

磁気テープ装置に対する科学計算での プログラムの一方式*

宮野高明**

1. まえがき

計数形電子計算機を用いる科学計算において、主記憶装置の記憶容量で不足するときは必然的に二次記憶装置を用いるが、新しい見地から磁気テープ装置をその目的に利用することを試みた。取り扱う情報を適当にまとめた計算機語の集団に分類し、その各々に意味を持たせてこれを新しく情報取り扱いの単位と考えることにより、計算機語の各々にアドレスを与え得るとい主記憶装置の性質に依存した従来のプログラムの考え方を、より多量の情報を扱うのに都合よく発展させ、また、主記憶装置、二次記憶装置での記憶および相互間の情報転送などの方法を、新しく定めた情報単位に応じて規定することにより、主記憶装置に情報を入れて計算を行ない、もしも主記憶容量で不足ならば二次記憶装置を補助的に用いるという基本的な二次記憶装置の利用法をはなれて、逆に、全ての情報を二次記憶装置に入れ、必要なもののみを主記憶装置に転送して計算を行なうというかたちにプログラムをまとめ、二次記憶装置を用いてスケールの大きな問題を処理するときにプログラム全体にまとまりを与え、そのプログラミングを容易にする一つのプログラム方式をまとめた。なお、この方式を macro-program 方式と仮称し、本文で以後この表現を用いることにする。

そこでまず、二次記憶装置としての磁気テープ装置の考察を行ない、次に macro-program 方式を説明し、特に磁気テープ装置を用いる時の問題点にふれ、最後に磁気テープ装置を用いた macro-program 方式による科学計算の例として、筆者が京都大学 KDC-I 電子計算機で最近に行なった連立1階準線型偏微分方程式の初期値境界値問題の差分法による数値計算のプログラムの一部を述べる。

2. 二次記憶装置としての磁気テープ装置 (MTU)***

一般に二次記憶装置は計算機により制御され自動的に利用される記憶装置ではあるが、情報取り扱いの単位が主記憶装置では語であるのに対して、語の集団を単位とする点が異っている。この相違を除けば、主記憶装置と二次記憶装置のあいだに記憶装置としての根本的な差異はないと考えられる。また、二次記憶装置に多くの情報を貯えねばならぬほどの科学計算では、データ量、プログラム量が多くて、情報を語単位で考えるよりも適当にまとめられた語の集団を単位として取り扱う方が便利なことも多く、この場合、語の集団を単位とした情報転送ははなはだ有効である。

二次記憶装置として MTU が用いられる時、一般に、プログラムの段階で問題が少し複雑になる。磁気テープ (MT) の前進、後退および停止などの動作が、情報転送と同時に、またはそのみを単独にプログラムの段階で処理されるのが普通で、取り扱い問題や使用する計算機により種々の工夫を要し、計算能率を大きく左右するので、MTU を用いる時の重要な問題となる。

さらに MTU では機械的動作部分の故障や MT の劣化などの原因から、記憶された情報が使えなくなる可能性が他の記憶装置におけるよりも多いと考えられるので、とくに長時間を要する計算ではその対策も考慮すべきであろうが、動作の不確実な MTU は論外としてもこの問題はほとんどが MT の寿命、取り扱いかた、および MTU の保守上の問題と考えるので、本文中ではあまりふれないことにする。

MTU を論じる時に必ず直面するのは、MTU の機能の機種による多様性である。主記憶装置と MTU 間の情報転送形式、MT 上での情報記憶形式に機種によりかなりの相違があるために一般的に論ずる場合、内容の具体性に欠けるうらみはあるが、本文で述べる

* A program system for the use of magnetic tape units in scientific computation, by Takaaki Miyano (Training Institute for Engineering Teacher, Kyoto University).

** 京都大学工学部 (現在京都大学工業教員養成所)

*** このように括弧内に略号を示し、以後本文中でそれらを適当に用いる。

方法は使用機種が定まればおのずから具体的な適用方法が定まる種類のものとする。そこで、MTU をしばらく二次記憶装置の範ちゅうにとどめて、まず、本文の骨子をなす macro-program 方式を概論的にまとめ、次に MTU を用いる場合の具体的な問題点について述べる。

