

# デジタルコンピュータによる飛しょう特性計算

渡 辺 勝・樋 口 一 雄・戸 川 隼 人

## は し が き

観測ロケットの性能が向上し、またレーダ観測の精度もよくなっていくに従って、ロケットのトラジェクトリ計算の基礎式の検討および計算精度の向上が必要なこととなってきた。とくに最近の電子計算機の進歩はめざましく、トラジェクトリ計算にも電子計算機を活用すべき機運は十分に熟するにいたった。このような情勢に応じて、SE 研究班のなかにトラジェクトリ計算のための研究班（略称 TC 班、班長野村助教）が本年初めに組織され、主としてトラジェクトリ計算に各種の計算機を応用することの研究を行ない、あわせて計算式の基礎的検討や観測結果との比較などを実施することになった。その活動の一つとして、航空技術研究所計算研究室の協力を得て、同所の電子計算機データロンを用いて、カップ 8 型、9 型の軌道計算の一部を実施することができたので、ここに報告する次第である。この報告はデジタル型計算機によるトラジェクトリの計算法の解説に主体をおいているが、観測結果などとの比較にもふれておいたから、実験および観測担当の方々もご参考になれば幸せに思う。

## §1. 運動方程式

ロケットの到達高度が 100 km 以上になると、重力の高度による変化が無視できなくなる。高度  $h$ 、地球半径  $=R$  とすると、重力変化の割合は  $2h/R \times 100\%$ 、たとえば 100 km で約 3% となる。また飛しょう距離がのびるにしたがって地球の丸みの影響があらわれてくる。大円に沿って距離  $s$  だけへだたった点では、海面は水平線より  $\Delta h = R(\sec s/R - 1) \approx s^2/2R$  さがる。たとえば 600 km 先では約 30 km もさがっている。

したがってカップ 8 型ないし 9 型のような大型の観測ロケットのトラジェクトリ計算には、上記の影響を考慮にいれて、運動方程式を立てなければならない。

ただしロケットの中心軸はいつも進行方向にむかっているものとして、

重心まわりの運動は以下の議論には省略してある。こうして第 1 図を参照しながらトラジェクトリを求める運動方程式を立てると、

$$\begin{cases} \frac{dV}{dt} = \frac{T-D}{m} - g \sin \theta \\ \frac{d\theta}{dt} = -\frac{g \cos \theta}{V} + \frac{V \cos \theta}{r} \\ \frac{dz}{dt} = V \sin \theta \\ \frac{d\varphi}{dt} = \frac{V \cos \theta}{r} \end{cases} \quad (1)$$

ただし

$$g(r) = g_0 \left( \frac{R_0}{r} \right)^2, \quad r = R_0 + z \quad (2)$$

$g_0$  は海面における重力加速度、 $R_0$  は地球の半径をあらわす。

推力  $T$ 、質量  $m$ 、抵抗断面積  $K$  は推力計画にしたがって、第 2 図に示すような時間  $t$  の函数となる。

$$\begin{aligned} T &= T(t) \\ m &= m(t) \\ D &= K(t) \rho(r) D(V) \end{aligned}$$

推力に対しては、さらに高度による補正項を加える。燃焼の各段を添字  $i$  により区別すると

$$T = T_{oi} + A_{ei}(p - p_0) \quad (4)$$

$A_e$  は噴射ガスのジェット断面積<sup>2)</sup>、 $p$  は気圧をしめす。

$$m = m_{oi} - \frac{T_{oi}}{I_{sp}}(t - t_{oi}) \quad (5)$$

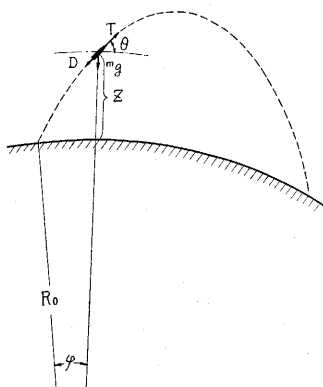
$$D = \frac{1}{2} A_e c_D V^2 = \frac{\pi \rho_0}{8} (d_i^2) \left( \frac{\rho}{\rho_0} \right) (c_D V^2) \quad (6)$$

