

# ケミカルエンジニアのための計算機演習

## - 計算機演習の基礎 -

(株)オメガシミュレーション

横山 克己

## 1. 数値計算

### 1.1 化学工学での数値計算

化学工学分野でコンピュータを用いた数値計算について少し振り返ってみましょう。

1960年代から技術計算に大型計算機が使われはじめ、1970年代には数値計算がさかんに行われるようになりました。プログラミング言語には FORTRAN が用いられました。線形連立方程式の解法アルゴリズムや、非線形方程式を収束計算で解くためアルゴリズムなどは数値計算手法と呼ばれ、汎用性があるためにサブルーチン化されて、共用できるように数値計算ライブラリとして整備されました。

また、この時期からこのライブラリを土台にして、プロセスフローシミュレータや蒸留計算プログラムのような特定の目的に使用する大型なプログラムが開発されました。ただ、大型計算機は企業であれば本社に、大学であれば計算機センターにあって、だれでも気軽に利用できるという環境ではありませんでした。

1980年代に入ってパソコンが急速に普及し、パソコン上でプログラミング言語に BASIC を用いて数値計算が行われました。身近に利用できるようになったわけです。さらに、パソコンのグラフィック機能を使って、計算結果をグラフ化して視覚的に評価できるようになりました。ただ、スピードや大きなメモリ空間が要求されるような流動解析やプロセスシミュレーションなどの分野では、パソコンでは能力不足でエンジニアリング・ワークステーションが使われました。大型計算機からのダウンサイジングが確実に進んだわけです。

1980年代後半から1990年代にかけて、数値計算をより簡単に行えるようなパソコン用のパッケージソフトがいくつか登場してきました。従来のようにひとつひとつアルゴリズムを考えてプログラムを作る必要がなくなり、方程式レベルで検討ができるため、利用者から見ると大きな進歩といえます。ここではこれを総称して「数値計算ソフト」と呼び、あとでくわしく紹介していくことにします。最近では、他の分野でもそうですが、パッケージソフトが充実してきたことや、利用者の範囲が拡大したことで、プログラムを作成して問題解決をはかることがほとんどなくなり、数値計算ソフトや表計算ソフトなどが活用されるようになってきています。

さて、化学工学分野での数値計算を扱った書籍をあたってみますと、1976年に化学工学協会（化学工学会の前身）編で、「化学工学プログラミング演習」<sup>1)</sup>という書籍が刊行されています。化学工学分野から演習問題を収集して、コンピュータを利用した解法と FORTRAN によるプログラムが示されています。1985年に、ほとんど同じ趣旨で FORTRAN を BASIC に置き換えた、「BASIC による化学工学プログラミング」<sup>2)</sup>が発刊されています。また、同じく1985年には古

崎らによって「マイコンによる化学工学計算」<sup>3)</sup> が出版されており、ここでも BASIC が使われています。

これらの書籍に上げられている数値計算分野を拾ってみますと、代数式計算、補間・微分・積分、線形方程式、非線形方程式、パラメータ推定、常微分方程式、偏微分方程式、確率・統計と多岐にわたっており、数値計算の主要な分野をすべて含んでいます。化学工学は様々な数値計算が利用される分野であることが分かります。

## 1.2 数値計算手法

数値計算ソフトや数値計算ライブラリを用いるときには、数値計算および数値計算手法の特質を理解して利用しますと、誤った解を正しい解と見なすような過ちを犯すこともなくなりますし、トラブルを起こした場合の対処に役に立つことが多いと思います。

ほとんどの方はどこかで聞いたことがあると思いますが、ここで少し復習をしておきましょう。浮動小数点数の性質を説明したあと、代表的な3つの数値計算手法の考え方を説明します。アルゴリズムを正確に理解することが目的ではありませんので、考え方を直感的に理解してください。

### 浮動小数点数

コンピュータでは実数を表現するために浮動小数点表現がもちいられます。多くの桁数を有限の桁数で近似することになりますので誤差が生じ、これをまるめ誤差といいます。また、同時に指数部の範囲を超えるような大きな数や小さな数が現れたときには、オーバーフローあるいはアンダーフローと呼ばれて表現できなくなります。数値計算で通常使われる倍精度（64ビットで表現）では、有効桁数が約15.95桁、表現できる範囲は $2.22 \times 10^{-308} \sim 1.79 \times 10^{308}$ となります。

2つの浮動小数点数の加算を行う場合に、大きさが極端に違うと小さな数値の下位の情報が失われ、情報落ちと呼ばれる現象がおきます。また、大きさがほぼ等しい2つの浮動小数点数の減算を行うと、上位の桁が打ち消し合って下位の桁の情報しか残らないことになり、これは桁落ちと呼ばれます。いずれも計算精度に影響を及ぼすこととなります。

### ガウスの消去法（線形連立方程式の解法）

つぎの(1)式で表されるような線形連立方程式を考えます。

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1N}x_N &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2N}x_N &= b_2 \\ &\vdots \\ a_{N1}x_1 + a_{N2}x_2 + \cdots + a_{NN}x_N &= b_N \end{aligned} \tag{1}$$

