

データベース指向UNIX環境の構築の 基礎概念とその応用

島山正行*

Database Oriented UNIX Operating Environment and Its Applications

by

Masayuki HATAKEYAMA

ABSTRACT

In the present work, we propose a work station system which is equipped with a large capacity hard disk. The work station operates under the UNIX operating environment. On this work station system, a general purpose database oriented system for both scientific studies and engineering purposes is constituted. A flexible and simple format is designed for the data file. A software bus concept is proposed to provide a common path for data transmissions. On the basis of this software bus concept, the database system is developed using the modular programming technique. The DBMS (Database Management System) now developed is a relational type DBMS, and is also based on the software bus concept. As for the graphic display system, it is developed by combining the native commands of the intelligent graphic terminal and siggraph-CORE system. It is confirmed that the database system now developed is useful for analyzing unknown phenomena by applying this database system to a fluid dynamical problem.

1. 序 論

最近の超大型コンピュータ又はスーパー・コンピュータの発達は見覚ましく、複雑な3次元流れの解析、航空機の空気力学的・構造力学的解析などの従来困難であった分野にも次々その利用範囲を広げつつある。更にはこのような実用につながる分野だけではなく、生体高分子の立体構造や分子軌道の精密計算、その他核融合分野等々の基礎科学の分野への導入も目立っている。時がたつにつれますますその適用分野を広げ成果を挙げるであろうことは衆目の一致するところであろう。

しかしながら、現在のスーパー・コンピュータの利用の仕方に対して、もう一工夫できる余地があるのではないかと考えられる。現在のスーパー・コンピュータ(以後超大型コンピュータも含めてSCと略記する)の計算の主対象は桁違いの高速性の故に可能となった実用的な計算問題である。勿論それはそれで科学として工学として重要な価値を持つことは疑いない。しかし、実用計算の場合は正確な数値が出てしまえば、ある意味ではそれが最終結果であって、SCからのデータを改めて詳細な解析の対象とする、などという事は少ないと考えられる。ここで今後のSCの利用という点について考えると、次のような計算対象がSCの次の段階の利用法の1つとして考えられる。即ち、長大な計算量の故に従来

* 東京都立航空工業高等専門学校

は困難だった計算対象のうちで、対象たる現象について計算は可能だけれども、どのようなデータをどれだけ求めれば対象たる現象が把握できるのかについて全く（ほとんど）不明であるような現象を取扱うこと、である。このような未解明の現象の解析においては、try and errorがつきものであり、同一の数値例を手を変え品を変えくり返し計算させ、又は多少アプローチの異なるデータを得るため多数の数値例を再計算という場合も予想される。このような莫大な計算を思うままにSCを用いて実行できる人は極めて少数であろうし、又、使い方としても非能率的である。又、SCはこのような解析を行なおうとするには全く適していない。それは次の理由である。

1. 多くの場合SCはバッチ処理マシンである。たとえinteractive processingが可能であるとしても現在の小型コンピュータの使い勝手の良さ比べると格段に操作性が低い。
2. ファイル（データ・セット）に関する不便さ。共用のSCではユーザー1人当りのディスク割当が狭い。多少大き目の数値計算ならば数値例1個で一杯になってしまう。更にはデータ・セットの定義が煩雑であり、スペースの柔軟性もない。又、数値データ・セットをeditorで見ながら編集するなどという事もできない。
3. データ解析を豊富な周辺機器を駆使して縦横に行なう事が難しい。周辺機器は計算センターに行かなければ使えないし、細かなデータ解析、データ加工に適したツールも存在せず、OSやユーティリティのそういった機能も低い。

要するにSCに期待できるのは超高速の計算のみと言えよう。けだし、このような欠点は実験やむを得ないものである。そのような用途に適するには全然設計されていないし、今後もそれは期待薄であろう。又、SCはそれでよいと考えられる。

とすると、上記の欠点として挙げた項目はすべてその必要に応じ、SCのユーザ自身が解決すべき事になる。つまり、大型のハード・ディスクを備え、使い勝手のよいOSとツールを備えたソフトウェア体