推力を一定とみなさず、実測した値を用いれば、質量の変化は直線的でなく

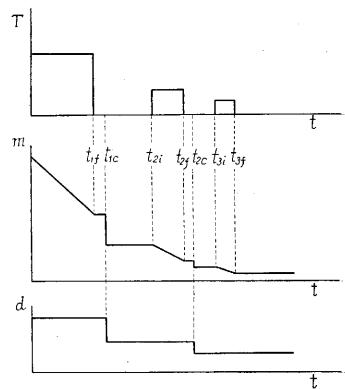
$$\frac{dm}{dt} = -\frac{T}{I_{sp}} \quad (7)$$

## §2. デジタルコンピュータによる解法

デジタルコンピュータによって、上記の運動方程式



第 1 図 ロケットの運動方程式



第 2 図 推力・質量・抵抗断面積の時間関数

を解くには

- (1) 微分方程式を解くルーチン
- (2) 微分方程式の右辺を計算するルーチン

を検討しなければならない。微分方程式の数値解法には周知のように、ルンゲクッタ法と階差法とがあり、それぞれ長所短所がある。われわれの使用した航空技術研究所の Datatron 205<sup>9)</sup> には、そのプログラムライブラリに、ルンゲクッタ法を用いた「Q000」というサブルーチン<sup>9)</sup>が用意されていたのでこれを利用した。このルーチンにはある任意に設定し得る回数<sup>9)</sup>の積分が完了するごとに、精度をチェックして積分間隔を変更するようなサブプログラムが付属しており、間隔の制御が行なえるので便利である。詳細は末尾の付録を参照のこと。

§3. 微係数の計算プログラム

微分方程式の右辺を計算して、各変数の微係数を求めるサブルーチンは、右辺ルーチンまたは微係数ルーチン(Derivative-box)と呼ばれる。ロケットの運動方程式についていえば、右辺ルーチンをプログラムする際に問題になる点が二つあり、以下に論じよう。

	時間 $t$	直径 $d$	推力 $T$	$-T/I_{sp}, m\cdots$
記憶番地とその内容	(600) $\infty$	(610) $d_3$	(620) 0	
	(601) $t_{3f}$	(611) "	(621) $T_3$	.....
	(602) $t_{3i}$	(612) "	(622) 0	
	(603) $t_{2c}$	⋮	⋮	0
	(604) $t_{2f}$	"	⋮	$T_2$
	(605) $t_{2i}$	"	⋮	0
	(606) $t_{1c}$	⋮	⋮	0
	(607) $t_{1f}$	(617) "	(627) $T_1$	.....
	(608) $t_{1i}=0$			

第3図 スラスト・スケジュールの組みこみ

(1) スラスト・スケジュールの組みこみ

燃焼の各段について推力、重量、空気抵抗などが不連続な変化をする。これを処理するため、第3図のようにスケジュールを表の形で記憶しておく。ここで添字 1, 2, 3 は各燃焼段階を、また  $i, f, c$  は点火、燃焼完了、ブースタ切断の各時刻をしめす(時刻の配列が逆になっているのは、プログラム作製上の便宜からである)。たとえば第一段の燃焼中( $t < t_{1f}$ )においては、表の  $t_{1f}$ (607番地)に対応する行、すなわち記憶番地 617, 627, .....にある各変数値  $d, T, \dots$ を用いる。これには Bレジスタを活用すればよい。一方燃焼完了して  $t > t_{1f}$ なることを監視するプログラムがあって、その際は一行上の各変数値に変更する。同時につぎの目標値を  $t_{1c}$ に変える。以下この手順をくり返す。

