

## Runge-Kutta-Merson による常微分方程式の数値的解法

藤 田 長 子

常微分方程式を数値的に解く方法として、最もよく用いられるのは Runge-Kutta 法である。しかし、微分方程式の形が複雑であったり、函数自身に不連続点が存在するような場合には、ふつうの Runge-Kutta 法では解けないことがある。Merson は最近、Runge-Kutta 法を改良し、初めに設定された許容誤差 (tolerance) にもとづいて各点における積分区間の幅を自動的に制御できる方法を提案した。OKITAC-5090 電子計算機のためのこの方法を用いた積分ルチーンを説明し、それを用いて解いた二三の例をあげてみよう。

## 1. Runge-Kutta 法

一般に (1) 式のように微分方程式の形が与えられ、かつ  $x$  のある値  $x_0$  において函数  $y$  の値  $y_0$  がわかっているとき、 $x_0$  から少し離れた点  $x_0+h$  における函数の値はどうなるであろうか。

$$\frac{dy}{dx} = f(x, y), \quad y(x_0) = y_0 \quad (1)$$

すなわち、 $y(x_0+h) = y(x_0) + \delta y$  としたときの  $\delta y$  をみつければよい。求める函数  $y$  を  $x_0$  の近くで Taylor 展開すると次式のようになる。

$$y(x_0+h) = y(x_0) + \sum_{n=1}^{\infty} \frac{1}{n!} y^{(n)}(x_0) h^n \quad (2)$$

一方、

$$\left. \begin{aligned} k_1 &= hf(x_0, y_0) \\ k_2 &= hf(x_0 + \alpha_1 h, y_0 + \beta_1 k_1) \\ k_3 &= hf(x_0 + \alpha_2 h, y_0 + \beta_2 k_1 + \gamma_1 k_2) \\ k_4 &= hf(x_0 + \alpha_3 h, y_0 + \beta_3 k_1 + \gamma_2 k_2 + \delta_1 k_3) \end{aligned} \right\} \quad (3)$$

を定義し、 $\delta y$  がこれらにおもみをつけた和であらわされると仮定する。したがって  $y(x_0+h)$  はつぎのようになる。

$$y(x_0+h) = y(x_0) + \omega_1 k_1 + \omega_2 k_2 + \omega_3 k_3 + \omega_4 k_4 + O(h^5) \quad (4)$$

(3) 式の右辺を 2 変数についての Taylor 級数に展開し、この  $k_j (j=1, 2, 3, 4)$  を (4) 式に代入して、これを (2) 式と比較し、 $h^4$  までの係数を等しいとすると、 $\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3, \gamma_1, \gamma_2, \delta_1$  および 4 個のおもみ  $\omega_j$  が代数的に求められる。その一つの結果はつぎの (5)、(6) 式のようになる。

$$\left. \begin{aligned} k_1 &= hf(x_0, y_0) \\ k_2 &= hf\left(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}\right) \\ k_3 &= hf\left(x_0 + \frac{h}{2}, y_0 + \frac{k_2}{2}\right) \\ k_4 &= hf(x_0 + h, y_0 + k_3) \end{aligned} \right\} \quad (5)$$

$$y(x_0+h) = y(x_0) + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4) + O(h^5) \quad (6)$$

これがいわゆる 4 次の Runge-Kutta の方法である<sup>1)</sup>、

<sup>2),3)</sup> すなわち、微分方程式の形とある点における函数の値がわかっているれば、つぎの点における値は (6) 式を用いて  $h^4$  まで正確に計算することができる。この値を新しい初期値として、同じ過程をくり返すことにより、つぎつぎと積分を進めてゆく。この方法は函数の現在の値のみを必要とし、過去の情報を消してゆくことができるので、電子計算機を用いて計算するのに便利である。しかし、1 点において 4 個の  $k_j$  を求めるので時間がかかり、また 1 個の従属変数につき、少なくとも 4 個の作業番地が必要である。Gill<sup>4)</sup> は (3) 式の 9 個の量の間にある一次従属な関係を仮定し、これらの作業番地の数を 1 従属変数につき 3 個で十分とした。いずれにせよ、(5) 式の右辺の函数形はすべての  $k_j$  にたいして同じであるから、プログラムを作るときは、ただ変数  $x, y$  のみをかえて同一のルチーンにとびこませればよい。このルチーンを「右辺ルチーン」とよぶことにする。