系を持ち、SCとの通信手段を含め豊富な周辺機器を備えたパーソナル・ユースのコンピュータを使えばよいわけである。そのような事は、実はここ2～3年でやっと可能になって来た。いわゆるワーク・ステーションといわれるコンピュータ・システムである。代表例はSVN-2, APPOLO社のDOMAINであろう。OSはいずれもUNIX^{(1),(2)}である。このような汎用のワーク・ステーションは特別に大量の数値解析データを収容・管理・処理するシステムなりツールなりを持ち合せているわけではないから、ユーザがそれらを作り上げることになる。このデータ・ベースを指向したシステムを作り上げるための基礎概念の構成と、実際の（プロトタイプの）システムを作ることが本研究の目的である。そのため、まず次節においてワーク・ステーション、データ・ベース、数値解析システムの構成を議論する。

2. データ・ベース指向ワーク・ステーション・システム

ワーク・ステーション（以下ではWSと略記する）というコンピュータ・システムの分類が出現したのはここ数年の事である。性能的には16ビット・パソコンとスーパーミニコンの間のレベルであり、大型コンピュータやSCとの通信機能を持ち、使い勝手のよいハード・ソフト上のツール（道具）、ハードではビット・マップ・ディスプレイ、マウス、ソフトではマルチ・ウィンドウなどの他ソフトウェア開発用の多くのツールを標準で持つ、などを特徴とする。OSはその大半がUNIXである。我々はこのWSの使い勝手の良さ、性能、OSたるUNIXの特徴、特に入出力におけるre-direction機構、ファイルシステムの簡潔さやソフトウェア開発ツールの充実ぶりに着目し、従来は用いられることのほとんどなかった大量データ解析用システムを、このようなWS上に作ることを試みる。

まずWSシステムが備えるべき最小限のハード・ソフト上の構成を述べると、

1. UNIX環境。できるなら多重仮想記憶機能を持つ4BSD版UNIX。従ってCPUは16ビット乃至32ビットのもの（68000, 68010, 68020等）が必須である。

2. 大容量ハード・ディスク。正味のユーザ領域が数十MB以上。できれば100MB以上。
3. カラー・グラフィック・ディスプレイ。できればハードコピー・プリンターも。
4. SCとのネットワーク装置。

となろう。ディスク容量は上記の程度ではたちまち一杯になるであろうが、パーソナル・ユースを前提とする以上やむを得ないであろう。1~2年後にはGBクラスの光磁気ディスクが使えるようになれば、この点の不便は解消されよう。図1にその概観を示す。

このような基本構成の上にSCにかけるソフトウェア(数値解析プログラム)を開発し、SCからの大量のデータを収容・管理・解析するシステムを構築するわけであるが、その前に重要な前提を述べておきたい。それはこのWSのユーザの問題である。ユーザは本WSを使って仕事をこなすわけであるから当然何らかの分野の専門家であろう。多分、科学か工学か又はメーカーなどの解析・設計家であろう事を想定している。そのユーザがUNIXについてのエキスパートであったり、FORTRAN77以外の言語に通暁していたり、WSのハードに通じていたり、又はデータベース・マネジメント・システム(DBMS)のimplementationができたりする、などはまず想定し得ない。流体力学分野の専門家の平均知識はFORTRAN77, SCとその端末機の使用技術と言ってよいであろう。しかし本研究における前提としては更に、UNIXの平均的レベルの操作方法とWSのシ

ステム管理が一応できる事を前提とする。従って、将来的には本システムが科学工学分野におけるデータ解析用データベースを指向する限り、上記の前提にあるユーザが充分駆使できるような簡潔なシステムを構築せねばならない。つまりソフトウェアはほとんどすべてFORTRANで書かれねばならないことを意味する。

次に後の第4節に述べる事情からシステムの構築をモジュラー・プログラミング方式とする。この手法は簡単に言えば複雑な処理を多数の単機能の小さなツールによって行なうようにシステム構築するというものであり、ツール間のデータ転送が適切かつ高速に行なわれるならばシステム構築・改良・再構築が非常に行ない易くなる。それは上記の想定ユーザに充分可能でなければならない。本システムの特徴もそこにあるべきだと考えている。つまりWS上のデータベース指向の解析システム自体の改良などは各ユーザが簡単に行なえることもシステム設計上の要件となる。