Bレジスタを使わず、表の一行分を、特定の記憶場所に移して、これを使用するという方法も考えられる。

(2) 任意関数の導入

空気密度や抵抗係数のように、解析的にあらわせない

関数の導入法である。これには適当な多項式ないしは有理関数によって近似する方法もあるが、空気密度はともかく、抵抗係数のように変化のはげしい関数に対してはあまりよい近似は望めない。また推力曲線のように、そのときどき変わるものでは、いちいち近似をやりなおさねばならない不便もある。それでメモリーをぜいたくに使うことになるが、数表の形で記憶させ、これを索引して(いわゆる Table look up であるが、以下にのべるような簡略な方法を用いている)、さらに一次補間を行ない、必要な関数値を求めている。

7000	FAD	7018
7001	SR	0008
7002	SU	7019
7003	STC	7000
7004	SB	7000
7005	1 SL	0000
7006	STC	7000
7007	SB	7000
7008	CAD	7019
7009	SL	0008
7010	STC	7000
7011	1 CAD	0900
7012	1 FSU	0899
7013	FM	7000
7014	1 FAD	0899
7015	ST	6007
7016	CUB	0460
7017		0
7018		0511000000
7019		00000000050

第4図 関数表の索引および一次補間のプログラム

たとえば高度  $z$  がアキュムレータにある場合、空気密度  $\rho$  を求めるプログラムは第4図のようになる。ただし空気密度<sup>9)</sup>は 0~99 km の間の 1 km ごとの値が 900~999 番地に記憶されているものとする。方法は  $z$  の整数部分によって索引すべき番地を求め、小数部分によって一次補間値を計算する仕組みである。

以上の手順をととのえた上で微係数の計算プログラムを作ると、第5図(a)のようになる。

推力に実測曲線を用いる場合は、重量の減少をあらわす §1(7)式をつけ加えればよい。しかしブースタを切り落とす過程はこの式にふくまれないから、別にそのプログラムを加える必要がある(第5図(b))。

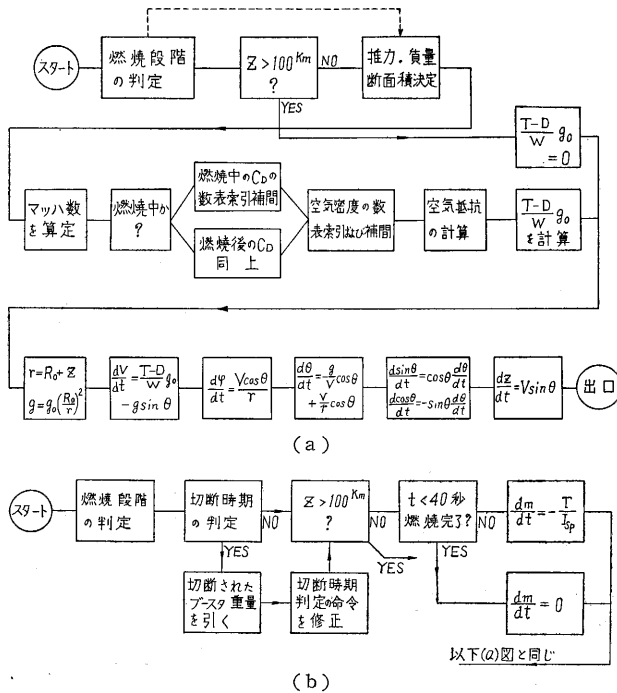
§4. 所要語数および計算時間

以上のプログラムに必要なメモリーの語数は	
ルンゲクッタ.....	360
微係数計算.....	160(+40)
データ.....	350(+400)

ただし括弧内はスラストに実測値を用いた場合の増し分である。

また計算時間は

- (1) 一区間の積分に.....約4秒



第 5 図 微係数計算プログラムの流れ図

(2) 全区間所要時間……………約40分

ただし (2) は、10 区間ごとに印刷、50 区間ごとに精度をチェックして、区間制御を行なった場合の一例である。プリンタにはラインプリンタ (IBM 407) を用いており、一時点のすべての変数を一行にプリントする時間が約 0.4 秒、プリントされたデータ数はすべてで約 50 時点である。