一般に (7) 式であらわされる  $n$  個の連立 1 階微分方程式にたいしても、同じようにして Runge-Kutta 法を拡張することができる。

$$\frac{dy_i}{dx} = f_i(x, y_1, y_2, \dots, y_n), \quad y_i(x_0) = y_{i,0}, \quad (i=1, 2, \dots, n) \quad (7)$$

つぎに Runge-Kutta 法による解の精度についてのべてみよう。一つの方法として<sup>2),5)</sup>、1 点  $x_0$  からつぎの点  $x_0+h$  に移るときの計算誤差が区間幅の 5 乗に比例すると思え、幅  $h$  で一とびにゆく場合と、 $h/2$  として二度でゆく場合の  $x_0+h$  における函数の値をそれぞれ計算してみる。これらの値から求められた相対誤差と、初めに設定された許容誤差との大小を調べて、つぎの積分区間の幅を半分にしたり、そのまま、または 2 倍にする。ただしこの場合には、誤差はだいたいの見当にすぎない。あるいくつかの点を選び、そこにおいてこの種のチェックを行なうことは、「すなおな」函数にたいしては十分な安全弁であろう。まったくチェックを必要としない場合さえある。しかし、このチェックではつかまえないような時はどうしたらよいであろうか。まず、許容誤差を初めに小さく設定し、積分区間の幅を短くすることが

考えられる。注意すべきことは、ここで議論している誤差は 1 点からつぎの点に移るときの計算において生ずるもので、いわば局部的なものであり、大局的なものではない。つまり、慎重だと思われるこまかい歩みを数多くかさね、時間をかけて、たとえ終局地に達することはできても、そこが目的地とかけ離れているかもしれない。一步ゆずってこの点に目をつむっても、つぎの点に絶対に進めないような場合があるのである。たとえば函数の形が複雑であったり、解に不連続点の存在する場合で、このような「たちの悪い」函数にたいしてはここでのべたふつうの Runge-Kutta 法はまったくお手上げである。

2. Merson の方法

簡単のために 1 個の従属変数の場合についてのべるが、 $n$  個の場合に拡張できることは前章と同じである。Merson は (5) および (6) 式を用いたふつうの 4 次の Runge-Kutta 法を改良し、つぎの (8), (9) 式であらわした<sup>3)</sup>。

$$\left. \begin{aligned} k_1 &= \frac{h}{3} f(x_0, y_0) \\ k_2 &= \frac{h}{3} f\left(x_0 + \frac{h}{3}, y_0 + k_1\right) \\ k_3 &= \frac{h}{3} f\left(x_0 + \frac{h}{3}, y_0 + \frac{k_1}{2} + \frac{k_2}{2}\right) \\ k_4 &= \frac{h}{3} f\left(x_0 + \frac{h}{2}, y_0 + \frac{3}{8}k_1 + \frac{9}{8}k_3\right) \\ k_5 &= \frac{h}{3} f\left(x_0 + h, y_0 + \frac{3}{2}k_1 - \frac{9}{2}k_3 + 6k_4\right) \end{aligned} \right\} \quad (8)$$

$$y(x_0+h) = y(x_0) + \frac{1}{2}(k_1 + 4k_4 + k_5) + O(h^5) \quad (9)$$

$\delta y = y(x_0+h) - y(x_0)$  の第 1 近似は  $h \cdot f(x_0, y_0)$  であり、これの  $\delta y$  におけるおもみが  $1/6$  であることは、ふつうの Runge-Kutta の場合と同じであるが、 $k_2$  以下の  $k_j$  (この場合は  $j=5$  まで) の決め方、およびそれらの  $\delta y$  におけるおもみの値がちがっている。

