 6.99872D+00
5.000000+01 4.21568D-04 6.85751D+01 6.99705D+00
6.000000+01 8.67530D-04 7.50568D+01 6.99393D+00
7.000000+01 1.63080D-03 8.08777D+01 6.98858D+00
8.000000+01 2.85125D-03 8.61695D+01 6.98004D+00
9.000000+01 4.70259D-03 9.10526D+01 6.96708D+00
1.000000+02 7.40117D-03 9.56420D+01 6.94819D+00

```

した代数方程式などの場合と全く同様で, 常微分方程式ももとの形そのままを書くことができる。

図 5 にそのソーステキストのリストと計算結果の一部を示す。リストの 3 行目から 22 行目までは使用する変数名の定義であり, 24~37 行目が方程式の記述になっている。式 (1) ~ (7) および (9) がそのまま記述されていることが理解できよう。微分方程式の微分項はアポストロフィ (') を微分記号として,

$$\frac{dy}{dt} \rightarrow y', \quad \frac{d^2y}{dt^2} \rightarrow y''$$

のように高階のものまで簡単に表現できる。この表記法では独立変数 (上の例では t) が何であるか明らかでないが, これは 40 行目にある INTEGRAL 文によって指定されている。

図5 スチレンの重合反応のシミュレーション
(EQUATRAN-Mのソースリストと計算結果の一部)

```

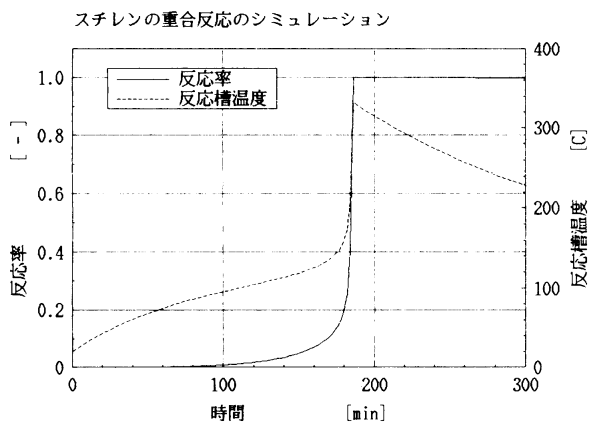
1: /* スチレンの重合反応のシミュレーション */
2:
3: VAR F      "供給量"      [m3/min] ..
4:   .Tf      "供給温度"    [K] ..
5:   .cf      "供給モノ-濃度" [kgmol/m3] ..
6:   .Tr      "反応槽温度"  [K] ..
7:   .tr      "反応槽モノ-濃度" [kgmol/m3] ..
8:   .c       "反応槽容積"  [m3] ..
9:   .V = 1   "ジャケット温度" [K] ..
10:  .Tj      "ジャケット温度" [K] ..
11:  .tj      "ジャケット温度" [C] ..
12:  .A = 2.0 "伝熱面積係数" [kcal/m2.min.K] ..
13:  .U = 2.0 "伝熱面積係数" [kcal/m2.min.K] ..
14:  .rho = 900 "濃密度" [kg/m3] ..
15:  .cp      "比熱" [kcal/kg.K] ..
16:  .x       "反応率" [-] ..
17:  .t       "時間" [min] ..
18:  .dH = -17600 "反応熱" [kcal/kgmol] ..
19:  .k0 = 1.39E9 "頻度因子" [/min] ..
20:  .E = 2.13E4 "活性化エネルギー" [kcal/kgmol] ..
21:  .R = 1.987 "ガス定数" [kcal/kgmol.K] ..