これを(2)式のような形に変形します。

$$\begin{aligned}
x_1 + c_{12}x_2 + c_{13}x_3 + \cdots + c_{1N}x_N &= d_1 \\
x_2 + c_{23}x_3 + \cdots + c_{2N}x_N &= d_2 \\
&\vdots \\
x_N &= d_N
\end{aligned}
\tag{2}$$

そのためには、1行目の式の両辺を  $a_{11}$  で割るとつぎのようになります。

$$x_1 + a'_{12}x_2 + \cdots + a'_{1N}x_N = b'_1 \tag{3}$$

これに  $a_{21}$ 、 $a_{31}$ 、 $\dots$ 、 $a_{N1}$  と掛け合わせて(1)式の2、3、 $\dots$ 、N行目の式よりそれぞれ引けば、(4)式が得られます。

$$\begin{aligned}
x_1 + a'_{12}x_2 + \cdots + a'_{1N}x_N &= b'_1 \\
a'_{22}x_2 + \cdots + a'_{2N}x_N &= b'_2 \\
&\vdots \\
a'_{N2}x_2 + \cdots + a'_{NN}x_N &= b'_N
\end{aligned}
\tag{4}$$

つぎに2行目以降について同様な操作を行うという具合に処理すると、(2)式を得ることができるわけです。

(2)式が得られれば、 $x_N$ の値は  $x_N = d_N$  で与えられますので、それをN-1行目の式に代入することで、 $x_{N-1}$ の値が求まります。以下同様に1行目の式までさかのぼれば、すべての  $x_i$  が求まることとなります。

### ニュートンラフソン法 (非線形方程式の解法)

方程式  $f(x)=0$  の解を繰り返し収束計算で求めることを考えます。この手法では独立変数  $x$  の  $k$  回目の仮定値を  $x_k$  とした時、つぎの仮定値  $x_{k+1}$  を

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \tag{5}$$

によって求めます。ここで、 $f'(x_k)$  は  $x_k$  における  $f(x)$  の微係数の値です。これを図で表現してみたのが図1です。

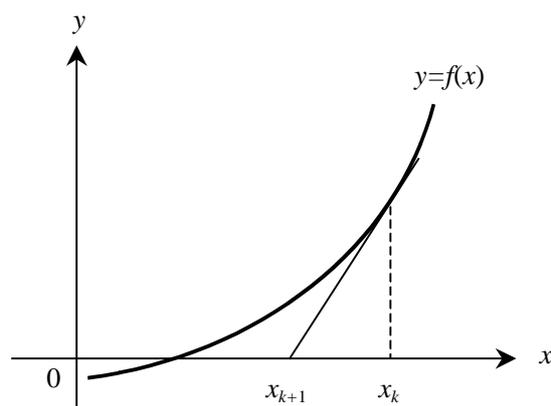


図1 ニュートンラフソン法による収束の様子

$x_k$  において  $y=f(x)$  の接線を引き、 $x$  軸との交点を  $x_{k+1}$  にしています。 $f(x)$  が 0 になったとき、すなわち  $y=0$  と交わったところが解ですが、 $x_{k+1}$  は  $x_k$  より解に近いことが分かります。与えられた  $x$  の初期値  $x_0$  として、上の手順を繰り返し、 $f(x_k)$  の値が指定した範囲（計算精度）に入った時に収束とみなして解としています。

このことから、関数の形、そして初期値の与え方が、収束に大きく影響することが理解できると思います。複数の解がある場合に初期値によっては求めたい解ではない他の解に収束したり、解とは反対の方向に進んで発散したり、試行点が戻ってしまっていて振動していつまでも収束できないなどが考えられます。

なお、ニュートンラフソン法では各  $x$  の値における  $f(x)$  の微係数の値が必要で、通常次式の数値微分により求められます。ここで  $\Delta x$  は微小変化量です。

$$f'(x_k) = \frac{f(x_k + \Delta x) - f(x_k)}{\Delta x} \quad (6)$$

#### 4 次のルンゲクッタ法（常微分方程式の解法）

$y$  を  $x$  の関数として

$$\begin{aligned} \frac{dy}{dx} &= f(x, y) \\ y(x_0) &= y_0 \end{aligned} \quad (7)$$

という初期値問題を解くことを考えます。数値積分のきざみ幅を  $h$  とし、第  $i$  番目のきざみ点における  $x$  および  $y$  を  $x_i$  および  $y_i$  で表すとき、第  $i+1$  番目のきざみ点での  $y$  の値  $y_{i+1}$  をつぎの式で与えるのが 4 次のルンゲクッタ法です。

$$y_{i+1} = y_i + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} \quad (8)$$