§ 5. 積分区間の制御の問題

デジタル計算機による数値積分法では、時間のとびとびの値に対して計算が進行する。そのためロケットの thrust スケジュールのように、ある時点で不連続がおこる場合、積分点がこれにちょうどまきひっかかることはあまりなく、たいてい積分区間の中に入ってしまう。かりにうまくひっかかったとしても、その際内側から接近した時と、外側に出てゆく時とは異なる値をとるべきであろう。

われわれの用いたプログラムには、精度の観点からする積分区間の制御は行なわれているが(付録参照)、上記の意味での不連続に対する制御は行なわれていない。そ

れでこの点を検討するため、ある計算例について、積分に使用された区間のパターンと thrust スケジュールを対照した図表を作ってみると、第 6 図のようになる。われわれのプログラムでは、不連続点の値として、時間のプラス側の値を用いるようにしたので、これを考慮すると計算の際の実効的な不連続点は、はじめに設定した各秒時に対し、およそ傍記した値程度、早目に行なわれたことに相当する。もっともこの不一致は、計算結果には問題になるほどの影響は与えていないようである。しかしこれを問題にするような精密な計算に対しては、つぎのような対策を講ずればよい。すなわち切りかえ点  $t_i$  に対し、 $t_i - t < h$  内に積分が進んできた時、 $t_i - t$  を  $h$  に代えて積分区間に採用する。その積分が完了したところでパラメータなどを更新し、次の段階の積分を再開するのである。

§ 6. 計算結果の検討

実施した計算はつぎの通りである。

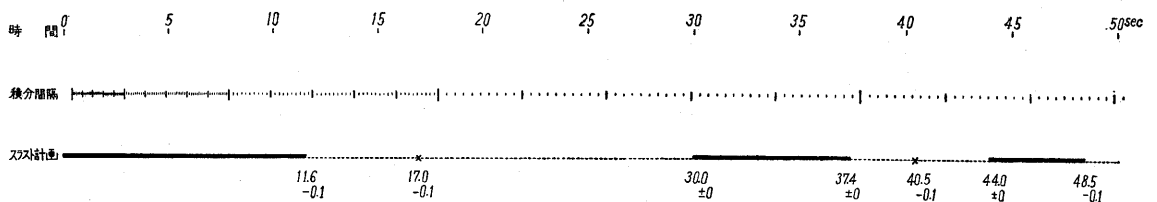
(1) 9 L, 発射角 80°, 78° (第 7 図)

(2) 9 L, 同上 80° で水平線の降下を省略した場合 (第 8 図)

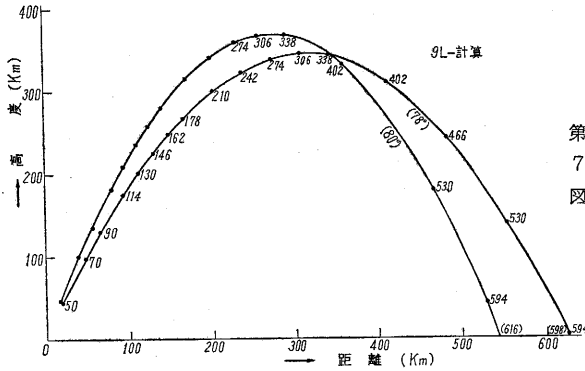
(3) 9 L, Re-entry の部分の精密計算

(4) 8 型, 発射角 80°, thrust 一定の場合および thrust 実測値を用いた場合 (第 9, 10 図)