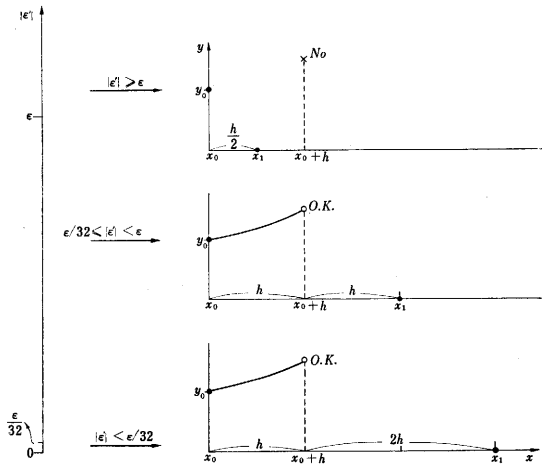
Merson の方法の最大の長所は、各点において計算誤差  $\epsilon'$  が、(8) 式により定義された量を用いて、つぎのように評価できることである。

$$\epsilon' = \frac{1}{5} \left( k_1 - \frac{9}{2}k_3 + 4k_4 - \frac{1}{2}k_5 \right) \quad (10)$$

ただし、 $\epsilon'$  は絶対誤差である。さて、初めに設定された許容誤差を  $\epsilon$  としよう。この  $\epsilon$  と (10) 式によって計算された  $\epsilon'$  との関係が、いま用いた積分区間の幅が適当であったかどうか、またつぎの段階ではどのように幅を選べばよいかという二つの点において重要な役割をもっている。すなわち、

$$a) \quad |\epsilon'| < \epsilon \quad (11)$$

$x_0$  から  $x_0+h$  に進むときの区間の幅は許容誤差の範囲内で適当であり、 $x_0+h$  における函数の値は (8), (9) 式により求められる。また、この値を出発点として、つぎ



第 1 図 Merson の方法の積分区間の幅の制御 ( $x_1$  はつぎの段階の点を示す)

の点に進むときの区間の幅は、つぎのように決められる。

- i)  $|\epsilon'| < \epsilon/32$  区間の幅を 2 倍にできる。
- ii)  $|\epsilon'| \geq \epsilon/32$  同じ幅を用いる。

b)  $|\epsilon'| \geq \epsilon$

いま用いられた区間の幅は適当でなかったため、これを半分にして、もう一度出発点  $x_0$  からやりなおす。すなわち、 $x_0+h/2$  における  $\epsilon'$  を (10) 式によって評価し、それに基づいて (11) 式の条件を判定する。もし (11) 式が成立すれば前述の i) か ii) の判定へと進む。また  $|\epsilon'| \geq \epsilon$  なら、区間をふたたび半分にし、 $x_0+h/4$  の点で同様の操作を行なう。新しい区間の幅に基づいて計算された  $\epsilon'$  が (11) 式を満足するまで、区間はつぎつぎに 2 等分されてゆくのである。

以上の議論は第 1 図によって示される。

3. Merson の方法のプログラム

OKITAC-5090 電子計算機にたいして、ふつうの Runge-Kutta 法<sup>5)</sup>および Runge-Kutta-Gill 法<sup>6)</sup>のプログラムはすでに開発されている。特に前者はロケットの軌道計算の分野において活躍している。しかし、あとでのべるように、この威力のおよばない「たちの悪い」函数に実際直面したので、あたらしく Merson による解法のプログラムをつくることを試みた。このプログラムは、前にのべたように、各点において区間の幅を自動的に制御できる大きな特徴をもっている。実際のプログラムは約 140 語の機械語で書かれた浮動小数点モードの主プログラム形式をとっている。このほかに、函数の個数  $n$  にたいし  $8n+10$  個の番地が、初期条件および途中の計算を格納する場所として必要である。この数の多いのは、一度よみこまれたプログラムは途中でこわされることなく完全に復元されているので、ある計算にたいする初期値が保存されていること、また (8) 式の計算のさいの「右辺ルーチン」へのつなぎにおけるデータと結果のうけ