最後にデータベース・システム(以下でDBシステムと略記する)について述べる。現在、ビジネス用、文献検食用などのDBマネジメント・システム(DBMS)はかなり普及している。又UNIX用としてもINGRES, UNIFY, INFORMIX等何種類かが存在する。しかし、本システムではこれらを導入してその上にシステムを作るという方針をとらない事とした。それはまず第一に、(次節で述べるように)、想定されるデータ構造の多様さ、非一様性、

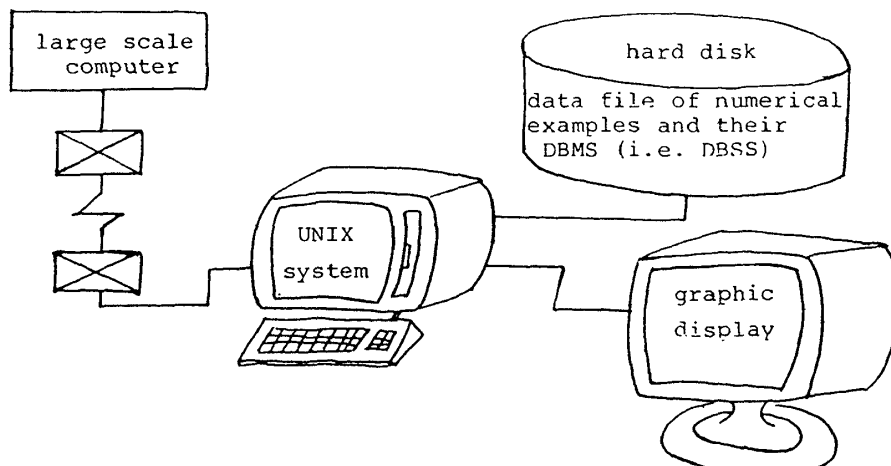


図1 データベース指向ワーク・ステーション

改変・追加・削除の多さ、その他データアクセス要求の多様さに充分対応できそうな DBMS が見当たらない事による。次に市販の DBMS はそれはそれで完結したシステムであり、本システムへの組込みとするには適切でない事が多く、又ユーザ自身が DBMS 自体を修正しようにも技術的にまず不可能であることが挙げられる。第3に現在開発中の DB モジュールを分類すると DBMS というよりデータ解析ツールとしての要求が高く、DBMS はむしろ簡単なものでも当面は充分ではないかと考えられるからである。システム開発が相当進んだ段階で、もし仕様があれば導入してもよからうと考える。

以下の節においては本 DB 指向システムの基礎として、データ構造、データ・ファイルの設計、基礎概念としてのソフトウェア・バスの定義、及び DBMS、グラフィックシステムの構成などを述べ、最後に現用のプロトタイプ・システムについて述べることにする。

3. データ構造の設計

本システムにおいては、その性格からして、初めに計算結果のデータが既に出ているとし、そこから出発するのが自然である。通常に行なわれているプロセス（処理プログラム）中心の設計ではなく、データ中心のシステム設計の方針を採る。まずどのようなデータ・フォーマットを想定するかであるが、形式から見ると数値と文字である。数値は対象現象の数値データや各種の情報、文字は対象現象の知識やコメントなどである。ここでまず設計への条件として、

1. 同一数値例データ・ファイル中には数値と文字の両形式のデータが混在し得ること。

が挙げられる。更には、対象現象を把握するのに必要なデータ形式やデータ量及び相関関係は現在不明であるとの前提がある。従って、

2. 数値データ・フォーマット及び量その他についての多様性を許容すること。それでもデータ・ファイルの読み書きはエラーなしに行なえること。

勿論、最小限のデータ・フォーマットを規定しなければデータ・ファイルのリード/ライト（R/Wと

略す）エラーを起こす。この解決はデータ・ファイル内部に自己のデータに関する情報をすべて内蔵し、その指示に従ってデータ・ファイルを R/W させるように仕組めばよい。self-contained structure を持つデータ・ファイルである事を意味する。次によく言われているが、