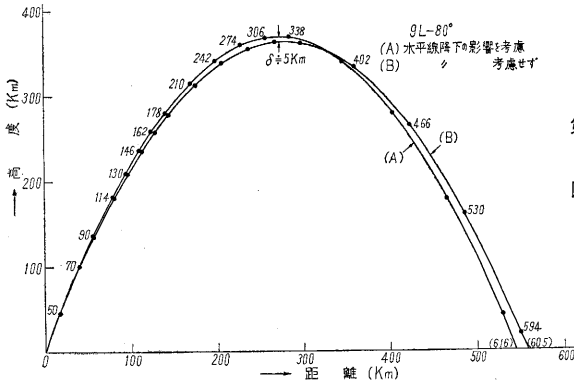
これらの計算結果につき簡単に検討しておこう。この研究のねらいは「デジタル計算機による trajectory 計算法の開発を行なうと共に、従来実施してきた簡易図式計算<sup>9)</sup>の精度を調べてみることをおよびアナログ計算機などで装置の数が限られていて、ある程度の省略を行なわなければならない場合の近似度のよさを確かめておくこと」以上の点にあったので、その観点からのべてみる。(1) まず簡易図式計算法による計算結果と比較して、ランチャー離脱直後の trajectory は、かなり急角度に立っている。たとえば 9 L では 4° 近く立っている。この点は、8 型 3 号機の観測結果 (生産研究ロケット特集号, 12 巻 12 号, 1960 年, 60 ページ) にはほぼ一致した trajectory を示すことから、図式計算法の誤差であるように思われる。一方、速度-時間曲線を比較すると、とくにメインロケットの燃焼の終期で、図式計