わたし、そして  $k_j(j=1,2,\dots,5)$  と  $\epsilon'$  を求めるために必要である。Merson 法では各点ごとにチェックしているの、チェック回数の少ないふつうの Runge-Kutta 法よりはいくらか時間がかかる。

実際の計算にあたっては、各問題にたいして (8) 式のための「右辺ルーチン」を閉じたサブルーチン形式で書けばよい。なお、つぎの  $n+5$  個のデータが初期条件として必要である。

- $n$  関数の個数
- $\epsilon$  初めに設定された許容誤差
- $h$  初めの積分区間の幅
- $x_{max}$  積分の終了に対応する独立変数の値
- $x_0$  最初の独立変数の値
- $y_{1,0}$   
 $\vdots$   
 $y_{n,0}$  }  $x_0$  における  $y_i$  の値  
( $i=1,2,\dots,n$ )

このプログラムの詳細についてはここでは省略する。

第2図はフロー・チャートを示している。なお、ALGOL で書かれた Merson の方法が近着の ACM 誌に発表されている<sup>7)</sup>。

4. 二三の応用例

a) 次式によって表わされた電流  $I$  と電圧  $E$  を時間  $t$  の関数として求め、その結果から周期をみつきたいという発振の問題<sup>8)</sup>。

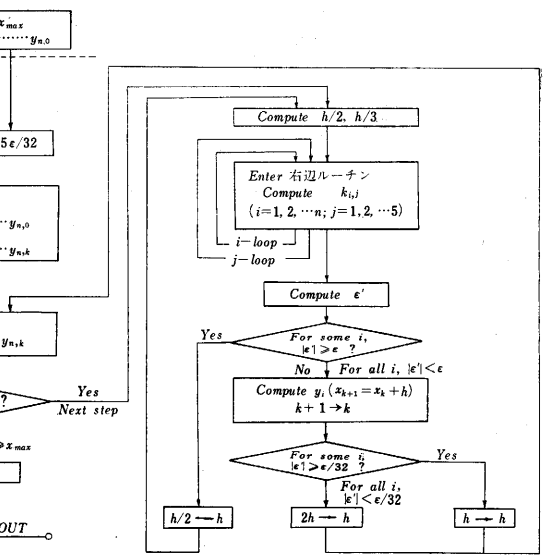
$$\frac{dI}{dt} = \frac{1}{L} (E_0 - E - RI)$$

$$\frac{dE}{dt} = \frac{1}{C} [I - E(\sigma_1 e^{-\alpha Z + \beta} + \sigma_2)]$$