ただし、 $k_1$ 、 $k_2$ 、 $k_3$ 、 $k_4$  は以下のように求めます。

$$k_1 = hf(x_i, y_i) \quad (9)$$

$$k_2 = hf\left(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right) \quad (10)$$

$$k_3 = hf\left(x_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right) \quad (11)$$

$$k_4 = hf(x_i + h, y_i + k_3) \quad (12)$$

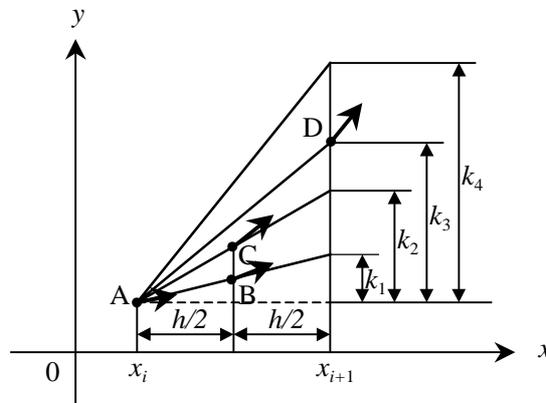


図 2 を使って積分の様子をみてみます。

図 2 4 次のルンゲクッタ法の積分の様子

- 点 A ( $x_i, y_i$ ) で微係数を(9)式で求める。
- $x$  方向に  $h$  だけ進むと、 $y$  方向には  $k_1$  だけ増加する。 $h/2$  進んだ点 B で方向を修正する。(10)式を使って点 B ( $x_i + h/2, y_i + k_1/2$ )での微係数を求める。
- この微係数を使って点 A から  $x$  方向に  $h$  進むと、 $y$  方向には  $k_2$  増加する。ここでも  $h/2$  進んだ点 C で方向を修正する。(11)式を使って点 C ( $x_i + h/2, y_i + k_2/2$ )での微係数を求める。
- この微係数を使って点 A から  $x$  方向に  $h$  進むと、 $y$  方向には  $k_3$  増加する。この点 D は( $x_i + h, y_i + k_3$ )で与えられ、(12)式を使ってこの点での微係数を求める。
- この微係数を使って点 A から  $x$  方向に  $h$  進むと、 $y$  方向には  $k_4$  増加する。
- 以上で求めた  $k_1$ 、 $k_2$ 、 $k_3$ 、 $k_4$  で 1:2:2:1 なる荷重平均をとる。

このことから、積分きざみ  $h$  をどのように取るかで計算精度が決まることが分かります。積分きざみ  $h$  を十分小さく取れば精度が上がりますが、それだけ積分計算回数が増大し、計算時間がかかるようになってしまいます。打ち切り誤差の推定を加えて、この積分きざみを自動的に調整する手法が可変きざみルンゲクッタ法です。

## 2. 数値計算ソフトの利用と比較

### 2.1 数値計算ソフトの利用

1.1 節でも触れたように、最近ではパッケージソフトの分類で「数学ソフト」なるジャンルもできて、数値計算ソフトが何種類か利用できるようになってきました。日本で手に入る数値計算ソフトを表1にまとめました。

表1 数値計算ソフト

ソフト名	タイプ	開発元	代理店
EQUATRAN-G	方程式解法ソフト	オメガシミュレーション	—
TK! Solver	方程式解法ソフト	Universal Technical Systems	伯東
HiQ	方程式解法ソフト	National Instruments	日本ナショナルインスツルメンツ
Mathematica	数式処理ソフト	Wolfram Research	住商エレクトロニクスほか
MAPLE	数式処理ソフト	Maple	サイバネットシステム
Mathcad	数式処理ソフト	Mathsoft	住友金属システム開発
カルキング	数式処理ソフト	シンプレックス	—
MATLAB	連続系シミュレーション	MathWorks	サイバネットシステム
ACSL	連続系シミュレーション	MGA Software	サイバネットシステム
VisSim	連続系シミュレーション	Visual Solutions	日立情報ネットワーク

これらのソフトはおおまかに3つのタイプに分類できると思います。

#### ・方程式解法ソフト

連立方程式を解くことのできるソフトで、方程式には非線形方程式や常微分方程式なども含まれます。同時に最適化計算のできるものもあります。方程式の記述は、解く順序に並べる必要がなく、関係式を列挙するようになっています。

#### ・数式処理ソフト

数式をさまざまに加工することができるソフトです。ある式の積分形や微分形を求めたり、因数分解などの式変形を行ったり、解析解を求めたりできます。また、数式処理に加えて数値計算も行えます。

#### ・連続系シミュレーションソフト

連立常微分方程式で表現された問題を解いて、連続系のダイナミックシミュレーションを行うソフトです。他の2つのタイプに比べて、ダイナミックシミュレーションに焦点を絞っています。

これらのソフトは数値計算を行えるばかりでなく、計算結果をビジュアルにグラフ表示できるもの、数式そのものを入れたドキュメントを作成できるもの、数式で表現したモデルをオブジェクトにしてブロック図上でシミュレーションができるものもあります。

## 2.2 代表的な数値計算ソフトの比較

前節で上げたソフトの中から、代表的な数値計算ソフトとして、米国産で日本でもポピュラーな Mathematica および Mathcad と、純国産で歴史のある EQUATRAN-G を、化学工学分野に適用したときの比較結果を表 2 に引用しました。

