

HITAC 102 のプログラムライブラリ

Program Library of HITAC 102

清 野 武*
Takeshi Kiyono

内 容 梗 概

HITAC 102 形デジタル電子計算機は KDC I 形と命名され、昭和 35 年 10 月に京都大学に納入された。本計算機のサブルーチンについては計算機納入 1 年前より京都大学において開発を進め、その後も改訂増補を行ない昭和 37 年 3 月末には 130 余種が完成した。今回これらのサブルーチンを中心として HITAC 102 形デジタル計算機のソフトウェアの一端について説明する。

1. 緒 言

HITAC 102 形デジタル電子計算機は中形汎用電子計算機として計画され通商産業省電気試験所に納入した MARK-V にその基本技術を取り、これを京都大学の長期間の指導と協力により改良し 35 年 10 月完成したもので、従来の計算機使用経験に基づく計算機の使用の容易化を図るため命令体系その他に種々のくふうを加えてある。

この HITAC 102 のサブルーチン（以下 S R と云う）については計算機納入前より京都大学において検討、作成が進められており、37 年 3 月末には 131 種が正式に完成し S R として登録された（第 1 表参照）。なお開発はこれで終わったものではなく、今後は質的に

高いものを開発しプログラムライブラリの内容充実をはかる予定である。

本報告ではこれらの S R を中心として、HITAC 102 のソフトウェアの一端を紹介する。なお HITAC 102 のハードウェアについてはすでに本誌にて紹介済である⁽¹⁾。

2. ライブラリ整備に関する方針⁽²⁾⁽³⁾

2.1 分 類

HITAC 102 のプログラムライブラリは便宜上第 1 表のように分類される。通常の数値計算に必要な S R の多くは、第 1 表の初めの 4 項目：F（関数）、M（マトリクス、一次方程式）、E（方程式）、I（積分、微分方程式）に含まれ、したがってこれらがプログラムライ

第 1 表 HITAC 102 のサブルーチン一覧表（37 年 3 月 31 日現在）

大 分 類	小 分 類	説 明	昭和 35 年度		昭和 36 年度		計	お も な 内 容
			4～9月	10～3月	4～9月	10～3月		
STANDARD ROUTINE	F 1	初 等 関 数	19		6	3	28	$e^x, \sin x, \tan^{-1} x, \sqrt{x}, \log x$
	F 2	特 殊 関 数	12	11	3		26	ベッセル関数, だ円関数, その他
	M 1	マ ト リ ク ス		2	1		3	行列の積, 逆行列, 固有値
	M 2	一 次 方 程 式	1	2		7	10	消去法, 傾斜法, 繰返し法
	M 3	行 列 方 程 式		1	1		2	Choleski 法, 永年行列式
	E 1	代 数 方 程 式		1			1	Bairstow 法
	E 2	超 越 方 程 式	2	2			4	関数の最小化
	E 3	逆 補 間	1				1	逆 補 間
	I 1	積 分	1	1			2	Gauss の 6 点法, Simpson 法
	I 2	微 分 方 程 式			1		1	関数表による数値微分
	I 3	微 分 方 程 式		3		2	5	Runge-Kutta-Gill 法, Hamming 法, その他
	I 4	偏 微 分 方 程 式		2	5		7	Laplace の偏微分方程式の解
	I 5	積 分 方 程 式					0	
	P 1	プ リ ン ト (AC)	3				3	(省 略)
	P 2	プ リ ン ト (ド ラ ム)	8	3			11	(省 略)
	P 3	特 殊 印 刷	2				2	(省 略)
	X 1	乱 数		1			1	一 様 乱 数
	X 2	補 間		4			4	Everett の公式による補間法
	X 3	プ ロ グ ラ ム に よ る 演 算		7			7	倍精度演算による加減乗除
	X 4	そ の 他					0	
	小 計			49	40	17	12	108
SPECIAL ROUTINE	ST	確 率, 統 計		1			1	系列相関係数
	LP	LP, OR					0	
	SX	そ の 他					0	
	小 計			0	1	0	0	1
UTILITY ROUTINE	RDIN	初 期 入 力 プ ロ グ ラ ム		1			1	初期入力ルーチン
	SYCS	Symbolic Coding System		2	1		3	SYCS
	TABL	作 表 プ ロ グ ラ ム	4				4	関数表の作表
	DEBG	デ バ ッ ギ ン グ	1	1	1		3	メモリダンプ
	DIAG	機 械 診 断 ル ー チ ン					0	
	TEST	テ ス ト プ ロ グ ラ ム					0	
	DEMO	デモンストレーションプログラム					0	
	UTILX	そ の 他					1	命令のひん度調査
	小 計			5	5	2	0	12
合 計			54	46	19	12	131	