22:
23:
24:  r = k*c
25:  k = k0*EXP(-E/R/Tr)
26:  V*c' = F*(c-c) - V*r
27:  V*rho*c*p*Tr' = F*rho*IcpdT + QR - Qj
28:  Qj = U*A*(Tr-Tj)
29:  QR = -dH*V*r
30:  cp = a + b*Tr
31:  a = 0.088 ; b = 0.103E-2
32:  IcpdT = a*(Tr-Tr) + b/2*(Tr^2-Tr'^2)
33:
34:  x = (cf-c)/cf
35:  tj = Tj-273.15
36:  tr = Tr-273.15
37:  tf = Tf-273.15
38:
39: INPUT F, cf, tf, tj
40: INTEGRAL t[0,300] STEP 1 BY RKV
41: Tr # Tf
42: c # cf
43: TREND x, tr, c STEP 10
44: OUTPUT1 t, x, tr, c STEP 2

```

[入力データ]			
F	=	0.001	: 供給量 [m3/min]
cf	=	7	: 供給モノ-濃度 [kgmol/m3]
Tf	=	20	: 供給温度 [C]
tj	=	140	: ジャケット温度 [C]

t	1:x	2:tr	3:c
0	0	20.00000	7.000000
10.000000	4.38990E-006	32.66117	6.999969
20.00000	2.07464E-005	43.55693	6.999855
30.00000	6.82605E-005	53.01350	6.999522
40.00000	0.0001827128	61.28455	6.998721
50.00000	0.0004215674	68.57511	6.997049
60.00000	0.0008675300	75.05679	6.993927
70.00000	0.001630799	80.87772	6.988584
80.00000	0.002851254	86.16947	6.980041
90.00000	0.004702591	91.05260	6.967082
100.00000	0.007401167	95.64201	6.948192

図6 シミュレーションの結果 (起動時)
EQUATRAN-M 作図機能による



INTEGRAL t [0, 300] STEP 1 BY RKV

では、独立変数を t とし、積分区間 $[0, 300]$ をきざみ 1 で積分すること、また、積分手法として RKV 法 (前出の図 2 の公式による方法) を用いることが指定されている。(8) 式で与えられている積分の初期条件は、EQUATRAN-M ではリストの 41~42 行目のように ' #' を用いて書くことにより、一般の方程式と区別されている。

43 行目の TREND 文は、積分過程の変数の変化を周期的に表示するためのもので、図 5 の出力はこの文の働きによる。44 行目の OUTPUT 1 文は同じく計算の過程をファイルに出力するためのもので、このファイルから図 6 のようなグラフを作ることができる。

ここで EQUATRAN-M による常微分方程式の取り扱いに関して、特徴的なこと (主に SLCS IV などのシミュレーション言語との比較において) についていくつか触れておきたい。

EQUATRAN-M では常微分方程式も他の方程式同様、文字通り方程式として扱われる。このことは方程式の変形が自動的に行われることによる使いやすさだけでなく、式と変数との充足関係が厳密にチェックされるため、計算結果の信頼性を著しく高めるといふ効果ももたらしている。たとえば、EQUATRAN-M では 1 つの変数を 2 つの計算式で計算していることに気がつかないといった誤りは起こり得ない。

常微分方程式とともに線形、非線形の連立方程式が混在する問題が、そのまま記述できるのも特徴の一つと言えよう。線形連立方程式の直接解法や、非線形連立方程式の反復解法は自動的に組み込まれて実行される。

さらに EQUATRAN-M では、ベクトルやマトリクスを含む微分方程式が特別な配慮なしにそのまま書ける。たとえば Y をベクトルとすると、 dY/dt は単に Y' と書いてよく、またその要素を添字をつけて、 $Y'(3)$ のように表わすこともできる。EQUATRAN-M には、他の連続系シミュレーション言語のように、FORTRAN プログラムを直接モデルの記述に組み入れる機能はないが、配列変数の扱いなど記述能力が高いため、むしろそのような機能は多くの場合必要がないと言ってよい。なお、前項で触れた定常解を求める機能については次項で、条件変更に関する機能については回を改めて詳述する。