表 2 数値計算ソフトの比較<sup>5)</sup>

	EQUATRAN-G	Mathematica	Mathcad
長所	<ul style="list-style-type: none"> <li>・ 計算順序に関係なく方程式を記述することができる。</li> <li>・ 線形・非線形方程式、常微分方程式、最適化計算などを混在して記述できる。</li> <li>・ 大規模方程式など未知数の多い方程式などを解くことが可能。</li> </ul>	<ul style="list-style-type: none"> <li>・ グラフィックス能力が優れており、多様なグラフが作成可能。</li> <li>・ 組み込み関数が豊富で作成の手間が省ける。</li> </ul>	<ul style="list-style-type: none"> <li>・ 文章の混在が容易で、方程式や数式記号をそのまま表示できるなどドキュメント性に優れている。</li> <li>・ 操作方法が簡便であり、高級電卓などとしても使える。</li> </ul>
短所	<ul style="list-style-type: none"> <li>・ グラフ表現が限られている（2Dのみ）。</li> <li>・ 組み込み関数の数が少なく、自分で作成しなければならない。</li> <li>・ 方程式や数式記号などの表示はプログラムのであり、ドキュメント性はあまり優れていない。</li> </ul>	<ul style="list-style-type: none"> <li>・ 計算順序どおりに方程式を記述しなければならない。場合によっては式変形する必要がある。</li> <li>・ 線形・非線形方程式、常微分方程式、最適化計算の混在ができないため、それぞれ個別に記述して組み合わせる必要がある。</li> </ul>	<ul style="list-style-type: none"> <li>・ 計算順序どおりに方程式を記述しなければならない。</li> <li>・ 線形・非線形方程式と常微分方程式との連立ができず、積分計算手法の記述が必要になる*。</li> <li>・ 最適化計算の機能がないため解析して解く必要がある。</li> <li>・ 線形・非線形方程式においては、未知数の多い方程式は解けない。</li> </ul>

\* 最新版の Mathcad 7 では rkfixed() という関数が用意されているが、Runge-Kutta 法を適用する形に式変形する必要がある。

## 2.3 プログラミング言語との違い

数値計算ソフトを利用するのと、プログラミング言語でプログラムを作るのではどこが違うのでしょうか。方程式解法ソフト EQUATRAN を例に図 3 で説明しましょう。この図は、問題を解く場合のアプローチの違いを示しています。

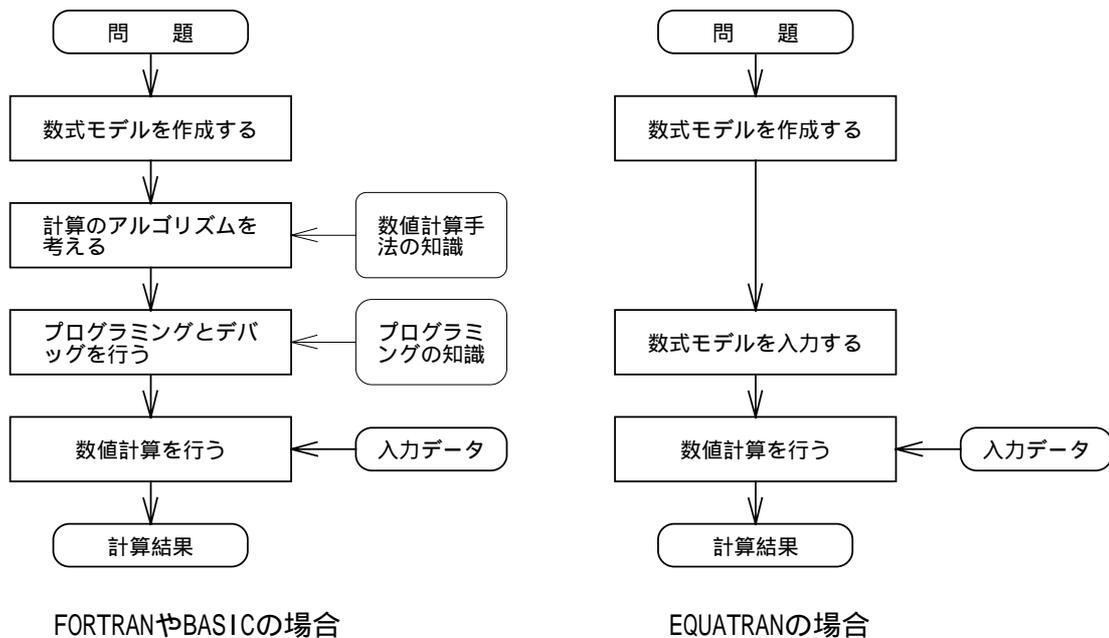


図3 問題解法のアプローチ

左側がFORTRANやBASICなどのプログラミング言語の場合です。数式モデル（方程式）を作成したら、まず計算の手続き、すなわちアルゴリズムを考えます。このとき、線形・非線形方程式や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、一番やっかい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さえすればよいので、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こととなります。

### 3. EQUATRAN-G の紹介

ここからは、数値計算ソフトの中から EQUATRAN-G を紹介していきます。

EQUATRAN-Gは、1985年に三井東圧化学が開発して販売をはじめたEQUATRAN-Mが原型になっています。その後改良を重ね、機能アップして1989年にEQUATRAN-Gが登場しました。現在は、業務を引き継いだオメガシミュレーションが開発および販売を行っており、最新バージョンはVer.3になっています。