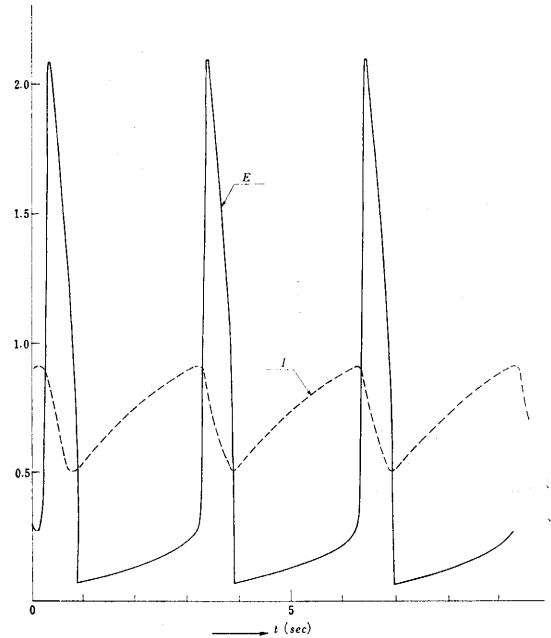
ここで、 $E_0, R, L, \sigma_1, \sigma_2, \alpha, \beta, C$  はそれぞれ定数である。初めにふつうの Runge-Kutta 法による解法をこころみだが、ある特別な組の定数にたいしては電圧を求めることができなかつた。すなわち、電圧  $E$  は非常に鋭い山形の最大値をもつ波形を示したが、最大値直後の急速な下降の途中で、二次的な、さらに鋭い、しかも不規則な高いピークが現われたのである。許容精度を高めたり、チェックの回数を多くしても、今度は  $E$  が不連続的に負の値をとったりしてどうしてもこの困難をのりきることができなかつた。しかし、Merson の方法を用いて解決することができ、その結果は第3図で示されている。

b) 副室付きディーゼル機関の計算の1例として、主・副両燃焼室の圧力経過をクランク角の関数として求める問題<sup>9)</sup>。

副室付き機関の場合に、主・副両燃焼室の圧力経過は熱力学的関係から次式で求められる ( $P_1 > P_2$  のとき)。



第2図 Merson の方法のプログラムのフロー・チャート



第3図

$$\left. \begin{aligned} \frac{dP_2}{d\theta} &= \frac{1}{F(c)} \frac{\kappa}{180\epsilon_2} P_1 \sqrt{RT_1} \cdot F_1(P) \text{ (kg/m}^2\text{/deg.)} \\ \frac{dP_1}{d\theta} &= -\frac{1}{F(c)} \frac{\kappa}{180\{\epsilon_1 + f(\theta)\}} P_1 \sqrt{RT_1} F_1(P) \\ &\quad - \kappa P_1 F(\theta) \text{ (kg/m}^2\text{/deg.)} \end{aligned} \right\} (12)$$

ここで、

- $\theta$  クランク角 (度)
- $R$  ガス定数
- $\kappa$  ガスの比熱比
- $T_1$  主室ガス温度

$P_1, P_2$  主・副両室の圧力

$\epsilon_1, \epsilon_2$  機関固有の定数

$f(\theta)$  主室の体積に関する  $\theta$  の函数

また、 $F_1(P)$  は主・副両燃燒室の圧力の函数であるが、 $F(c)$  が非常に小さい場合には  $P_1$  と  $P_2$  は接近して  $F_1(P) \rightarrow 0$  となり、ふつうの Runge-Kutta 法では計算できなかった。具体的にいうと当然  $P_1 > P_2$  でなければならないにもかかわらず、計算途中で  $P_1 < P_2$  となってしまう、事実と反することになる。この問題もここへのべた Merson の方法のプログラムを用いて解決された。第 4 図はこの方法で計算した結果の 1 例を示す。同じ図の上方にある二つの曲線は、完全ガスの状態方程式  $PV = RGT$  によって計算された主・副両燃燒室のガスの全温度  $T_1, T_2$  を表わしている。ここで必要な各室のガスの重量  $G_1, G_2$  は、これらについての流量式と (12) 式を Merson の方法を用いて連立に解いたものである。