3. すべてのプログラムに共用できるデータ構造を持たせること。これはデータ・ファイルのソフトウェアからの独立性を持たせる事になる。
4. データ・ファイルの構造は簡単であり、データの追加・訂正・削除等が簡単であり、しかもそれらに伴うプログラムの修正などは不要なこと。

UNIXではデータ・ファイルをエディターで見たり編集したりが自由自在である。従って直接エディターでデータ・ファイルを加工する時には第4項が必要である。

5. データ・ファイルの R/W 及びデータ処理の効率の悪さ、即ちマシン効率の悪さ、は無視する。まず取扱い易さ、データ構造の柔軟性を優先。

以上のような条件を満たす汎用のデータ・ベース・システムはまれであるか存在しないであろうと推論される。

上記の条件を満たすべく決められたのが図2に示すデータ・ファイルの構造図である。本データ・ファイルの最小単位はブロックである。1つのブロックは冒頭と末尾に initiator と terminator を示す実数値 $+0.2e+37$ と $-0.2e+37$ が置かれ、この2つの実数値の間が1つのブロックと見なされる。次の行はブロックのタイトルで80文字以内の英数字で表現される。第3行目と第4行目は当ブロックのデータ・フォーマット、種類、性質、量を表わす。第3行目は行数、第4行目は列数を表わす。行数はプラスの整数が普通であるがこれがマイナス値の時は行数不明を表わすとする、列数も通常は正の整数であるが、これがゼロの時は次に続くデータが英数字から成るコメント（知識）ブロックであることを意味する、と約束する。次に続くのがデータ（数値データ、

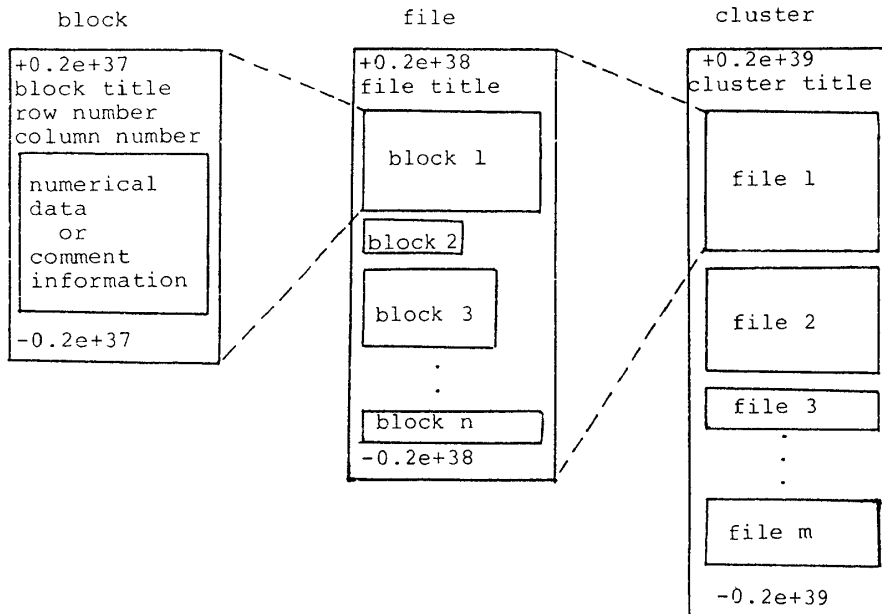


図 2 データ・ファイルの構造

知識データ)である。ブロックは terminator が出現すると終了したと見なされる。上記のブロックを 1 個以上直列に並べ冒頭と末尾に initiator と terminator を示す $+0.2e+38$ と $-0.2e+38$ を、及びファイル・タイトル (80文字以内) を付け加えるとファイルという単位となる。このファイルを 1 個以上直列に並べ、冒頭と末尾に initiator と terminator を示す $+0.2e+39$ と $-0.2e+39$ を、及びクラスター・タイトルを付け加えることによりクラスターと呼ばれる単位となる。3つの階層の initiator と terminator については、その絶対値 V_b , V_f , V_c を