3. Macro-program 方式

3.1 情報の単位と種類

普通の情報単位である計算機語のかわりに、適宜にまとめられた語の集団をあらたに情報の単位として、macro-word (mW) と名づける。mW のうちで、計算機本来の命令語よりなる語の集団を macro-order (mOd), また、数値語*よりなる語の集団を macro-number (mNo) と名づけ、原則として mW はこの2種類とする**。mW に含まれる語数でもって mW の大きさをあらわし、それを mW の語長と名づける。一般に、mOd と mNo は語長が同じとは限らないし、それらの語長が一つの機種について常に一定である必要はない。ただ、ある機種で一つの問題を取り扱う限りでは、それらの語長が定まっている方が便利である。mOd は主として機種により語長が定まり、mNo は問題により語長が定まると考えられる。二次記憶装置と主記憶装置間の情報転送が、定まった語数を単位として行なわれる場合と、ある範囲で任意語数の転送が行なえる場合について、もう少し具体的に述べよう。

情報転送が定まった語数を 1 block として行なわれる場合には、その 1 block の情報を 1 mW と定める。本来の命令語の集団としてまとめるのに適当な語数を、1 mW の語数の整数倍にえらび、それを mOd の単位と定める。mNo については 1 mW の大きさを単位とし、これを越える数値語の集団は mW 単位に分割し、その分割の数 k により、1 mW を単位とした k 倍長の mNo と名づける。

1 block の語数が可変である場合は、mNo も mOd も、1 block の語数として許される限度内におさまるものは、上記のように 1 mW の語長を定めることなく適当に大きさを定めうる。しかし、許された語数限度内で mNo がおさまらなければ、その数値語群を分割して前述の場合と同様に mNo の数倍長表示をしない

なければならない。mOd に関しても同様のことがいえる。

mNo が具体的に意味する原像を持つ時、その原像のボタンを示す数値群、すなわち、それにより mNo と原像が対応づけられるようなものを、その mNo の list と名づける***。list を構成する数値群がまた mNo を作る場合もあろうし、その量が少ない場合は他の扱い方が考えられる****。

3.2 記憶装置の分割

計算に必要な全ての情報は mW として二次記憶装置に記憶されているのを原則とし、必要に応じて主記憶装置に読み出されることになる。すなわち、二次記憶装置が従来の主記憶装置の役目を受け持ち、あらためてこれを macro-memory (mM) と名づける。mM での mW の記憶形式は使用する機種、および mNo, mOd の定めかたにより定まる。

主記憶装置は見掛け上、mW を扱う演算装置、補助記憶装置****、およびこの方式でまとめられたプログラムの制御装置として用いられる。そのために主記憶装置を適当な大きさのメモリ群に分割し、その各各を macro-register (mR) と名づける。1 mW の語長が定まっている時は、それを置くことのできるメモリ群を mR の単位としてさらに次の分類に従うものとする。mW が、mNo, mOd の形で語長の定まる時は、その各々について次の分類に従い mR の大きさを定める。

1 mOd を置く mR を macro-order register (mOdR) と名づける。mOd は mM より mOdR に転送されて、そこで mOd に含まれる本来の命令語群を普通の意味で実行することになる。

演算を行なう時に数 mNo またはその一部を置く mR を特に定めた時は、それを macro-operation register (mOpR) と名づける。

mW の転送、mOd についての fetch cycle などプログラムの進行制御に必要なルーチン類、これを macro-program control program (mCP) と名づける、を置く mR を定めて macro-program control register (mCR) と名づける。mCP に含まれるルーチン類は本方式でまとめられたプログラムの進行を特

*** mNo に対する list は、実数を浮動小数点表示した時の指数部のような意味をもつと考えればよい。

**** たとえば、 n 次正方形行列での list は $\langle n \rangle$ という 1 正整数だけである。

***** mM に対する小容量の補助記憶装置、また mW の各種転送時の buffer register にもなる。

* 文字語も含める。

** これら以外に、本方式に必要なルーチン類であって、その転送にあたっては mW 的に取扱われるが、それ以外では本来の命令語の集りとしてしか意味のない情報がある。

長づけるもので、もう少し具体的な構成については後でふれる。

これら以外の mR は macro-temporary register (mTR) と名づけ、その働きを限定しないでおく。また mTR の大きさは任意に定めてよく、1 mW の語長が定まっている時は、それを置ける大きさの mR を単位として分割し、その各々を mTR とすることができし、また、1 mOd, 1 mNo を置ける大きさの mR に分割してもよい。

3.3 macro-program 方式でのプログラム進行過程

本方式はあくまでもプログラムの段階で考慮されるもので、計算機本来の内部機能をなんら変更するものではないから、演算進行の各時点において見るならば計算機本来のプログラムの進行が行なわれているに過ぎないが、プログラム全体から見た時には次の四つの過程に分類される。