#### 3.1 機能

つぎにEQUATRAN-Gの各機能をすこし細かく見ていきましょう。

##### 方程式解法機能

EQUATRAN-G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）そして最適化計算、最小2乗法計算（非線形を含む）を数値的に解くことができます。

方程式はそのままの形で入力すればよく、変形したり、解く順序に並び替える必要はありません。解く順序は、EQUATRAN-Gが自動的に生成してくれます。

線形・非線形連立方程式の場合は、そのまま入力するだけです。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-G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方法を指定することも可能です。常微分方程式の場合は、積分の指定をする文をひとつ記述するだけで、高階の方程式でもそのまま扱うことができます。最適化計算も独立変数と評価変数を指定するだけです。

EQUATRAN-Gでは、以上の方程式を解く問題のほかに、これらの方程式が混在した複合問題、たとえば常微分方程式と線形・非線形連立方程式が混在した問題、最適化計算の中に非線形連立方程式が含まれるような問題なども、扱うことができます。さらにユーザー関数を利用すれば、高度な複合問題（多重積分、2点境界値問題、動的システムのパラメータ同定問題、MINIMAX問題など）も扱えます。

##### 方程式記述機能

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。

- ・ユーザー関数とマクロ 規模の大きな数式モデルはモジュール化して記述できます。

#### 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、1～3次式による近似曲線など、科学技術分野向きのグラフ機能が用意されています。自動設定機能により、簡単にグラフが作れ、しかも、各種のメイクアップ機能により、完成度の高いグラフが得られます。

#### レポート作成機能

レポートの形に作成したテキスト(フォーム)中に、計算結果を任意の位置に、任意のフォーマットで埋め込むことができます。これにより、レポートの作成が容易にできます。

### 3.2 応用例

#### ・カーブフィッティング

一番身近な応用は、データから近似式を作るカーブフィッティングでしょう。観測データや文献などのデータから近似式を求めるのに最小2乗計算をし、データと近似式をグラフにプロットして評価できます。理系の学部を卒業した人なら必ずや経験があるでしょうし、実験を伴う研究業務にはつきものといえるでしょう。

#### ・バランス計算

線形・非線形連立方程式を使えば、時間依存のない電気回路の計算、化学平衡計算、管路網の計算やプロセスの物質収支計算などのいわゆるバランス計算を解くことができます。

#### ・ダイナミックシミュレーション

連続系のダイナミックシミュレーションは、常微分方程式でモデル化できます。振動系のシミュレーション、電気回路の過渡現象の解析、制御系の解析や設計、反応速度の検討、反応器のようなプラント機器の設計、血液循環系のシミュレーションや社会システムのシミュレーションなどがあります。

### 3.3 例題集

特に化学工学分野では適応例が数多くあり、例題集<sup>6)</sup>にまとめられています。表3にその例題一覧を示しました。

表3 例題集 [ 化学工学編 ] 例題一覧

分類	タイトル
熱力学および物性	SRK状態方程式による圧縮係数の計算
熱力学および物性	Antoine式による蒸気圧の近似
熱力学および物性	定圧モル比熱の2次式への近似
熱力学および物性	Wilson式のパラメータ推定および再現計算
熱力学および物性	NRTL式による液液平衡計算
熱力学および物性	化学平衡計算
移動現象	層流境界層の速度分布の計算
粉粒体の特性	気流中の粒子の運動シミュレーション
流動	縮流管での圧力損失の計算
流動	管路網の流量と圧力の計算
流動	コーナータップオリフィスの設計
流動	サージタンクの動的シミュレーション
伝熱	平板の1次元熱伝導の計算
伝熱	正四角柱の2次元熱伝導の計算
伝熱	向流型熱交換器の計算
伝熱	並流型熱交換器の計算
伝熱	1-2 shell-tube型熱交換器の計算
伝熱	三連向流熱交換器の計算
伝熱	三連熱交換器ネットワークの最適計算
蒸発	三重効用缶の設計
蒸留	単蒸留の計算
蒸留	平衡フラッシュ計算
蒸留	沸点計算
蒸留	簡易蒸留計算
蒸留	McCabe-Thieleの図計算法による段数計算
蒸留	Thiele-Geddes法による蒸留計算
蒸留	回分蒸留計算
蒸留	充填塔の充填高さや塔径の計算
吸収	吸収塔の塔高計算
吸着・イオン交換	吸着平衡式のパラメータ推定
吸着・イオン交換	圧力スイング吸着の循環定常法計算
調湿・水冷却	空気調湿の計算
反応装置設計	連続攪拌槽型反応器の解析
反応装置設計	回分攪拌槽型反応器の解析
反応装置設計	管型反応器（押し出し流れモデル）の解析
反応装置設計	管型反応器（軸方向拡散モデル）の解析
反応装置設計	管型反応器（半径方向拡散モデル）の解析
生化学反応	酵素反応の追跡計算
プロセスの計画と設計	Williams-Ottoプロセスの物質収支計算
プロセスの計画と設計	Williams-Ottoプロセスの最適設計
プロセスの計画と設計	プロピレン製造プロセスの物質収支計算
プロセスの計画と設計	スチレン重合反応器のシミュレーション
プロセスの管理と制御	分解炉の温度制御シミュレーション