* 京都大学教授 工博

ブラリの過半を占める。

P (プリント) は主として出力用 SR であるが、デバッキング (debugging) 用の SR も含まれている。X (雑) には上記以外の各種の SR が含まれる。

特殊ルーチンとしては、ST, LP, SX などの項目が挙げられているが、具体的に検討されたものはわずかに 1 種である。

ここでいう utility routine とは、標準的な SR の形はとらないが、主プログラムの実行を助け、あるいはその診断に役立つような補助ルーチンを指している。これは全く便宜上の分類であって、若干の独立したプログラム (機械の検査用など) もこれに含ませてある。

2.2 SR に関する規約

閉じた SR に関しては、その使用に当たっての混乱を避けるために、二、三の一般的規約を設けている。いずれもごく普通的方式であるが、プログラムの負担をなるべく軽くするとともに、SR の側にも甚だしい面倒が起こらない点で実務的であると思われる。

(1) 主プログラムとの結合

主プログラムから SR に移る際には命令 JSX を使用し、これに必要なインデックスレジスタとしては IR 3 を当てる。プログラムパラメータが必要な場合には、JSX に引き続いて calling sequence の形にパラメータを並べる。

(2) アドレス方式

HITAC 102 は 3 種の異なった記憶装置 (磁気ドラムの normal band: 0~3999 番地, 磁気ドラムの quick access band: 4000~4199 番地, 磁気コア: 4200~4249 番地) をもつことができ、1 個の SR が異なった場所で実行されることも予想しなければならないので、SR 内でのアドレスはすべて line-relative (present address relative) に書かれている。

(3) IR の saving

SR 内で IR を使うときは常にその内容をセーブ (save) し、主プログラムに帰るときは、必ずもとの内容を復元するように考慮されている。したがって、プログラムは SR 内で 3 個の IR がどのように使われているかについて全く顧慮する必要はない。ただし、IR 3 は上述のように linkage に使われるので、SR にはいる前の IR 3 の内容をセーブしたいときは、その手続きは当然主プログラムで行なわれなければならない。

3. プログラムライブラリの現状⁽⁴⁾

昭和 37 年 3 月 31 日現在登録を終わった SR は第 1 表のとおりである。これらの SR については特にとりたてて説明すべき点もなく、またその精度、時間、長さに関しては改善を要するものも少なくないと思われるが、現在の段階におけるプログラムライブラリの内容をごく簡単に紹介しておくことにする。

3.1 関数値の計算

HITAC 102 は固定小数点 (11 けた) および浮動小数点 (仮数部 9 けた, 指数 -100 ないし +99) の数を扱うことができるので、原則として、変数または関数値がこの範囲を越えないかぎり計算が可能であることをたてまえてしている。また大部分の SR は浮動小数点に関するものであって、固定小数点の演算はむしろ補助的な意味で考慮されている。

(1) F1 (初等関数)

多くは Chebyshev の多項式近似によっているが、Padé の近似その他を利用したものもある。

平方根および立方根に対しては、第 1 近似を

$$y_1 = (a + bx) / (c + dx) \dots\dots\dots (1)$$

の形で計算し、Newton の逐次近似を一定回数だけ反復適用する

方法をとっている。最近の高速度計算機では、Newton の方法を 1 回だけ適用すれば足りるだけの精度をもつ近似値から出発する方法がとられているが待時間の長いドラム計算機では適当な妥協が必要であると思われる。特に現在の HITAC 102 のように、小容量の高速度記憶装置 (磁気コア) をもつ機械では、SR の長さが大いに問題になり、50 語を境として使いやすさが非常に違ってくことも考慮しなければならない。現在のところ、浮動小数点に対する SR のうち、 $\log_e x$, $\sin x$, $\cos x$, $\tan x$ および $\cot x$, $\tan^{-1} x$ および $\cot^{-1} x$, $\sqrt{x}$ (2 種), $\log_{10} x$, e^x (2 種), 10^x の 10 種と、固定小数点に対する e^x , $\log_e x$, $\sin x$, $\cos x$, $\tan x$, $\tan^{-1} x$, $\sqrt{x}$, $1/\sqrt{x}$ などはいずれも 50 語以内にまとめられている。

(2) F2 (特殊関数)