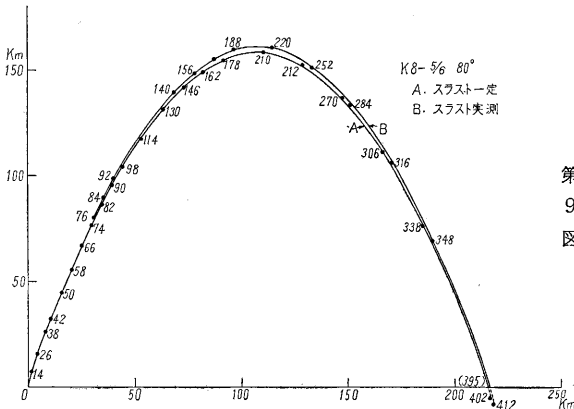
第 6 図 積分区間対照図



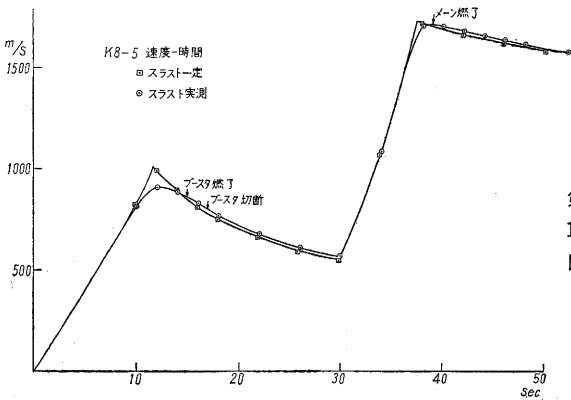
第 7 図



第 8 図



第 9 図



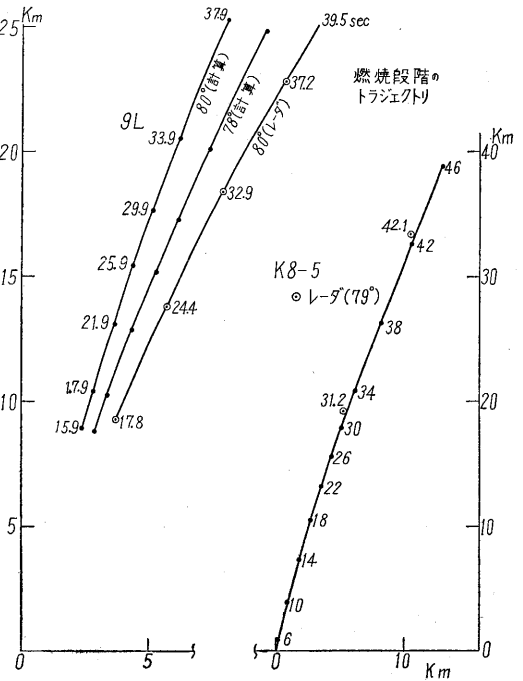
第 10 図

算法よりも速度が低目に出ている。これらの結果より、図式計算法では軌道はねているが、速度が大きくなっており、そのため最高高度などでくらべると、食違いが少ない。(2) 水平線の降下する影響をあらわす §1 (1) 第 2 式の右辺第 2 項を省略した計算では、最高高度で 5 km 程度の違いしかあらわれていない。(3) 落下に際して大気層に突入する、Re-entry のところは計算結果の印刷間隔が大きい。それでこの部分だけを、前に得た解を初期値にとり、間隔を細かにして、やりなおしてみた。その結果多少誤差はあるが、問題にするほどのものではないことがわかった。しかしこの部分は実用上重要であるから、間隔制御の回数をふやすか、印刷回数をふやすか、あるいは計算時間を犠牲にしても積分間隔をこまかくするなどの対策が必要と思う。(4) スラストの値に実測値を用いた方が、一定とした場合よりも、高度がやや高く出る。これは燃焼の終わりごろのスラストの尾を引く部分の効果であろう。しかしこれも予想以上に食違いが少ないことがわかった。

§7. 観測結果との比較

9L: 観測値の方がかなり低くなっており(約 4° 位)、速度も小さく出ている。このことはテレメータやレーダの記録から推定される落下時間(発射点より引いた水平線を通過する時刻)が 510 秒位で、計算値よりもかなり早いことと符合する(第 11 図)。

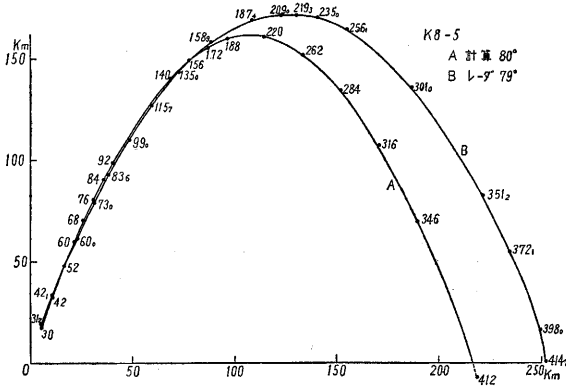
8 型 5 号: 発射角は 79° であるが、80° の場合の計算値と発射直後のトラジェクトリにおいてほぼ一致



第 11 図

第 12 図

し、速度はむしろ上まわっている (第 12 図・第 13 図)。



第 13 図

8 型 6 号: 逆に計算値よりも低く飛んでいる。

以上に関して共通にいえることは、発射点付近の風の影響が相当に大きいことである。

§ 8. 今後注意・改良すべき点

(1) 推力の値は計算結果への影響が大きいので、デジタル計算機を用いて、高精度計算を行なう際には、推力曲線を積分して  $I_{sp}$  の値を確かめておく必要がある。(2) ランチャー上、とくにランチャー離脱時の運動を正確に解析すること。およびその前後の運動方程式の切換えを自動的に行なわせることの検討。(3) 高度によるスラストの補正項はかなり影響が大きいから、十分検討を要する。(4) 切り離されたブースタの運動方程式を解くこと。これは切断時の変数値を次の (5) の方法で求めれば、接続ができる。(5) スラストスケジュールに応じた時間間隔の制御をプログラムに組み込むこと。

本研究の実施にあたってご配慮下さった中西航技研所長、糸川・玉木両教授、推力曲線・抵抗係数・ランチャー長さなどにつき資料を整理提供して下さい森・秋葉両助教授および関係の方々、レーダ観測の結果を提供して下さい浜崎助教授ほかレーダ班の方々、また計算の際のテープの作製・計算機の操作など援助して下さい航技研計算研究室の方々、以上いずれも記して厚く感謝の意を表する次第である。(1961 年 8 月 18 日受理)