ブロック・クラス: $0.1e+37 \leq V_b \leq 0.9e+37$

ファイル・クラス: $0.1e+38 \leq V_f \leq 0.9e+38$

クラスター・クラス: $0.1e+39 \leq V_c \leq 0.35e+39$

なる範囲に定義すればよいことにしてあるので、ユーザ各個がその定義範囲内を数多く区分することにより各単位のフラッグとして又は亜階層として用いる事が可能である。

現在はブロック、ファイル、クラスターの3階層を持っているが、より上のスーパー・ストラクチャ、より下位のサブ・ストラクチャの定義も容易であるが現在は特に使われていない。3階層データ・ファイル構造における利用の仕方については、例えば、ブロックは速度分布データ、境界層厚さなどを表わ

す単種のデータ、又は1まとまりの知識を述べるコメントなどの収容を、ファイルはブロックの集合体として1組の流れのパラメータに対応する単一の数値例データ全体の収容を、クラスターはある条件を満たすファイルの集合体の収容を、それぞれ想定して定義されている。このデータ・ファイルの R/W には一般には専用のサブルーチンが用いられる。サブルーチンでデータが読まれると、データ(数値、知識)は自身に関する情報、タイトル、initiator の値などともに配列 (common) に収められ、使用に供される。データ・ファイルの構造と使用法は以上であるが、これで本節の初めに挙げた5個の条件をすべて満たしており、当初の設計目的は達成されている。

4. ソフトウェア・バス

まず基礎概念としてのソフトウェア・バスについて述べる。バスとはもともとはハードウェア関連の用語であり、各ハードウェア・モジュール (CPU、メモリー、ハード・ディスク、端末機、グラフィック・ディスプレイ、ネット・ワーク・デバイス) 間のデータ伝送をすべて行なう共通路の事である。各ハードウェア・モジュールのバスとの接続は専用コントローラーを介して実施される。マルチ・バス、

Qバス, VMEバスなどがよく知られている。バスのメリットは構成すべきハードウェアをコントローラーを介してバスに抜き・差しするのみで機能する, ということにある。このような機能をソフトウェア・システムの世界で実現するのを目的に考えられたのがソフトウェア・バスである。かなり制限された形式ながらソフトウェア・バスの役目を果たしているのは配列 (common) である。しかし, 通常のプログラムにおける使い方ではとても汎用とは言えないし, プログラム・モジュール (サブルーチン) の抜き差しもエラーなしでは難しい。これも含めてソフトウェア・バスの役目を果たし得るものを挙げると,

- (1) 配列 (common)
- (2) 中間ファイル
- (3) UNIX ツール (pipe, I/O redirection)
- (4) プロセス間通信機能 (4.2 BSD 専用)

が挙げられる。(2)の中間ファイルとは要するに共通の I/O 用ファイルを定義し, プログラム・モジュール間のデータ伝送をすべてこのファイルを通じて行なうことである。機能としては完全であるがいかんせん能率が悪い。UNIX の System V ではこの機能を OS でサポートしている。シェアド・ファイルと呼びメモリー上に展開されているのでアクセス速度が極めて速い。(3)のUNIXツールは高速で伝送できるがデータの送受モジュールがあらかじめ決まっています変更がきかない欠点がある。(4)のプロセス間通信機能は最も高緩な方法で, モジュール間の動的なデータ伝送を可能にするが, 現在試作中でありまだ実現されていない。以上の如く概念として確立し, 使われてはいるけれども欠点が目立つ手法が多い。本システムにおいてもプロセス間通信法を用いて実現を計っており, 各ツールもそれを前提として作成されつつある。

これまでの説明のように, (プロセス間通信機能を除いて), ソフトウェア・バスは特別な概念ではなく, 従来から用いられて来たいくつかのデータ伝送の方法のうち, ソフト的にデータ伝送共通路 (common path) として使えるものを集めただけである。