5. さらに検討すべき問題

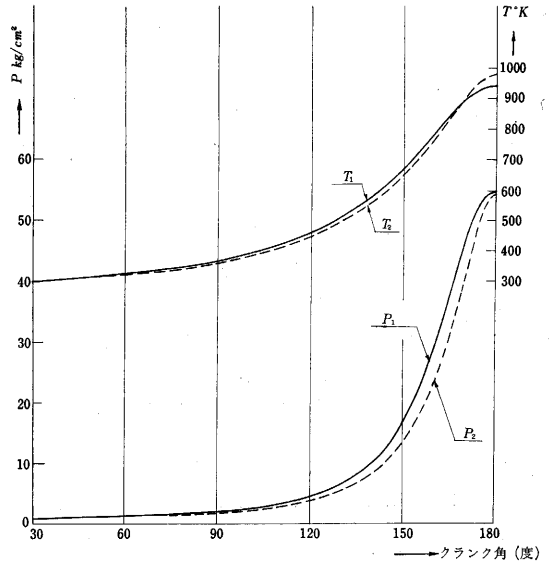
a) 誤差について

Merson の方法は各点における計算の誤差が (10) 式により求められ、これにより、いま用いられた積分区間の幅の妥当性、およびつぎの区間の幅も指示される。ただしこの誤差は、ふつうの Runge-Kutta 法の場合と同じく、あくまで局所的なもので大局的なものではない。函数の値の既知なものの例として  $y_1 = \sin x, y_2 = \cos x$ , すなわち  $\frac{dy_1}{dx} = y_2, \frac{dy_2}{dx} = -y_1$  なる連立微分方程式を許容誤差  $\epsilon = 10^{-6}$  で解いてみた。約 500 回の段階が行なわれたあとで、函数の絶対誤差は約 10 倍大きくなっていった。これはただ一つのテストにすぎず、もっと多くの例につき統計的に調べてみたいと思う。

また、(10)式によってえられた誤差は絶対誤差であることに注意したい。 $n$  個の函数にたいしては、(11)式の内容はすべての函数について満足されなければならない。 $\epsilon$  はもちろんすべてにたいして共通であるから、たとえば、もしある函数一つだけが他の函数に比べて相対的にとてつもない大きさをもつならば、(11)式の判定にはまががかるであろう。初めに、各函数を同じ程度の大きさにしておくことが有効である。

b) 積分の一時停止、または終了について

現在できている Merson 法のプログラムの積分終了の判定の標準形は、第 2 図のフロー・チャートで示されるように  $x \geq x_{max}$  であり、最も原始的な方法である。しかし、独立変数の条件によって積分終了を指示することは、プログラムに少し手を加えただけで比較的簡単にできる。たとえば、積分の途中のある指定された点  $x_h$  で函数の値を計算したいときは、Merson 流に自動的に制御された区間幅のつみかさねで、ちょうど  $x_h$  に達する



第 4 図

ことのできるのはおそらく偶然であろう。このときは、 $x_h$  の値を  $x_{max}$  として、いったん積分を  $x_h$  で止め、函数の値を求めてからふたたび出発すればよい。

一方、いくつかの従属変数間の条件や、または微分方程式にふくまれたデータで、独立変数に対応してある等区間での値が表としてあらかじめ与えられているとき、これらの量の相互の条件などが入ってくるとことはめんどうになる（積分の一つの段階、 $x_0$  から  $x_0+h$  にうつる間に、これらの条件はかわってしまうかもしれない）。こういう条件のもとで、解くべき微分方程式自身を選定しなければならないような場合も、この問題と関連がある。たとえば前述の副室付きエンジンの圧力計算では、従属変数  $P_1, P_2$  の間に  $P_1 > P_2$  なる条件にたいする式のみをあげたが、実際はある点から  $P_1 < P_2$  となり、ここからの微分方程式は、別のものを解かなければならない。徐氏の計算では、その他にも困難な問題にたびたびぶつかった。そのつど解決につとめたが、一般的な形にまとめて検討してみたいと思う。

なおこの項に関して、簡単にまとめたプログラムとしては文献 (10) を参照されたい。

c) 最も有効な方法の撰択について

実際にある微分方程式をときたいとき、ふつうの Runge-Kutta 法か Runge-Kutta-Merson 法かを決めるのにはどうすればよいであろうか。最初に函数のすなおさを透察できればよいが、函数も「見かけによらない」ことがあるので、お話は簡単ではない。いちばん実際的な方法は、まずチェックの点のあいだふつうの Runge-Kutta 法で大勢をみってみる。これでうまくいけばしめたものである。不幸にして思わしくないときは、チェックの点をこまかくとったり、または許容誤差をより小さくとして、