1) initial process (IP): 計算に必要な全ての情報を、mM に mW 単位で記憶する過程で、mCR の内容となる mCP を入力装置より読み込み、次にその mCP の働きにより mR を介して情報を mM に移す。

2) loading process (LP): mCP を含めて、全ての情報が mM にある時、mR を計算開始に必要な状態に準備する過程で、mCP を mM より mCR に転送し、その働きにより、最初に実行すべき mOd を mOdR に転送し、その mOd で処理されるデータを mTR または mOpR に転送する。

3) production process (PP): LP の終わったのちに、mOdR の mOd を実行に移し、次の順序で計算を続行する過程である。すなわち、mOdR に置かれた命令語群の最初の命令に jump したのは、普通の意味でのプログラムの進行を続ける。ただし、mOd はその語長の範囲内で閉じたプログラムで、その mOd を終了した時は mCP の定まったルーチンに次の mOd を指定してプログラムの制御を移し、mCP は指定された mOd を mOdR に移して再びその最初の命令語に jump する。mOd の実行中に、必要に応じて mM と mTR 間で mW の転送を行なったり、入出力装置を介して mR と外部記憶との間で情報転送を行なうが、この時も mOd は一時プログラムの制御を mCP のしかるべきルーチンに移し、これらの転送に必要な手続または転送そのものがすめば、再び mOd によってプログラムは進められる。

4) stopping process (SP): PP を中断して、mR にある情報のうちで次に計算を再開続行するのに必要なものを mM に移して計算機本体 (CPU) をあけわたす過程で、SP が必要となる例としては、長時間を必要とする計算で計算機使用時間が分割される時、他の計算を優先的に処理する必要から CPU を途中であけねばならぬ時、計算途中の出力結果から、計算機外での判断を必要とする時などが考えられる。SP を実行するのは PP の任意の時とすることは無理であろう。問題と機種によりその方法と時期は考えねばならない。

これら四つの過程は適当に重複させることができる。たとえば、IP を実行する時に、入力装置から読み進む mW の順序、buffer として用いる mR を適当にえらべば、mM に情報を書き込み終わった時には同時に LP を完了させることも可能である。また、SP で CPU をあけわたしたといっても、直ちに計算続行できる状態、すなわち次の LP が完了した状態にすることもできる。

3.4 mCP の内容*

mCR におかれるルーチン類は本方式を特長づける最も大切な部分で、その構成が計算能率と深く関係し、一番問題を多く含むところであって、一般に次のようなものを必要とする。

1) 読み込みルーチン: 入力装置から情報を受け取り mM に移す mW の読み込みルーチンで、本来の計算機の読み込みプログラムを含み、次の書き込みルーチンを利用して、mR を介し転送が行なわれ、mR は入力装置と mM の buffer の役をつとめる。それで、〈使用する入力装置〉、〈buffer として用いる mR〉、〈mM 上の書き込み場所〉、〈読み込むべき mW の数〉を指定して実行させる。

2) 書き込みルーチン: mR から mM への mW の書き込みルーチンで、〈mW のある mR〉、〈mM 上の書き込み場所〉、〈書き込むべき mW の数〉を指定して実行させる。

3) 読み出しルーチン: mM から mR への mW の読み出しルーチンで、〈mM 上の mW の場所〉、〈mW を移すべき mR〉、〈読み出すべき mW の数〉を指定して実行させる。

4) mOd に対する fetch cycle 用ルーチン: 〈次

* mM と mCR 間の転送、mM 上での記憶にあたっては mW の取り扱われるが、mOd, mNo の特長を備えず、普通の utilityprogram の集合と考えればよい。

に実行すべき mOd を指定して実行させる。すなわち、指定された mOd を $mOdR$ に移してその最初の命令語に $jump$ する。

5) SP を実行するためのルーチン：書き込みルーチンを利用して実行するが、中止した時の状態を記録し、次に LP を実行する時の資料をととのえる。

6) LP を実行するためのルーチン：読み出しルーチンを利用して実行するが、前回に中止した状態を認識し、計算続行に必要な処置をとる。

7) 出力ルーチン：計算機外へ情報を送り出すルーチンで、計算の途中結果を知るためや、計算の進行状態を監視するのに必要な印字やさん孔を行なうもので、問題により出力ルーチンはさらに幾種類かにわけて考えた方が便利であろう。

mCP のさらに詳細な構成については、機種と問題により最終的に定められるのは当然であるが、一つの機種については、問題により定めねばならないいくつかの要素を含むとしても、多くの問題に普遍的な mCP の構成を定めることができると考える。