しかし, 本システムでソフトウェア・バスを強調

するのは重要な理由がある。それは本データベース・システムを完全なモジュラー・プログラミングで以って作り上げる方針だからである。この手法で作られたシステムは小さなモジュール (又はツール) を取り替えるだけで自在に機能を変更することができ, 従ってシステム構築上, 又, ユーザー自身によるシステムの改変を容易にするという方針上, 重要なのであるが, この手法はシステムが汎用でかつ open system (基本システムにない機能をユーザーが付加したり, 変更できるようなシステム) を狙う限りにおいて, ソフトウェア・バスの実現なしには構築が困難になってしまうという事情がある。つまりソフトウェア・バスが実在してこそ, 単にソフトウェア・モジュールの抜き・差しだけでプログラム・システムがエラー・フリーで正常に機能できるわけであるからである。現在, モジュール・プログラミングとかソフトウェアの部品化・再利用が言われているけれども, 言われるほどに発展していないのはこのソフトウェア・バスが充分でないからではないかと考える。つまり global software bus が存在せず local software bus だから互換性の効く範囲が狭いであろう。早急な解決が望まれる。

5. DBMS 及びデータ解析ツール

前にも述べたように DBMS は完成システムとしてではなく単にツールの集合体に過ぎない。図3にツールの主なものを示す。他に UNIX のコマンドやツールも DB 処理ツールとしてよく用いられるし, 勿論コマンド・ファイルもいくつかある。図中のツールも完成品として fix されているものは必ずしも多くなく, 後述の実際のテスト・ケースで試用しつつ機能を充実させて行く。これらのツールは一応部分的にはリレーショナル DBMS の機能と (少なくとも見かけ上は) 同じ機能を持つように作られた (エミュレーション)。ツールはすべて FORTRAN 77 を用いて作られている。

本 DBMS 中のツールは常にシステムの中で使わねばならぬという制約もなく, 又, データ・ファイルもその構成上から即座に了解できるように, すべてのプログラムから直接アクセス及び使用が可能である。従って, このツールがデータ・ファイルとともに

newfil	make a new data file
mkfile	make a data into the data file
inform	count and display the data file
addblk	add a new block into a data file
purblk	purge a new block from the data file
dataR0	
dataR1	read data into bus array
dataR2	
dataW0	
dataW1	write data into the data file
dataW2	
shdata	make a table from a data file
chkdat	check the data format
srblk0	serch necessary block by key words
srblk1	serch necessary block by manual work
pfil0	serch necessary files by keywords
srfill	serch necessary files by manual work
differ	show difference between two files
diffmt	show difference among files
sort0	
sort1	sort the data blocks
sort2	
sort3	
common	select the common attribute
addset	add new attribute to the present data
projec	pick up necessary attributes with main key
select	select the necessary taple
chgkey	change the main key
jonrow	join necessary taples in the data file
joncol	join necessary attribute in the data file
merge0	merge new data
pickup	pickup necessary blocks
elimin	eliminate the data blocks
chgdt0	
chgdt1	change the contents of data blocks
chgdt2	
cutour	cut high values of the main key
cutodr	cut lower values of the main key
DDDmak	make basic DD/D automatically
DDDshw	show basic DD/D of a file
updata	updata the informations and DD/D

図3 データベース用ツール(コマンド)

に別のプログラムシステムの中で使われる、という場合もある。データ解析ツールは現在開発途上にある。特殊な内挿を行なう SPLINE, 行列の形をしたデータに対して各種の代数計算を行なう ARITH, 種々の平均をとる MEAN, etc. である。以上の如く DBMS は未だ完成とは言えない。これを補っているのは UNIX のエディターである。ed, ex, ked, vi, sed と各種あり、必要に応じて使いわける。UNIX のファイルはすべてエディターで編集できるので、DBMS の機能はすべてエディターで代替が効く。少量のデータ加工で、DBMS では煩雑な処理ならばエディターで行なうと実に簡単に出来る事が多。最も有用なツールであると言えよう。

