

ライブラリー・プログラム利用の手引

(数値計算編)

(昭和57年6月)

名古屋大学大型計算機センター

三訂版のまえがき

ライブラリー・プログラム利用の手引改訂版ができてから、またもや4年の歳月が流れました。この間に昭和54年11月のFACOM 230-75からM-200への劇的な機種切替えがあり、その後のTSSの急速な発展などもあって本センターも大いなる変貌を遂げました。誠に隔世の感があります。一方ライブラリー・プログラムの方も多くの新開発や改訂が行われ、地味ではありますが着実な一步一步を歩んでまいりました。その結果、いろいろな意味で、もはや旧版が物の用に立たなくなってきました。それよりも何よりも、旧版の在庫が底をつき、多くの方々に迷惑を掛けている有様です。というわけで、やや遅きに失するそしりは免れませんが、三訂版を発行する運びになりました。

従来のもに新しく開発されたものを加えますと非常な大冊になってしまいますので、今回からは手引書を、「数値計算編」と「サービス・プログラム、応用プログラム編」の2冊に分けることにいたしました。このようにしてもほとんど御不便をお掛けするおそれはないものと信じます。

旧版以上の御愛用を切にお願いいたします。

昭和57年6月

名古屋大学大型計算機センター

ライブラリー・データベース小委員会

目 次

三訂版のまえがき	i
手引の利用に当たって	x
凡 例	xi

連 立 一 次 方 程 式

連立一次方程式ルーチンの選び方	1	
MNORMS/D/Q/C/B	行列の正規化	2
MNRSPS/D	対称正値行列の正規化	3
MNRMBS/D/Q/C/B } MNMBS/D/Q }	帯行列の正規化	4
MDETS/D/Q/C/B	行列式の計算	7
LEQLUS/D/Q/C/B	LU-分解法による連立一次方程式の解法	8
GAUELS/D/Q/C/B	LU-分解法による連立一次方程式の解法	10
LEQBDS/D/Q/C/B	ガウスの消去法による帯行列係数 連立一次方程式の解法	11
BUNCHS/D	バンチの方法による対称行列係数 連立一次方程式の解法	14
BUNCBS/D	バンチの方法による対称バンド行列係数 連立一次方程式の解法	16
CHOLFS/D/Q } MCHLFS/D/Q }	コレスキー法及び改訂コレスキー法による 対称正値連立一次方程式の解法 (密行列)	18
CHOLCS/D/Q } MCHLCS/D/Q }	コレスキー法及び改訂コレスキー法による 対称正値連立一次方程式の解法 (密行列, 圧縮表現)	20
CHOLSK/CHOLSD	コレスキー法による対称正値な 連立一次方程式の解法	22
CHLBDS/D/Q } MCHLBS/D/Q }	コレスキー法及び改訂コレスキー法による 対称正値連立一次方程式の解法 (帯行列)	24
CHLVBS/D	コレスキー法による対称正値な 連立一次方程式の解法 (帯行列, 圧縮表現)	26
PRCGFS/D	前処理付き共役勾配法による対称正値 連立一次方程式の解法	28
CGHTCS/D	共役勾配法による対称正値 連立一次方程式の解法 (圧縮モード)	31
TRIDGS/D	三項方程式の解法	34
TRDSPS/D } TDSPCS/D }	対称正値三項方程式の解法	36

LEQLSS/D	ハウスホルダー変換による一般 連立一次方程式の最小二乗解及び最小 ノルム解	38
LSMNS/D	特異値分解による一般連立一次方程式の 最小二乗最小ノルム解	42
行 列 の 逆 転		
MINVS/D/Q/C/B	行列の逆転	45
MINVSP/MINVDP/MINVQP	対称正值行列の逆転	47
GINVS/D	特異値分解による一般化逆行列	48
固 有 値 解 析		
固有値解析ルーチンの選び方		51
HEQRVS/D	ダブルQR法による実非対称行列の 固有値解析	52
HOQRVS/D } HOQRUS/D }	ハウスホルダー・QR法による実対称行列の 固有値解析	54
HQRIIS/D	ハウスホルダー・QR・逆反復法による 実対称行列の固有値解析	56
HOBSVS/D	ハウスホルダー・二分法による実対称行列の 固有値解析	58
JACOBS/D	閼ヤコビ法による実対称行列の固有値解析	60
RHQRVS/D	ルティスハウザーQR法による実対称帯行列の 固有値解析	61
RHBSVS/D	ルティスハウザー・二分法による対称帯行列の 固有値解析	63
GHQRVS/D } GHQRUS/D }	ハウスホルダーQR法による $Ax = \lambda Bx$ 型の 固有値解析	65
GHQRIS/D	ハウスホルダー・QR・逆反復法による $Ax = \lambda Bx$ 型の固有値解析	67
GHBSVS/D	ハウスホルダー・二分法による $Ax = \lambda Bx$ 型の 固有値解析	69
JENNFS/D } JENNBS/D } GJENBS/D }	ジェニングスの同時反復法による 実対称行列の固有値解析	71
SVDS/D	特異値分解	74
NSHOUS/D	ハウスホルダー・二分割QR・逆反復法による $Av = \lambda v$ 型の固有値解析	77
NGHOUS/D	双三角分解・ハウスホルダー・二分割QR・ 逆反復法による $Av = \lambda Bv$ 型の固有値解析	79
NSJENS/D	ジェニングス法による $Av = \lambda v$ 型の 固有値解析	81
NGJENS/D	双三角分解・ジェニングス法による $Av = \lambda Bv$ 型の固有値解析	83