連続重合反応の温度制御シミュレーション

前述のステレン重合反応器において、安定した運転を継続するためには反応槽の温度制御が不可欠なことは既に述べた。次は、図7に示すカスケード制御系をもつ反応器について、制御の安定性を検証するためのシミュレーションを行ってみよう。2つのコントローラTC1とTC2は、いずれも通常のPIコントローラであるが、TC2によるジャケット温度の制御は、反応槽の挙動に比べて十分速く追従するように設計されているので、TC1の出力（すなわちジャケット温度の設定値）とジャケット温度とは同一と考えてよいものとする。したがって方程式としてはコントローラTC1について、

$$e = T_r - T_{set} \quad (10)$$

$$T_j = KP \cdot \left(e + \frac{1}{T_i} \int e \, dt \right) \quad (11)$$

を追加すればよい。 T_{set} は反応槽の設定温度 [K]、 KP はTC1のゲイン [K/K]、 T_i は積分時間 [min] である。なお、式(11)は $U1 = \int e \, dt$ とおくことにより、

$$T_j = KP \cdot \left(e + U1/T_i \right) \quad (12)$$

$$\frac{dU1}{dt} = e \quad (13)$$

と書きなおすことができる。

シミュレーションはある定常な運転状態において、外乱としてモノマーの供給量 F がステップ状に変化した場合について行うものとしよう。定常状態での供給量を F_s [m^3/min]、また $t = 10min$ の時点で F が10%ステップ状に減少するものとするれば、

$$F = F_s (1 - 0.1 \cdot u(t > 10)) \quad (14)$$

ここで $u(t > 10)$ は、 $t > 10$ で1、 $t \leq 10$ で、0の値をとるステップ関数を表わす。その他、追加した計算条件を表3に示してある。

数式モデルとしては前回のものに(10)、(12)、(13)、(14)式を追加すればよいが、今回はシミュレーションを開始する前に定常状態の計算を行って、 T_r 、 e 、 F_s および $U1$ の値を求めておく必要がある。一般にシミュレーションによって制御系の検討などを行う場合、その初期条件(必ずしも定常状態とは限らない)を求めることが必要であ

図7 スチレンの重合反応器
(カスケード温度制御)

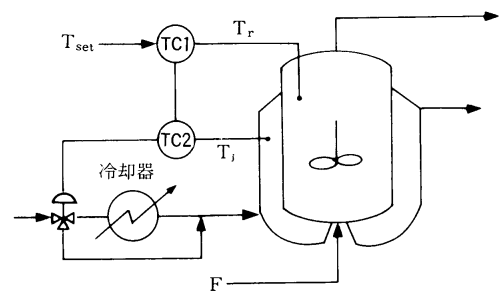


表3 重合反応シミュレーションの計算条件 (温度制御)

定常状態の反応率 (x)	0.5
反応槽設定温度 (t_{set})	140℃
TC1のゲイン	-10 K/K
TC1の積分時間	100 min

り、これがなかなかやっかいである。定常状態を求めるには、前述のように微分項を0としてできる代数方程式を解けばよく（といっても簡単ではない）、このような機能を有するシミュレータ（DPSなど）もある。しかし、実際の問題では初期条件を求めるための計算条件と、シミュレーションを行う際の計算条件とは一致していないことが多く、たとえば本例題のようにシミュレーションでは供給量 F が与えられるが、初期条件としては必要な反応率 ($x = 0.5$) が与えられており、これに見合った供給量を計算する必要があるといった場合には、このような機能だけでは不十分である。初期条件を求めるためにわざわざそのためのプログラムを作らなければならないことが多く、これがまた別のトラブルの原因となりかねない。

この点、EQUATRAN-Mには常微分方程式を含んだモデルをそのまま使って初期条件を計算する機能があるので、簡単にこの問題を解決できる。原理は極めて簡単で、まず微分項 (T_r' 、 e' など) をそれぞれ全く独立な変数であるとみなす (INTEGRAL文を書かないことによつてこのようにみなされる)。

図8 温度制御シミュレーションの初期値計算
(EQUATRAN-Mのソースリストと計算結果)

```

1: /* スチレンの重合反応のシミュレーション (温度制御) */
2:
3: VAR F      "供給量"      [m3/min] ..
4:   .Fs      "初期供給量"  [m3/min] ..
5:   .Tf      "供給温度"    [K] ..
6:   .cf      "供給モノマー濃度" [kgmol/m3] ..
7:   .Tr      "反応器温度"   [K] ..
8:   .tr      "反応器温度設定" [C] ..
9:   .tset    "反応器温度設定" [C] ..