4. MTU を mM として用いる時の問題点

mM として MTU を用いる時、macro-program 方式でまとめられたプログラムの能率は mM と mR 間の情報転送形式に大きく左右され、それは使用する計算機によって定まっているから、残る問題は与えられた機能の範囲でいかに mW の転送を行なうかになるが、これらに関する事項をあげてみると、

(i) MT が動き出してから求める情報位置*にたどりつくまでの待ち時間(呼出し時間)。

(ii) 情報位置に一番近い所から、情報転送に要する時間(読み出し、または書き込み時間)。

がまずあげられよう。

前者では、 MT 上の情報位置探索が CPU との同時動作で行なえれば、呼出し時間を見掛け上短縮できるが、そうでない時は MT 上の情報位置配列にのみ工夫の余地が残される。そして、呼出すべき情報位置が MT 上で順を追って並ぶことは多くの場合むずかしいと考えられる。

後者では、 MT の起動、停止および正味の読み出し、書き込み時間などの組み合わせられたものが問題になる。 mM と mR 間、または、buffer storage を持つ計算機で mM と buffer storage 間の情報転送が

CPU の演算実行と並列に行なえる時はプログラムの能率を良くしうる。

一般に、計算機のこれらに関する機能が単純なものほど mM についての読み出し、書き込みをプログラムの上で工夫する余地が少なく、したがって、 mCP の構成は簡単になるがプログラムの能率を良くすることは難しくなる。

一方、 mM よりの読み出しについては先き廻り制御、 mM への書き込みについては、あと廻し制御が可能であれば能率をあげる代りに mCP の構成が複雑になるのは当然で、この場合、さらに次のことを考慮しなければならない。

(iii) 先き廻り、あと廻し制御による情報転送に要する時間と、それに並行して行なわれる CPU の演算時間との関係。

できうれば、それらの情報転送が、並行して行なわれる CPU の演算時間内に完了することが望ましく、そのためには、先き廻り時期などの判断を mCP で行なわねばならず、 mOd の語長が適当に長いことが必要であり、 MT 上の情報位置の配列にも工夫を要する。

CPU と直結された MTU が複数個あることは、磁気ドラムで記憶装置がいくつかの track に分けられ、その各々に磁気ヘッドがあることにより、平均待ち時間が短くてすんでいることと本質的には同じ効果をもたらすが、複数個の MTU を適当に利用して計算能率をあげるためには、さらに複雑な mCP の構成が必要となる。

mM として MTU を用いる時のもう一つの問題は、長時間計算における途中での情報確保である。計算途中で MT の情報が正しく読み出せない、逆に、正しく書き込めないなどの事態が生じた時に、それまでの計算をむだにしないための対策を必要とすることがある。そのためには、取り扱う情報のうちで全計算を通じて変らない部分と、変化する部分にわけて、後者を適当な形で途中確保するなどの処置が考えられる。

5. Macro-program 方式による科学計算の例

大阪湾気象高潮の解析的推算のために導かれた基礎方程式の吟味のため、京都大学 KDC-I 計算機により、この計算機としては長大な計算を行なった。そのうちで湾内気象潮の計算のプログラムを macro-pro-

* 現に情報の記憶されている場所または情報を書き込むために予定されている場所。

gram 方式によりまとめた例として簡単に説明する。したがって、提起された問題の全体、プログラムの詳細および計算結果などについてはここではふれない。

5.1 基礎方程式とその差分方程式および計算機に課せられた問題

高潮現象解析のために理論的に考えられる最も詳しい運動方程式が、計算可能な範囲まで簡略化されて最後のにまとめられた基礎方程式は次のとおりである。

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -9.8 \frac{\partial}{\partial x} (0.0102 p + \zeta) + (4.05 \times 10^{-6} \cdot q_x - 0.0018 \cdot u) / h \quad (1-1)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -9.8 \frac{\partial}{\partial y} (0.0102 p + \zeta) + (4.05 \times 10^{-6} \cdot q_y - 0.0018 \cdot v) / h \quad (1-2)$$

$$\frac{\partial \zeta}{\partial t} + \frac{\partial}{\partial x} (h \cdot u) + \frac{\partial}{\partial y} (h \cdot v) = 0 \quad (1-3)$$

ただし、静止水面上に直交座標 $0-xy$ をとり、 $u, v: x, y$ 方向の流れの速度成分でともに深さについての平均をとったもの [m/sec]