## 4. EQUATRAN-G での演習概要

EQUATRAN-G で問題を解くために必要な知識は、大まかに言って操作方法と文法です。EQUATRAN-G for Windows は標準的な Windows ソフトの操作流儀に則っていますから、Windows 95/NT を使って仕事をなさっている方は容易に習得できると思います。文法とは、方程式を記述するための決まりごとです。ただ、これもプログラミング言語の文法に比べれば、ほんのわずかでやさしいものです。

### 4.1 操作方法

ここでは操作方法を説明するために、具体的な問題を使って実際に解く手順で説明していくことにします。<sup>7)</sup>

#### 問題と解法

〔例題〕 最小2乗法

つぎのような実験データがある。

温度[K]	熱容量[cal/mol・K]
300	8.89
500	10.66
700	11.85
900	12.68
1100	13.26
1300	13.68
1500	13.99

温度を  $x$ 、熱容量を  $y$  として、最小2乗法でつぎの2次式に近似しなさい。

$$y = a + bx + cx^2 \quad (1)$$

(1)式の係数  $a$ 、 $b$ 、 $c$  を決定するわけですが、これらにより計算される熱容量の値を  $ycal$  として、 $i$ 番目の測定値に対する量に添え字  $i$  を付けて表すと

$$ycal_i = a + bx_i + cx_i^2 \quad (2)$$

となります。最小2乗法の考え方により、誤差  $e$  の2乗和  $S$

$$e_i = y_i - ycal_i \quad (3)$$

$$S = \sum_{i=1}^8 e_i^2 \quad (4)$$

を最小とする  $a$ 、 $b$ 、 $c$  を求めることとなります。(1)式は係数について線形になっていますので、正規方程式を導出して解けば一意的に決めることができます。ここでは別の方法、すなわち最小 2 乗法問題を解法する機能を使って解いてみることにします。なお、 $a^2$ のように係数にべき乗が付いていたり、 $\sin a$  のように係数が初等関数のなかにあるなど非線形の場合には、正規方程式が利用できませんので、こちらの方法を使うこととなります。

## 起動

タスクバーの[スタート]メニューにある[EQUATRAN-G]グループの中の[EQUATRAN-G for Windows]にマウスポインタを持っていき、クリックします。

起動できるとオープニング画面が表示されたのちに、下のようなEQUATRAN-Gのウィンドウが表示されます。



## ソーステキストの作成

問題を解くために数式モデルを入力して、ソーステキストを作成します。新規に作成する場合は[ファイル] - [新規作成]コマンドを、すでにあるファイルを修正するのであれば、[ファイル] - [開く]コマンドを実行します。子ウィンドウとしてエディタが開かれますので、つぎのようにここにソーステキストを入力していきます。

```
ソーステキスト - caicho2.eqs
/* 最小2乗法 */
GLOBAL N = 7
VAR x(0) "温度 [K]" ..
    , y(0) "熱容量 [cal/mol·K]" ..
    , ycal(0) "熱容量の計算値" ..
    , s(0) "残差" ..
x = ( 300, 500, 700, 900, 1100, 1300, 1500 )
y = ( 8.89, 10.68, 11.85, 12.68, 13.26, 13.68, 13.99 )
ycal = a + b*x + c*x^2
s = y - ycal
FIND (a#1, b#1, c#1) LEAST s
OUTPUT a, b, c
OUTPUT1 x, y, ycal
```

/\*と\*/で囲んだものはコメントであり、1行目は特別にタイトルとして扱われます。

3行目のGLOBAL文はNというパラメータの定義をしており、配列変数の配列の大きさを与えています。5～8行目のVAR (VARIABLEの略)文は変数の定義をする文です。配列変数は使用する前に定義する必要があります。一方、スカラー変数はBASICと同じように定義を省略することもできますが、定義しておけば変数の後ろに付けた説明項が表示の際に使われやすくなり、後からこのソーステキストを見たときに分かりやすくなります。

10～11行目は配列変数 x と y に実験データの値を与えています。13～14行目は、(2)、(3)式がそのままの形で記述されているので説明はいらないでしょう。

そして、16行目のFIND文が最適化計算あるいは最小2乗法計算を指示する文で、この場合、LEASTのつぎにある e の2乗和を最小にする a、b、c の値を求めることを指示しています。また、#1 は初期仮定値を意味します。18行目のOUTPUT文は計算結果を出力する変数の指定です。

### 計算実行

さて、この問題を解くために EQUATRAN-G を実行してみましょう。 [実行]メニューを開き、[ラン]コマンド をクリックします。

実行プログラムが起動されて計算が行われ、答えが新しく開かれた計算結果の子ウィンドウに表示されます。



したがって近似式は

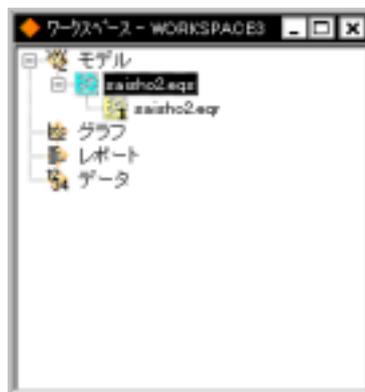
$$y = 6.2343 + 0.010304x - 3.4673 \times 10^{-6} x^2$$