10:  .c      "反応器モノマー濃度" [kgmol/m3] ..
11:  .V = 1   "反応器容積"     [m3] ..
12:  .Tj      "ジャケット温度" [K] ..
13:  .tj      "ジャケット温度" [C] ..
14:  .A = 2.0 "伝熱面積係数"   [m2] ..
15:  .U = 2.0 "伝熱伝熱係数"   [kcal/m2.min.K] ..
16:  .rho = 900 "密度"         [kg/m3] ..
17:  .cp      "比熱"          [kcal/kg.K] ..
18:  .x      "反応率"         [-] ..
19:  .t      "時間"           [min] ..
20:  .dH = -17600 "反応熱"     [kcal/kgmol] ..
21:  .k0 = 1.39E9 "頻度因子"   [/min] ..
22:  .E = 2.13E4 "活性化エネルギー" [kcal/kgmol] ..
23:  .R = 1.987 "ガス定数"     [kcal/kgmol.K] ..
24:
25:
26: MACRO PIC /* PI コントローラ */
27:   e = Y - (SET)
28:   U1 = e
29:   U = KP*(e-U1/TI)
30: END PIC
31:
32: r = k*c
33: k = k0*EXP(-E/R/Tr)
34: V*dC = F*(cf-c) - V*r
35: V*rho*c*Tr = F*rho*c*Tr + F*rho*icpD + QR - Qj
36: Qj = U*A*(Tr-Tj)
37: QR = -dH*V*r
38: cp = a + b*Tr
39: a = 0.088 ; b = 0.103E-2
40: icpD = a*(Tf-Tr) + b/2*(Tf^2-Tr^2)
41:
42: x = (cf-c)/cf
43: tj = Tj-273.15
44: tr = Tr-273.15
45: tf = Tf-273.15
46: tset = Tset-273.15
47:
48: TC1: CALL PIC( Y=Tr, SET=Tset, U=Tj, KP=-10, TI=100 )
49:
50: F = Fs * (1-0.1*STEP(t>10)) /* ステップ変化 */
51:
52: /* 初期値計算 */
53: Tr = 0 ; c = 0 ; Tj = Tset ; t = 0
54: INPUT x, cf, tf, tset
55: OUTPUT ALLI, Fs, cf, tf, tset
56: OUTPUT1 ALLI, Fs, cf, tf, tset

```

[入力データ]			
x	= 0.5	: 反応率	[-]
cf	= 7	: 供給モノマー濃度	[kgmol/m3]
tf	= 20	: 供給温度	[C]
tset	= 140	: 反応器温度設定	[C]
[計算結果]			
Fs	= 0.007494069	: 初期供給量	[m3/min]
cf	= 7	: 供給モノマー濃度	[kgmol/m3]
tf	= 20	: 供給温度	[C]
tset	= 140	: 反応器温度設定	[C]
Tr	= 413.15	: 反応器温度	[K]
c	= 3.5	: 反応器モノマー濃度	[kgmol/m3]
Tr1,U1	= -3891.47259		

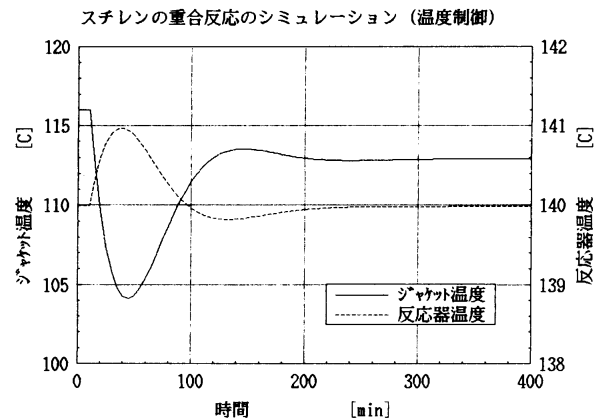
図9 温度制御シミュレーション
(EQUATRAN-Mのソースリスト, 図8の変更分)

```

52: /* 積分計算 */
53: INTEGRAL t[0,400] STEP 1 BY RKV
54: INPUT Fs, cf, tf, tset
55: INITIAL ALLI
56: OUTPUT2 t,x,tr,c,tj,Qj,F STEP 5

```

図10 温度制御シミュレーションの結果



こうしてできた代数方程式などには、微分項を独立変数とみなしたことによって微分項の数+1 (+1は独立変数の分)の自由度が生じている。