ζ : 水面上昇量 [m]

p : 大気圧 [milibar]

q_x, q_y : 風速自乗の x, y 方向成分 [m²/sec²]

h : 静止時の水深 [m]

t : 時間 [sec]

この3式を差分方程式に直すために、時間変数 t は解の安定条件その他を考慮のうえ、 $\Delta t = 40$ sec の前進差分を、空間変数 x, y は $\Delta S = 2$ km の中央差分を用い、表現の都合上 $\Delta S/2$ を Δl であらわす。使用する mesh と、 u, v, ζ の計算位置は第1図のとおりとし、水深 h は u, v の計算位置に与え各々 h_u, h_v とあらわす。(1-1, 2, 3) 式を差分方程式に直すと次のようになる。

$$u_{i+1,t} = (PQX_i + \zeta_u) + 0.01 \times \{ (h_u' + u_i^{d-1}) \cdot u_i + u_i^{d-2} \cdot V_i \} \quad (2-1)$$

$$v_{i+1,t} = (PQY_i + \zeta_v) + 0.01 \times \{ (h_v' + v_i^{d-2}) \cdot v_i + v_i^{d-1} \cdot U_i \} \quad (2-2)$$

$$\zeta_{i+1,t} = \zeta_i - 0.02 \times (h_u \cdot u)_i + h_v \cdot v)_i \quad (2-3)$$

ただし、

$$u_{i+1,t} = (u|x, y; t + \Delta t), \quad u_i = (u|x, y; t)$$

$$v_{i+1,t}, v_i, \zeta_{i+1,t}, \zeta_i \text{ も同様}$$

$$\zeta_u = 0.196 \{ (\zeta|x, y - \Delta l; t) - (\zeta|x, y + \Delta l; t) \},$$

$$\zeta_v = 0.196 \{ (\zeta|x, y - \Delta l; t) - (\zeta|x, y + \Delta l; t) \}$$

$$h_u' = (100 - 7.2/h_u(x, y));$$

$$h_u = 0 \text{ なら } h_u' \text{ に無意味な定数を与える.}$$

$$u_i^{d-1} = (u|x - \Delta S, y; t) - (u|x + \Delta S, y; t),$$

$$u_i^{d-2} = (u|x, y - \Delta S; t) - (u|x, y + \Delta S; t)$$

$$V_i = (1/4) \cdot \{ (v|x - \Delta l, y - \Delta l; t) + (v|x - \Delta l, y + \Delta l; t) + (v|x + \Delta l, y - \Delta l; t) + (v|x + \Delta l, y + \Delta l; t) \}$$

$$h_v', v_i^{d-1}, v_i^{d-2}, U_i \text{ は各々 } h_u', u_i^{d-1}, u_i^{d-2}, V_i \text{ の } u \text{ を } v \text{ に書き直した式である.}$$

$PQX_i = 0.19992 \times 10^{-3} \{ (p|x - \Delta l, y; t) - (p|x + \Delta l, y; t) \} + 0.162 \times 10^{-3} \{ (q_x|x, y; t) / h_u(x, y) \}$

$$PQY_i = 0.19992 \times 10^{-3} \{ (p|x, y - \Delta l; t) - (p|x, y + \Delta l; t) \} + 0.162 \times 10^{-3} \{ (q_y|x, y; t) / h_v(x, y) \}$$

$$h_u \cdot u)_i = h_{(x+\Delta l, y)} \cdot (u|x + \Delta l, y; t + \Delta t) - h_{(x-\Delta l, y)} \cdot (u|x - \Delta l, y; t + \Delta t)$$

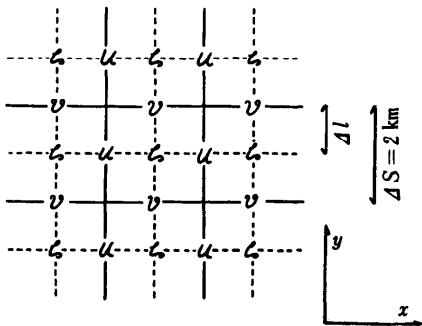
$$h_v \cdot v)_i = h_{(x, y+\Delta l)} \cdot (v|x, y + \Delta l; t + \Delta t) - h_{(x, y-\Delta l)} \cdot (v|x, y - \Delta l; t + \Delta t)$$