ということになります。

### グラフ作成

つぎに求めた近似式と実験データをプロットして比較してみます。

自動設定の機能によりプロットしたい変数を指定するだけで、とりあえずグラフにすることができます。あとは自分の気に入るように編集することになります。



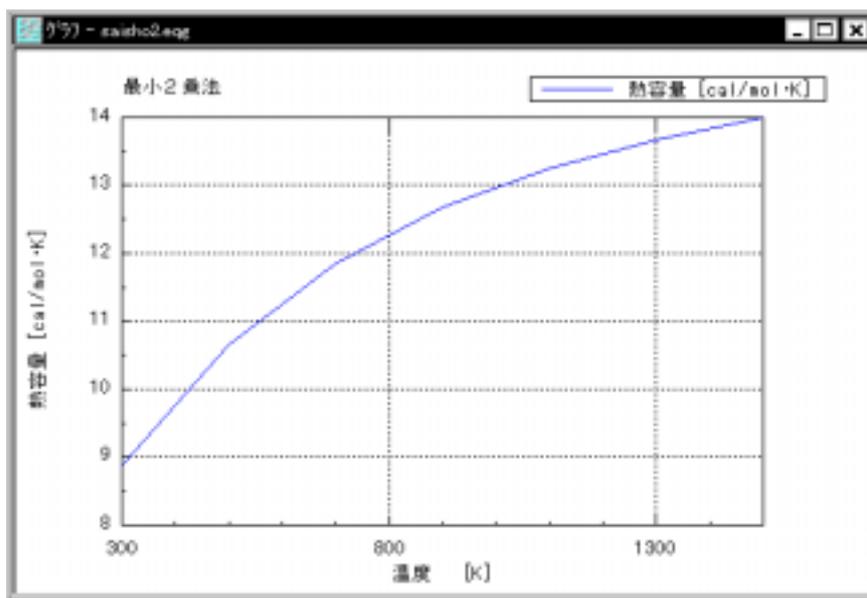
自動設定するために、ワークスペース上でリザルトファイル saisho2.eqr を選択して、[実行] - [グラフ表示]コマンドを実行します。また、ワークスペース上でリザルトファイルをドラッグして [グラフ] テキストラベルにドロップしても同じ操作ができます。

[自動設定]ダイアログボックスが表示されます。[変数名]グループボックスには、自動設定する変数を指定するテキストボックスが3つあり、それぞれ3つの軸に対応しています。とりあえず

ず X 軸は横軸、Y 軸は縦軸、それも Y 1 軸は左側の縦軸、Y 2 軸は右側の縦軸と考えてください。各軸にどの変数を取ってプロットするのかを指定するわけです。今は、X 軸に  $x$  を、Y 1 軸に  $y$  を取ることにします。



<OK> ボタンをクリックすることでグラフウィンドウが開いてグラフが表示されます。

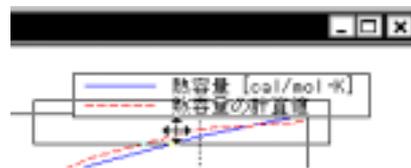


### グラフの編集

これに熱容量の計算値  $ycal$  を追加してみます。[設定] - [変数名...] コマンドを実行します。

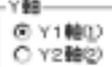


つぎに、凡例を図枠の中に入れ、下の方に移動してみます。凡例の中にマウスポインタを置いて左ボタンをクリックすると、マウスポインタが + になり、凡例のまわりにマークが付きます。これは、今、凡例を選択していることを意味しています。つぎにマウスポインタを凡例の中に置いた状態で、左ボタンを押し込んだままマウスをドラッグし、凡例を図枠の中の右下に移動します。



つぎに測定値yのデータ点は のマークにして線を引かないように変更してみます。凡例の熱容量(上の項目)の文字のところにマウスポインタを置き、左ボタンをダブルクリックします。