6. グラフィック表示システム

現用のグラフィック・ディスプレイ装置は NEXAS 5500 といい、通常のグラフィックの外にカラー・カメラからの画像入力・画像処理もできる。グラフィック表示システムはこの装置へのコマンドを UNIX コマンドと同一形式で用いる事ができるようにするとともにプログラムからサブルーチン・コールも出

来るようにしてある。又、ホストコンピュータ側に ACM の siggraph-CORE も装備されており、NEXVS の固有コマンドと結びつけて同時に利用できるようになっている。

7. プロトタイプ・システム

現在、上に概略述べたシステムが、ある流れの解析問題をテスト・ケースとして一応動いている。解析対象としての流れは「一次元定常超音速凝縮流れ」であり、図4にその概観を示した。流れは $y \rightarrow \infty$ か

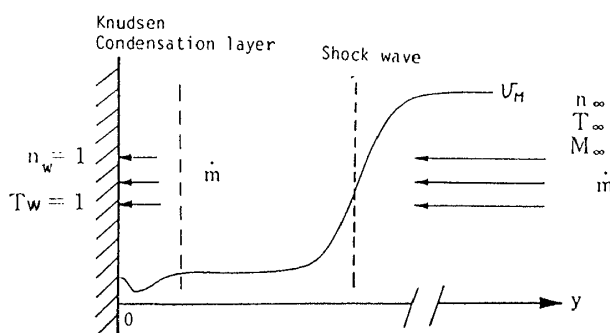
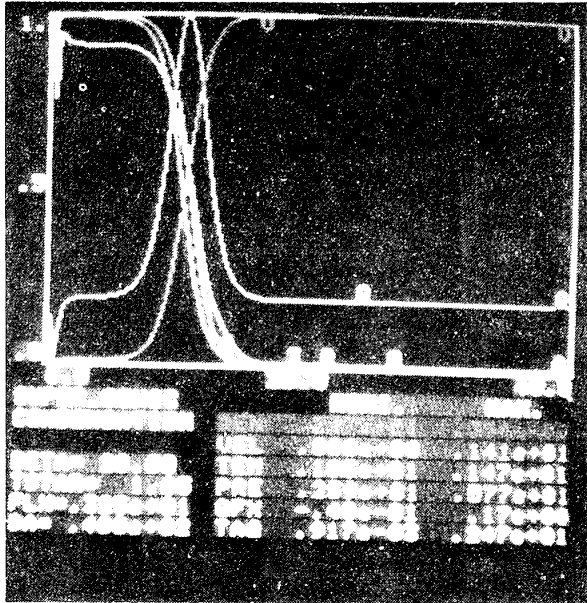