大阪湾を第2図の mesh で近似し、1950年 Jane 台風からえた気象データ (p, q_x, q_y) を用い、(2-1, 2, 3) 式から湾内気象潮を計算するために次の条件を与えた。

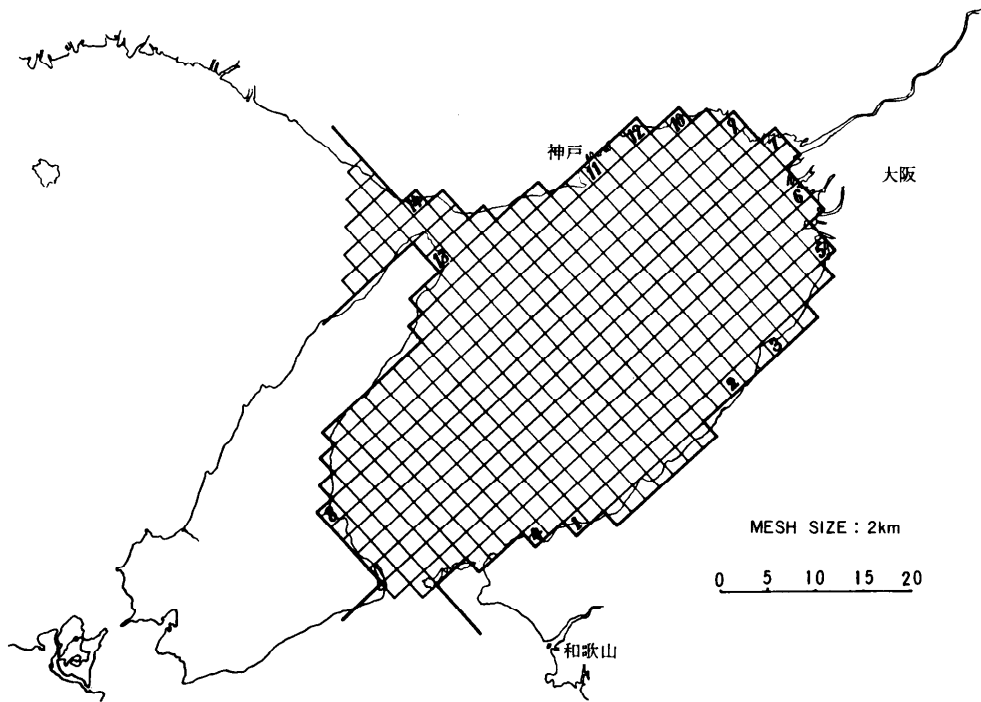
初期条件: $t=0$ で $u=v=0, \zeta=0$

境界条件: 明石海峡の十分外部、友が島水道で常に $\zeta=0$ 。岸上の u, v はその境界に垂直な方向の値は常に 0。それと一対をなす v, u は必要がないので問題にしない。これに伴い、境界線またはそれに近い所で u_i^{d-1} 以下の各式の右辺のどれか1項が不在または不定の時は適当な近似式を用い、それにも該当しない特別な点ではその近傍の値を用いるなどの処置をとる。

大阪湾水深分布は計算の収束性を考慮して実測値に若干の修正を加えた。また、気象データは本来ならば (x, y, t) を変数とするデータだが、計算の都合上実時間7時間のデータを、適当にえらばれた10個の時



第1図



第 2 図

間点で取り換える 10 組の定数 $p_i(x, y)$, $q_{xi}(x, y)$, $q_{yi}(x, y)$, ($i=1, 2, \dots, 10$) として与え, (2-1, 2) 式の PQX_i , PQY_i の代りにその定数を用いて計算した PQX_i , PQY_i , ($i=1, 2, \dots, 10$) を用いた.

計算は実時間 7 時間分について行ない, その間観測すべき主なデータは第 2 図の番号位置での実時間 200 秒ごとの水位変化とで, なおこのほかに, truncation error を切りならすために実時間 20 分ごとに u, v の平滑化の計算を行なっている. 以上が湾内気象潮の問題のあらましである.

5.2 情報の単位および記憶装置の分割

KDC-I は 4,200 語 (200 語の高速記憶を含む) の容量の磁気ドラム (DM) を主記憶装置とし, 約 7 万語の容量の MTU を二次記憶装置に備え, これと DM との間に 50 語の core buffer (CB) を有する記憶装置の構成をもつが, CB は DM と MT 間の buffer としてのみでなく, DM に続く通し番号によりメモリの延長にも使える. normal band (NB), quick access band (QB), CB 間および CB と MT 間で 50 語を 1 block とする情報の集団転送が行なわれ, MT 上での記憶形式は 50 語を単位とする block 情報書き込

みと同時に 4 桁以内の任意整数を block number (BN) として付けうる浮動ブロック番号方式がとられている.