文 献

- 1) 渡辺・渡部・名古: “微分解析機のロケット性能計算への応用”, 日本航空学会誌, 7 巻 63 号 (昭和 34 年), pp. 90~95.
- 2) G. P. Sutton: “Rocket Propulsion Elements” (1949), p. 12.
- 3) 航空技術研究所計算研究室: “電子計算機使用法” (1961).
- 4) Burroughs 205 Program Specification, 高千穂交易電子計算機部.
- 5) V. I. Feodosiev and G. B. Siniarev: Introduction to Rocket Technology. (ソ連原本の方が使いやすい)
- 6) 秋葉他: “カップ 4 型などの性能計算”, 生産研究, 10 巻

10 号 (昭和 33 年), pp. 259~262.

7) 日高孝次: 数値積分と数値計算, 岩波書店.

付 録

Datron 205 のための微分方程式ルーチン Q000 (旧称 R221) について<sup>4)</sup>

1. ルンゲクッタ法<sup>7)</sup>の要点 (第 A 1 図)

$$\frac{dy}{dx} = f(x, y)$$

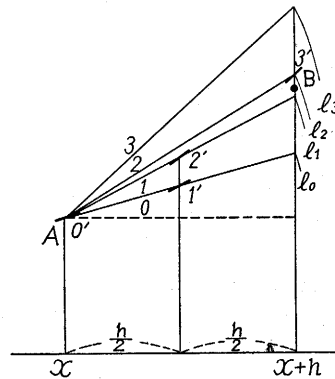
$$\begin{cases} l_0 = hf(x, y) \\ l_1 = hf\left(x + \frac{h}{2}, y + \frac{l_0}{2}\right) \\ l_2 = hf\left(x + \frac{h}{2}, y + \frac{l_1}{2}\right) \\ l_3 = hf(x+h, y+l_2) \end{cases}$$

$$y + \frac{1}{6}(l_0 + 2l_1 + 2l_2 + l_3) \rightarrow y(x+h)$$

誤差:  $O(h^5)$

2. 積分間隔の制御

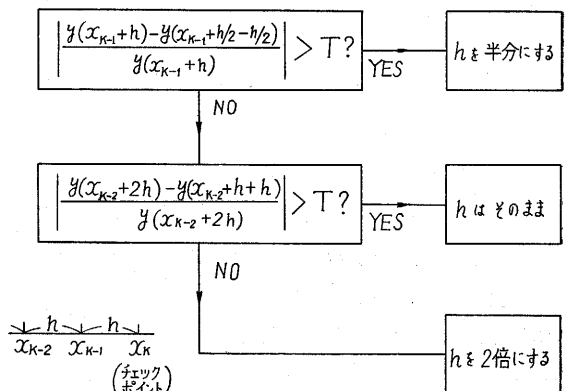
解の精度に応じ て積分間隔を伸縮することにより、能率よく解を求めることができる。すべての変数に共通に、許し得る相対誤差の最大値 (tolerance) =  $T$  をあらかじめ設定し



第 A 1 図 ルンゲクッタ法の図解

ておく。たとえば  $T=10^{-4}$ . 積分間隔  $h$  で  $K$  回積分を進め、独立変数  $x_0, \dots, x_{K-2}, x_{K-1}, x_K$  の各点の値が計算されて  $x_K$  に達したとき、第 A 2 図に示す制御を行なう\*。

\* 積分間隔  $h$  のときの誤差を  $\beta h^5$  とすれば、間隔を  $h/2$  にして 2 回積分して同じ  $x$  の値に到達したときの誤差は  $2\beta(h/2)^5$  とみなし得る。そこでこの両者の差をとれば  $\frac{15}{16}\beta h^5$  となり、これは積分間隔  $h$  のときの誤差にほぼ等しい。



第 A 2 図