整数次の Bessel 関数 $J_n(x)$, $Y_n(x)$ がまず開発されたが、小さい変数 ($|x| < 3$) に対する $J_0(3)$, $J_1(3)$, $Y_0(3)$, $Y_1(3)$ はそれぞれの定義式を Chebyshev の多項式で変形したものを利用している。特に $J_0(3)$ および $J_1(3)$ は両者を合わせて 50 語に収まり、計算時間も短い。

比較的大きい変数 ($x > 10$) に対しては、 $J_0(A)$, $J_1(A)$, $Y_0(A)$, $Y_1(A)$ を使用する。これらは変数の領域を分割し、適当な整数値のまわりの展開式を利用したもので多くの係数を preset constant として含むためにかなり長い SR となっているが、計算時間は比較的短い。

YN は $Y_0(x)$ および $Y_1(x)$ に対する SR を併用して、高次の $Y_n(x)$ を計算する SR で、周知の漸化式を利用している。

$JN(A)$, $JN(B)$ は広範囲の変数および次数に対して $J_n(x)$ を正確に計算する SR であって、計算方式は簡単であるが、かなり長い時間を要するのが欠点である。一例として $JN(A)$ について説明すれば、まず与えられた変数 ($|x| < 15$) に対し、近似値

$$J_{30}^*(x) = a, \quad J_{31}^*(x) = 0 \dots\dots\dots (2)$$

を仮定し (a は全く任意に選んだ小さい数), 漸化式

$$J_{n-1}^*(x) = \frac{2n}{x} J_n^*(x) - J_{n+1}^*(x) \dots\dots\dots (3)$$

を利用して、近似値 $J_0^*(x)$, $J_1^*(x)$, $\dots\dots J_{29}^*(x)$ を求める。この値を用いて

$$k = J_0^*(x) + 2 \sum_{i=1}^{15} J_{2i}^*(x) \dots\dots\dots (4)$$

を計算すれば、次の関係から $J_n(x)$ の正しい値がえられる。

$$J_n(x) = (1/k) J_n^*(x) \dots\dots\dots (5)$$

この計算をドラムの normal band で行なうと数秒を要するが、quick access band とコアを併用すれば 0.6 秒以下で行なわれる。

Bessel 関数の計算法については古くから多くの研究があり、特に amplitude and phase の方法は利用価値が高いと思われるがまだ試みる機会をもたない。その後さらに他の多くの関数、変形ベッセル関数、だ円関数、 θ 関数、Legendre 関数などが検討され、開発された。

3.2 方程式その他

(1) M (マトリクス, 一次方程式)

この項目に対してもかなりの SR が整備され、特に一次方程式についてはやや詳細な数値的実験を行なわれた。これらに関しては別の機会に報告する予定である。

(2) E (高次代数方程式, 超越方程式)

高次方程式に対してはまず Bairstow の方法が検討され、収束の判定法に工夫を施して実用的な形にまとめられた。他の方法についても検討中である。

任意の方程式 $f(x) = 0$ の実根を求める SR として、RLRT 1,

RLRT 2 がある。これらはそれぞれ、根の近似値および区間が与えられた場合に適用される。

逆内そうの SR (II 1) は、任意の関数 $f(x)$ に対して、 $f(x)=0$ を満足する $f^{-1}(0)$ を求めるためのもので、あらかじめ $f^{-1}(0)$ をはさむ 2 点とそれらの点における関数値が知られているものとする。

3.3 積分 その他

この項目も実際上きわめて重要であるが、現在までに完成された SR はあまり多くない。

(1) I1 (数値積分)

Gauss の 6 点法と Simpson の方法について別個の SR が作られている。一般に Gauss の方法は少数の関数値によって高い精度がえられる点ですぐれているが、精度の判定のために変域の分割を進めると、前回の関数値がほとんど役に立たなくなるという欠点をもっている。Simpson の方法は領域の分割を 2 進的に行なうかぎり、前回の計算結果がすべて有効に利用される点で、計算機に適している。

最近 Chebyshev の近似を利用する方法が報告されているが、これは上記の二つの方法の利点を備えた点ですぐれたものと思われるので、機会をえて検討したいと考えている。

(2) I4 (偏微分方程式)

二次元の Laplace の方程式および拡散方程式に対しそれぞれのプログラムが検討されている。Laplace の方程式について、Kulsrud の加速法を採用し、この効果を確認するための多くの実験を行なった。

3.4 出力 その他

UA または AC の内容を種々の形式でプリントするもの、ドラムに置かれた多くの数を表の形にプリントするもの、ドラムにある多倍長の数値 (11 けた + n 語) を読みやすい形 (5 けたまたは 10 けた区切り、1 行に 50 けた) に並べてプリントするもの、HITAC 102 の標準浮動小数点形式から常用の小数点形式に変換してプリントするもの、短縮された浮動小数点の形でプリントするもの、その他多くの出力用 SR が用意されている。