したがって、この数だけの条件を追加してこの方程式を解くことができる。たとえば微分項にあたる変数をすべて0とし、独立変数に出発値を与えれば定常解が得られることになる。与える条件は何でもよいので、定常解以外の条件を与えることもでき、また、本例題のように反応率xを与えて供給量Fsを計算するように変更するなど自由に行えるわけである。

それでは実際に計算を行ってみることにする。図8は初期値計算のためのソーステキストとその計算結果であ

るが、リストの51行目まではシミュレーションのためのものと同一であり、52行目以下を図9のように変更すれば、そのままシミュレーションのためのソーステキストとなる。26~30行目のマクロ定義はPIコントローラの式(10),(12),(13)をまとめたもので、48行目のCALL文によってコントローラTC1のための式として取り入れられている。供給量を与える式(14)は、ステップ関数STEPを用いて50行目に加えられている。初期値計算の条件(この場合自由度は4つ)は53行目のように書ける。54行目のINPUT文で(Fsの代わりに)xを入力しているので、入力したx(=0.5)を満たすFsの値が計算されることになる。56行目のOUTPUT1文は計算

結果をファイルに出力するためのものであり、この出力はのちにシミュレーションの初期入力として読み込むことができる。この文で出力項目に指定されている ALLI は「すべての被積分変数」を意味している。図 8 の計算結果のうち、 T_r, c および TC 1. U 1 が ALLI による出力である。なお、TC 1. U 1 はマクロコール文 TC 1 によって呼び出されたマクロ内の U 1 という変数に与えられる変数名である。

図 9 によって図 8 の 52 行目以下を置き換えれば、供給量のステップ変化に対する応答がシミュレーションできる。図 9 の 55 行目の INITIAL 文は積分初期値を入力するためのもので、これによって先ほど計算した T_r, c および TC 1. U 1 が読み込まれる。シミュレーションの結果をグラフ化して図 10 が得られた。この結果では、反応器温度の制御はたいへんうまく行われているが、実際にはジャケット温度制御の遅れの影響なども無視できず、より詳細な検討が必要である。

おわりに

連続系のシミュレーションの分野では、古くから方程式解法ソフト的なアプローチが採用されており、この種のシミュレーション言語は化学工業でも多くの実績をあげている。さらに方程式解法ソフトとしての性格が明確な EQUATRAN-M は、機能面でも使い易さの面でも、際だった特徴を有しており、この分野の汎用ソフトとして定着していくものと期待される。

※本連載で使用している EQUATRAN-M バージョン 2 は、9 月から一般に発売され始めた。価格は本体（代数方程式などが扱える）が 28 万円、積分作図オプション（本体に付加して常微分方程式とグラフ作成を可能にする）が 12 万円となっている。

また、大学向にはオプションを含んだものが 10 万円で提供されている。

(おぐち ごろう, おおむら けんいち 三井東圧化学(株))

＜参考文献＞

- | | |
|---|---|
| (1) M.E. Leesley (Editor); COMPUTER-AIDED PROCESS PLANT DESIGN, GULF Publishing Company, 1982, p695-710 | 装置設計”, ケミカルエンジニアリング, 第 12 巻第 10 号, (1967) p940-945 |
| (2) 中西俊男; “シミュレーション言語の新しい動向”, システムと制御, Vol 29, No 11, (1985) p706-713 | (4) 高分子学会; スチレン系ポリマーの製造プロセス, 1977 |
| (3) 木本浩二, 柳沢孝成; “連続重合反応のシミュレーションと | (5) 田中正次; “打ち切り誤差評価能力をもつ Runge-Kutta 型公式について”, 情報処理, Vol. 9, No 5 (1968) p261-271 |
| | (6) 富士通; SLCS IV 解説書 (言語編), 1979 |