やり直してみる。どうしてもいけなかったら、奥の手、Merson 氏の力をかりよう。こういう過程はめんどろに思われるが、同じ計算を多くの異なったデータを用いて行なう場合、けっきよは時間を節約できる可能性があるので、大切である。たたかなくてもよい石橋は、そのまま前進したほうが賢明であるから。

最後に、この研究について種々の有益な討論をしていただいた渡辺勝教授および、多くの興味ある問題とその結果を提供して下さった山本啓太（もと今岡研究室、現在、理化学研究所）、徐錫洪（平尾研究室）の両氏にたいし、深く感謝する。（1964年1月16日受理）

#### 文 献

1) Runge, C. und König, H. "Numerisches Rechnen"

- (Berlin: Verlag von Julius Springer), p.286 1924  
 2) 日高孝次 数値積分と数値計算 (岩波書店) 1949  
 3) Lance, G.N. "Numerical Methods for High Speed Computers" (London: Iliffe & Sons LTD), p.51, p.56 1960  
 4) Gill, S. Proc. Cambridge Philos. Soc., 47, 96, 1951  
 5) 渡辺勝, 樋口一雄, 戸川隼人 生産研究 13, 347, 1962  
 6) OKITAC-5090 サブルチーン・ライブラリー (沖電気工業KK)  
 7) Lukehart, P. ACM, 6, 737, 1963  
 8) 山本啓太 未発表の論文の一部  
 9) 徐錫洪 "予燃焼室付きディーゼル機関の燃焼に関する研究" (学位論文) 1963  
 10) Lesh, F. JP DEQ (Jet Propulsion Laboratory, California Institute of Technology) 1959,

## 東京大学生産技術研究所報告予告

第 13 卷 第 6 号

川 井 忠 彦 著

### INFLUENCE SURFACES OF ORTHOTROPIC PLATES

「直交異方性板の影響面」(英文)

移動荷重を受ける橋梁、クレーン等の構造物において影響線 (Influence Lines) が非常に重要であることは周知のとおりである。同じようなことが走行荷重を受ける高速道路橋床板についてもいえるのであって、影響線の考え方を二次元の平板の場合にまで合理的に拡張したのがこの平板の影響面 (Influence Surfaces) である。1938年 A. Pucher によってその理論的基礎が確立されて以来、欧米、特にヨーロッパにおいてこの方面の研究が著しい進展を遂げ、数多くのデータチャートが作成されて橋梁床板の安全かつ経済的な設計に欠くべからざるものとなった。この報告は著者が 1954 年から 3 年間米国留学中に直交異方性平板の影響面ダイアグラムを作成するために行なった基礎的研究を主体とし、その後の二三の研究結果を取り入れてまとめたもので、いささか古きに失する嫌がないでもないが、高速道路時代を迎えようとしているわが国の関連分野の方々に、なんらかの参考になれば著者の望外の喜びである。

第 13 卷 第 7 号

丸 安 隆 和・大 島 太 市 著

### SOME APPLICATIONS OF PHOTOGRAMMETRY TO PRECISION MEASUREMENT OF SCULPTURES

「写真測量による文化財の精密測定への応用」(英文)

著者らは写真測量の技術を各種の目的に利用するための研究を行ってきたが、この報告では特に重要文化財の計測図化への応用について紹介した。

測定の対象は鎌倉大仏・大谷磨崖仏であるが前者は修理を行なうに当たって必要な諸要素—各部の重量、重心の位置等の数量を求め後者では大谷石に刻まれた諸体の関係位置、風化破壊による仏像の実態等を刻明に調べるのが目的であった。

この研究では、これらの目的に写真測量を利用する場合に、特に考慮しなければならない事項の究明も行なわれた。