KDC-I の MTU は magnetic tape control unit (TCU) と magnetic tape handler (MTH) 2 台よりなり, TCU が MT 関係の命令を受け取ると CPU より独立して操作を続け, その間に CPU は別の命令を実行できる機能を有するが, この機能を十分利用するには CB を主記憶の延長に用いることをあきらめねばならず, CB を高速記憶として利用できないための不利益を免れない.

そこで検討の結果 CB を主として高速記憶として用いることにした. これと mCP の構成との関係はのちのべる.

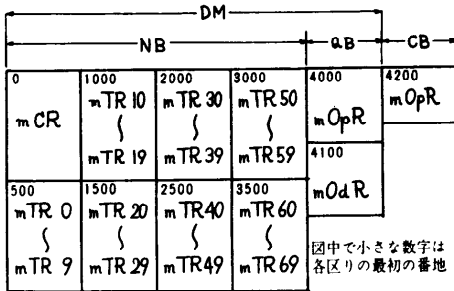
これらのことから, 1 mW の語長を 1 block 分の情報と定め, DM と CB は mW の語長単位の mR に分割され, MT は mW 単位での記憶場所, すなわち mM となる.

取り扱う問題も考慮に入れて 1 mOd は 2 mW とし, QB の半分を mOdR に定め, 残りの QB と CB を mOpR と定めた. したがって mOpR は 3 mW

の容量をもつ。

1 mNo の語長を 1 mW の語長に等しく定め、二次元的に分布する数値類は、 u : 434 個、 v : 446 個、 ζ : 411 個およびこれと同じ分布を持つものを各々 9 倍長の mNo と考えた。これら以外の数値語*もそれぞれ mW 単位にまとめた。

mCP は約 10 mW の大きさになるので、10 個の mR を mCR にあて、のこりの mR は mTR- i ($i=0, 1, \dots, 69$) とした。mR の分割を第 3 図で平面的に示す。



第 3 図

5.3 mCP の構成およびプログラムの進行

先にもふれたように KDC-I は、ある程度の先き廻り、あと廻し制御による MT と CB 間の情報転送を行なえる機能を有するが、CB を高速記憶として利用する方に重点を置いたので、第 4 節の (iii) については何も考慮せず (i), (ii) の時間が計算時間に直接ひびく方法をとった。

幸にしてこの計算では mM 上の mW の配列がほとんど最適配列に近くできたので、第 4 節の (i) の時間はかなり短くなり、CPU が持つむだ時間の大部分は第 4 節の (ii) の時間となった。

各 step** の終わりでは、mM 上の最初の mW を読み出す準備をする呼出しだけの先き廻り制御として MT の巻戻しを行なったので、mM として endless tape を用いたのと同じ状態になった。

さらに問題の性質から、mTR の利用に工夫をして、普通の計算進行中には mM への書き込みがなく、読み出しだけで計算が進行するようになされた。また、2 台の MTH のうちで 1 台だけを用いた。

* たとえば、list 類やプログラムの進行状態を示すに必要な数値類。

** $t=t$ の時の u, v, ζ を用いて $t=t+dt$ のそれらを計算する一連の計算を、 $t=t$ にはじまる 1 step とよぶ。

したがって、mCP の内容はかなり単純なものになり、次の 10 種類、全てで 10 mW 以内のルーチン類になった。その各々に S*d という記号をつける。

- S*1 mOd に対する fetch cycle 用ルーチン。
- S*2 MT を起動してすぐ次からの mW を mR に移す読み出しルーチン。
- S*3 MT を起動して BN を頼りに前進方向に mW を求め、mR に移す読み出しルーチン。
- S*4 MT を起動しそこへ mW を書き込むルーチン。
- S*5 さん孔紙テープより mM への読み込みルーチン。
- S*6 SP 実行用ルーチン。
- S*7 LP 実行用ルーチン。
- S*8 観測データとその時の step 数を印字する出力ルーチン。
- S*9 u, v, ζ の全分布と計算進行判断の資料を印字さん孔する出力ルーチン。
- S*10 上記以外の印字ルーチン。主として message print を行なう出力ルーチン。

macro-program 方式でのプログラムの進行は mCP により制御されるので mCR に mCP が置かれていることを前提とし、IP, LP の実行は最初に mCR をセットすることよりはじまる***。