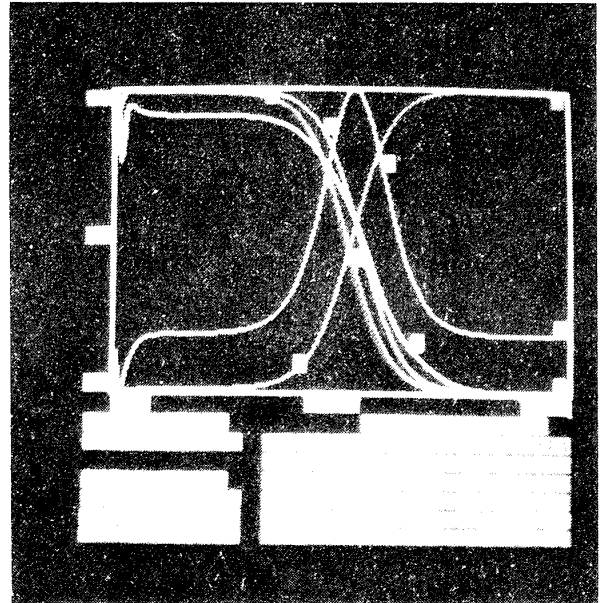
図4 テスト・ケースの流れ

ら超音速で流入し、 $y \simeq 15$ 付近で衝撃波を通過し、 $y = 0$ にある壁の近傍に形成された凝縮層を通り、壁に吸収(凝縮)される。このような流れが Boltzmann 方程式の近似解として数値解析されているが^{(3),(4)}、これらの数値データを一たんデータ・ファイルに入れ、本データベース・システムを通してグラフィック表示したのが写真1~3である。これらの表示とその変更は全部で1分以内で可能である。DBシステムの特徴はすべてのデータが常にハード・ディスク内にあり、即座に取り出せる事にある。各数値例から特性値を1個ずつ取り出してグラフ化する、などというのはその典型である。写真4は流れのパラメータである (β, p_∞) の組について解として得られた凝縮量との関係を表わしたグラフである。これはデータ・ファイル操作がかなり混じるので約10分くらいの時間がこの写真のようなグラフを出力させるまでにかかる。

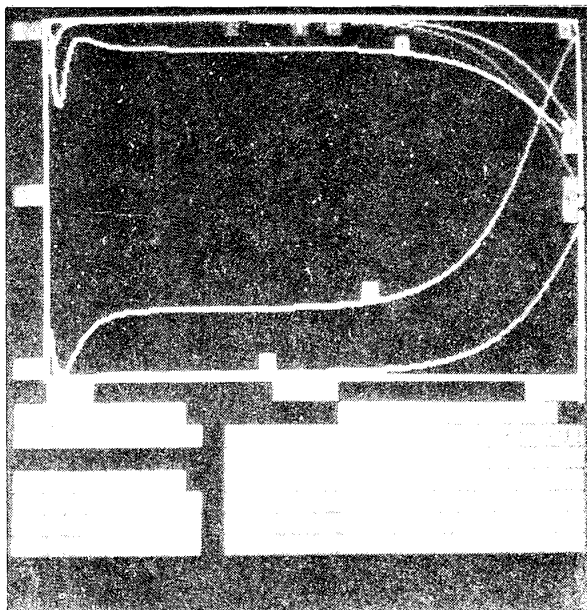
現在も、システムは開発中であり、解析とシステム開発を同時進行させている。



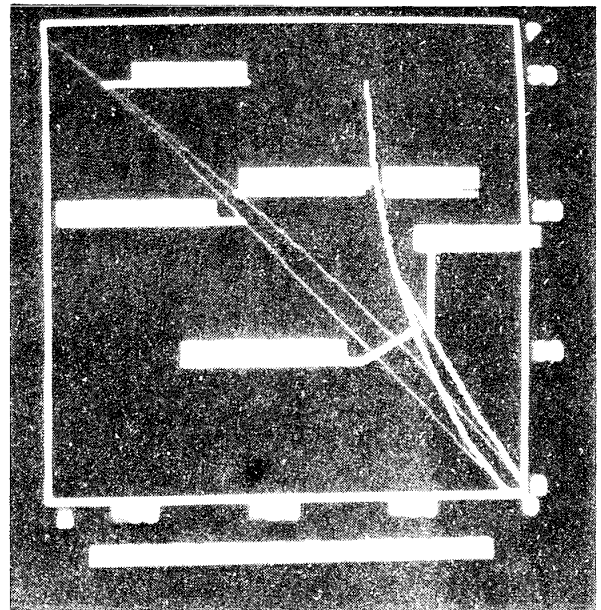
(1)



(2)



(3)



(4)

図5 データのグラフィック表示例

参考文献

- 1) Henry McGilton and Rachel Morgan, "Introducing the UNIX System", McGraw-Hill, New-York, 1983.
- 2) UNIX programmer's manual Vol. 1, 2a, and 2b, - Supplementary Documents Seventh edition. -, Bell Laboratory, 1978.
- 3) M. Hatakeyama and H. Oguchi, "Kinetic Approach to Non-linear Condensation of Flowing Vapor," Rarefied Gas Dynamics, Vol. 2, edited by R. Campargue, Comminssariat A L'energie Atomique, Paris, 1979, pp. 1293-1303.
- 4) M. Hatakeyama, "Kinetic Approach to Supersonic Condensation Processes and Bifurcation Phenomena," in Rarefied Gas Dynamics", ed. by H. Oguchi, Vol. 2, University of Tokyo Press, 1984, pp. 909-916.