なお、プログラム進行中に、あるステップにおける各レジスタの内容をプリントする SR (PRREG) は、監視用として、またデバッグ用として役立つであろう。

そのほかの SR としては、擬似乱数の発生、内そう公式、単位変換などがライブラリに収められている。

3.5 Utility Routine

(1) RDIN (初期入力プログラム)

最初に読み込みルーチンをドラムの 0 ~ 6 番地に入れるための操作をできるだけ簡単にするために、特殊の方法が考案され、数次にわたる改良を経て、SY 法が完成され、現在、われわれの標準方式として活用されている。

(2) TABL (作表プログラム)

関数発生用の SR を検査するには、しばしば関数表 (必要によって検査結果を併記) を作らせる方式がとられるが、その都度、適当な標題と正しい format を与える検査用プログラムを作るとは、はなはだ煩わしい作業である。そこで作表に必要なデータ (標題と見出し——アルファベットのまま——; 関数の個数, 1 ブロックの行数, 総行数; 変数の初期値と刻み; 氏名, 日付など) を試験すべき SR のあとから読み込ませると自動的に関数表がプリントされるようなプログラムを作り、多くの SR の検査に利用しているが、非常に便利であることがわかった。

(3) DEBG (デバッグ)

このためにもかなり多くのプログラムが計画されているが、現

在使っているのは、比較的簡単な memory dump である。

(4) その他

機械の診断用には乱数による memory check その他の専用ルーチンを使用するほか、毎朝の試験には簡単な complete program のいくつかを実行させている。

第 1 期受入検査の 1 項目として、長時間連続運転を行なったが、その際には π , e , $\sin 1^*$, $\sqrt{2}$ などの多倍長計算を反復させた。これらのプログラムも日常の試験に流用している。

4. 自動プログラム⁽⁴⁾⁽⁵⁾

広い意味での自動プログラムの一つとして HITAC 102 には symbolic coding system (SYCS) がある。記号方式の利点については改めて述べる必要もないと思うが、京都大学におけるように、学内一般の利用者を対象とし、研究者がプログラマである場合、いわゆる Open shop 制の使い方をする場合には、コーディングについての労力をできるだけ軽くし、誤りを犯す機会をへらすことが望ましく、このためにも SYCS の確立が強く要求されるわけである。

われわれの場合には、計算機の細部の設計の完了と同時に命令に対する略号 (symbolic code) を決定し、プログラミングにはすべてこの記号を用いてきた。すなわち命令語は symbolic location, symbolic operation code, symbolic address を用いて書き、最後にこれらを内部語に書きかえる方式をとっている。この symbolic なプログラムの原稿をほとんどそのままの形でテープにパンチし、これを自動的に内部語に翻訳するのが SYCS の目的であって、多くの assembly program と同様に、(i) floating address, (ii) mnemonic code, (iii) error detection の三つの効果をもっているほかに、プログラム編成を容易にするために若干の pseudo instruction を含んでいる。

SYCS については、詳述を避けるが⁽²⁾⁽³⁾、比較的短いプログラム (ドラム容量の約 17%) で一応 SAP-like な機能をもたせるようにまとめられているので、プリントおよびパンチにかなり長い時間を要する欠点を顧慮しても、なおかなりの実用価値があるものと考えられる。

次に本格的な自動プログラムとしてコンパイラの問題は各方面から注目されているが、特に ALGOL 60 が発表されて以来、これに対する関心が著しく高まり、中形電子計算機にもこれを適用しようとの試みが盛んになってきた。京都大学でも ALGOL 60 の研究グループを作り、これに準拠したコンパイラの作成を進めつつある。

5. 結 言

以上 HITAC 102 のプログラムライブラリの概要について述べ、さらに自動プログラムについて述べた。最後に緒言にもふれているようにプログラムライブラリはここに紹介しただけのものではないので、今後も改訂増補を行ないさらに質的に高度のものを開発する予定である。

参 考 文 献

- (1) 矢島, 太田, 伊与部: 日立評論 42, 116 (昭 35-12)
- (2) HITAC 102 プログラミング, マニュアル (昭 36-3)
- (3) HITAC 102 ライブラリ (昭 36-3)
- (4) 清野: 情報処理 1, 185 (昭 35-12)
- (5) 京都大学電子計算機室: 電子計算機室速報, KDC-I, 1961 (昭 37-2)

* 余談ではあるが Peters の大対数表に出ている $\sin 1$ の値に誤りのあることを見出した。