IP では、次に S*5 を利用して mW の読み込みを行なうが、mM での記憶順序は次の方針で行なった。各 mOd はその実行の順に記憶される。各 mOd は、それを mOdR に移して実行する時に必要とする mNo のうちでその時すでに mR にあるものを除き、mM 上ですぐあとに必要な mNo を従えて記憶する。そして、その mOd 実行中に必要に応じて S*2 で mTR に読み出すものとする。

LP では、S*7 により次のことが行なわれる。各々 9 mNo よりなる u, v, ζ を mTR 10~18, 30~38, 50~58 に、これらの mNo に必要な list 類を mTR 64~69 に、計算進行状況を記録するデータ類を mTR 60~61 に、最初に実行する mOd を mOdR に転送する。

SP は、必要に応じて各 step の終わりに実行できる****。console にある jump switch を押しておくとその step 終了とともにプログラムの制御は S*6 に

*** そのために、さん孔された紙テープを光電式読取機より読み込ませる手続を必要とする。

**** 各 step は平均 5 分 28 秒で計算された。

移り、その時までには計算済の u, v, t および LP で mTR 60~61 に読み出したデータ類を mM の定められた場所に転送することにより SP は完了する。

PP で, $S*1$ により mOdR に転送され実行に移される mOd の各々はプログラム全体の粗い flow chart と深い関係を持つ。この例では $(h_u' + u_t^{d-1}) \cdot u_t$ の計算が 1 mOd にまとまっており、各 step はこの計算よりはじまる。その他 z_u, z_v, v_t^{d-2} なども 1 mOd で計算される。このように各 mOd は計算の各段階、すなわち粗い flow chart の各ボックスにかなりよく対応するのでプログラム全体が理解しやすく、debugging が容易になる。また各 mOd は閉じたプログラムとして $S*1$ で接続されていくから、プログラム全体の検討は 1 mOd の範囲での検討の積み重ねとして行なえた。同様の理由で、新しい mOd の付加、割り込み、すでにある mOd の順序の変更も容易である。

この問題では、9倍長の mNo が最も大きな mNo で、その計算にあたっては前もって続いた9個の mTR に移し、計算のために mW 単位で mOpR に移して処理したが、step ごとに変化して行く mNo は mTR に置いたままにできるように工夫したので、PP 中の mM からの mNo の読み出しは定数に限られ、mM への書き込みを必要とせず mCP の構成を簡単にできた。

6. あとがき

ここでは MTU を用いる場合を主として述べたが、macro-program 方式を他の二次記憶装置に用いるとさらに有効であろう。それは、第4節でふれた種類の問題をかなり避けうると考えるからである。徹視的には本来のプログラムの進行が行なわれているので、それに規制を加えることの必然的な結果として、絶対的

な計算時間はかえっていくらか長くなり、プログラムの段階での工夫は策におぼれる危険性もあるが、二次記憶装置を使わねばならないスケールの大きな問題のプログラミングの方針を与え、プログラムの構成を理解し易くする本方式は、計算機に課せられる問題がいつもその主記憶容量の範囲内で処理できるものとは限らず、むしろ問題のスケールがそれを越える傾向を示すことが多く、その意味から計算機の利用度の拡張という点で有効な一つの考え方と思う。

紙面の都合上、第5節の例は簡単に過ぎるが、くわしくは最後の各文献を参照されたい。KDC-I の MTU はわが国最初の試作品であり、本学に設置以来長時間の本格的な利用は未だ試みられたことがなかったので、いろいろと問題があったが、現在では十分実用に供しうるものになった。

筆者の MTU の使用に関して御協力下さった計算機室関係の諸氏に謝意を表す。特に本文全般にわたり御意見、御助言をたまわった清野教授、および矢島氏に、また本文第5節の問題に関して御校閲いただいた山田教授に深く感謝する。

参考文献

- 1) 山田彦児, 宮野高明, 外諸氏: 大阪湾気象高潮の解析的推算について。(印刷準備中)
- 2) 宮野高明: 大阪湾気象高潮の計算—磁気テープ装置の応用例—; KDC-I レポート (MT-002, SP-001)
- 3) KDC-I の命令語; KDC-I マニュアル第1巻, KDC-I のプログラミング; KDC-I マニュアル第2巻
前田, 坂井, 矢島, 古谷, 太田, 万代: KDC-I の磁気テープ記憶装置; 1961年電気通信学会電子計算機研究専門委員会資料
(昭和38年1月28日受付)