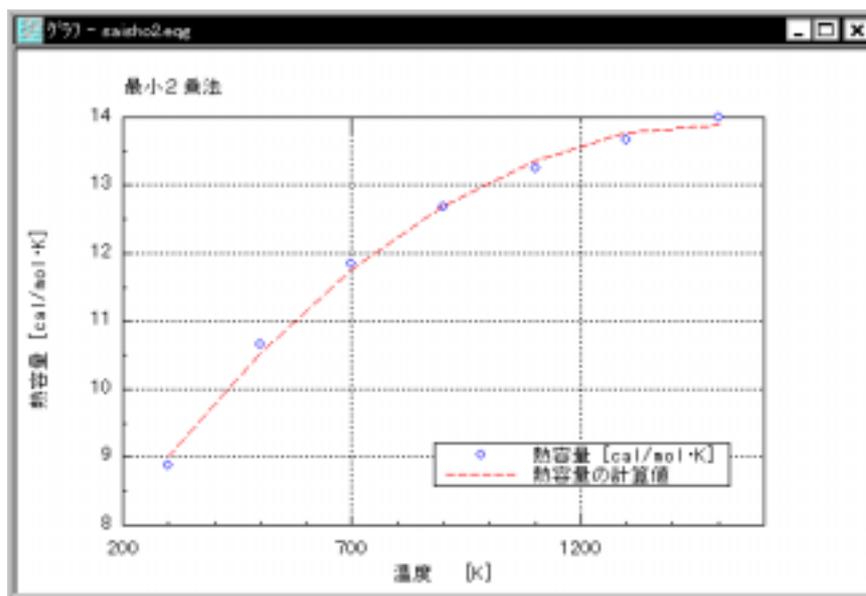


[線の種類]ドロップダウンリストボックスから「線を引かない」を選択し、[マークのタイプ]ドロップダウンリストボックスからは「」を選択します。

最後にX軸の上下限界を変更します。X軸の目盛りの文字の付近にマウスポインタを置いて、左ボタンをダブルクリックします。



上のように下限値を 300 から 200 に、上限値を 1500 から 1600 に変更します。  
以上の操作でグラフが完成しました。



## 4.2 文法 (ソーステキストの記述方法)

文法の主要な項目をリファレンスマニュアル<sup>8)</sup>から抜粋します。

### 式の記述

式は通常の数学的な記法に準じて記述します。

<例>

$$2x + y^3 = \frac{z}{4} \quad 2*x + y^3 = z/4$$

## 配列変数

EQUATRANでは、2次元までの配列を扱うことができます。配列はあらかじめ VARIABLE文によって定義しなければ使えません。また、配列を含む式ではどの要素とどの要素が演算対象となるかに注意しなければなりません。

配列変数同士の演算は、対応するそれぞれの要素の演算を意味します。

<例>

```
VAR a(3),b(3),c(2,3),d(2,3)
```

と定義されている時

a-b は

( a(1)-b(1), a(2)-b(2), a(3)-b(3) )

c\*d は

```
[ c(1,1)*d(1,1)  c(1,2)*d(1,2)  c(1,3)*d(1,3) ]
[ c(2,1)*d(2,1)  c(2,2)*d(2,2)  c(2,3)*d(2,3) ]
```

をそれぞれ意味します。

## 非線形方程式

EQUATRAN には、収束計算が必要な方程式を自動的に判別して収束計算を行う機能（これを自動リセットといいます）がありますが、収束計算の精度や精度計算の対象となる式などを RESET文によって指定すること（これをリセット指定といいます）ができます。

<例>

```
eq: z^2+y=5.2
```

```
RESET z#0 BY eq
```

## 常微分方程式

変数の微分は微分記号アポストロフィ「'」を使って

$$\frac{dy}{dt} \quad y'$$

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

のように書きます。  $y'$  や  $y''$  は  $y$  とは別個の変数となります。独立変数の変数名（この例では  $t$ ）は、別に INTEGRAL文によって指定します。

常微分方程式は上の記法を用いてそのままソーステキスト中に書きます。

<例>

$$\frac{d^2y}{dt^2} + a \frac{dy}{dt} + cy = t + d$$

$$y'' + a*y' + c*y = t + d$$

被積分変数に初期値を与えるためには、次のような # 式を用います。

$y \# 0$

積分計算は INTEGRAL文によって指定します。

<例>

```
INTEGRAL t[0,50] STEP h BY RKV BREAK cond
h=50/300
cond=IF(t>tnext)
```

## 最適化計算

EQUATRANでは、1変数もしくは多変数の最適化問題を解くことができます。FIND文で最適化する独立変数、評価関数を指定して、最適化計算を行います。

<例>

```
FIND(x#1[0,20],y#1[0,20]) ..
      MINIMIZE p UNDER cond UNTIL 0.5%
p=3*x^2+3*y^2-2*x*y-10*x-50*y+330
cond = IF(x<y & x+y<=20)
```

## 入出力

変数に計算実行時に対話形式で入力するか、データテキストから読み込むことで値を与えることを指定するためにはINPUT文を用います。

計算結果を出力する変数を指定するのにOUTPUT文、OUTPUT0文、OUTPUT1文、またはOUTPUT2文を用います。OUTPUT文は画面(または一時ファイル)への出力を指定し、OUTPUT0文はリザルトテキストファイルへの出力を、OUTPUT1 文および OUTPUT2文はそれぞれリザルトファイル 1 およびリザルトファイル 2 への出力を指定します。

<例>

INPUT A, B, C

OUTPUT Xcal, Ycal

OUTPUT1 t, Xcal, Ycal

## < 文献 >

- 1) 化学工学協会編：「化学工学プログラミング演習」、培風館、1976
- 2) 化学工学協会編：「BASIC による化学工学プログラミング」、培風館、1985
- 3) 古崎 新太郎ほか：「マイコンによる化学工学計算」、培風館、1985
- 4) 藤野 清次：「数値計算の基礎」、サイエンス社、1998
- 5) 山本 哲：卒業論文「化学工学における数値計算ソフトの応用および比較」、明治大学工業化学科 古谷研究室、1998
- 6) EQUATRAN-G 例題集 化学工学編、1996
- 7) EQUATRAN-G for Windows Ver.3 ユーザーズガイド [ 入門編 ] 1998
- 8) EQUATRAN-G for Windows Ver.3 リファレンスマニュアル、1998