代数方程式, 非線形方程式

QUADRS/D/Q } CUBICS/D/Q } QUARTS/D/Q }	低次実係数代数方程式の解法	87
GJMKS/D/Q	ガーサイド・ジャラット・マック法による 実係数代数方程式の解法	88
BRENDS/D	ブレンツの方法による非線形方程式の解法	90
BROYDS/D	ブロイデンの方法による非線型 連立一次方程式の解法	92
MINSXS/D	シンプレックス法による関数の最小化	94
FLPOWS/D	ダビドン・フレッチャー・パウエル法による 関数の最小化	96
NOLLS1	準ニュートン法による非線型最小二乗法 サブルーチン	98

補間, 平滑化, 数値微積分

補間, 平滑化ルーチンの選び方		102
DSCI1A DSFI1A } DSCI2A DSFI2A } DSCI3A DSFI3A } DSCI4A DSFI4A }	スプライン補間 (1変数)	103
DSCI1D DSFI1D DSCI5D DSFI5D } DSCI2D DSFI2D DSCI6D DSFI6D } DSCI3D DSFI3D DSCI7D DSFI7D } DSCI4D DSFI4D }	スプライン補間 (2変数)	108
SPLINE } SPLINT } SPLN22 } INTGRL }	スプライン・フィット曲線による近似計算 (導関数, 内挿, 積分の値)	121
HERM31 } HERM51 }	区分的エルミート補間による曲線のあてはめ ...	123
HERM32 } HERM52 }	区分的エルミート補間による曲面のあてはめ ...	125
CFS1A } SFC1A }	スプラインによる曲線のあてはめ	127
CFS2A } SFS1A }	スプラインによる曲面のあてはめ	130
DCOMD1 } DCPFR1 }	複合多項式による曲線のあてはめ	134
LSAICS } LSAICD }	直交多項式による最小二乗近似	136
TRIPCK	不規則分布 2 変数関数データに対する C^k 級補間法 ($0 \leq k \leq 3$)	138

TETPCK	不規則分布3変数関数データに対する C^k 級補間法 ($0 \leq k \leq 1$)	140
フ ィ リ エ 解 析		
FFTC/B	複素高速フーリエ解析	142
FFTS/D	複素高速フーリエ変換	143
FFTR/FFTRD	実高速フーリエ解析	145
FFTRI/FFTRID	実高速フーリエ合成	147
FT235C } FT235R }	サンプル数が $2^k 3^l 5^m$ の形の場合の複素 及び実高速フーリエ変換	148
FFT2DC/FFT3DC	2次元及び3次元複素高速フーリエ解析	150
FFT2DR/FFT3DR	2次元及び3次元実高速フーリエ解析	152
BITREV/BITRVD/BITRVB	ビット逆転によるデータの並べ換え	155
FCOSMS/D } FSINMS/D }	中点公式に基づく高速cosine変換 (FCOSMS/D) 中点公式に基づく高速sine変換 (FSINMS/D)	156
TRIGQP/TRIGQD	2進逆順に並べられた三角関数表	161
FCOSTS/D } FSINTS/D }	台形公式に基づく高速cosine変換 (FCOSTS/D) 台形公式に基づく高速sine変換 (FSINTS/D)	162
FCOSCS/D } FCOSOS/D } FSINOS/D }	閉区間 $(0, \pi)$ で与えられた偶関数のcosine 級数展開 (FCOSCS/D) 閉区間 $(0, \pi)$ で与えられた偶関数のcosine 級数展開 (FCOSOS/D) 閉区間 $(0, \pi)$ で与えられた奇関数のsine級 数展開 (FSINOS/D)	166
FCHB1S/D } FCHB2S/D } FCHB3S/D } FCHBOS/D }	第1種チェビシェフ多項式による関数の フーリエ展開 (FCHB1S/D) 第2種チェビシェフ多項式による関数の フーリエ展開 (FCHB2S/D) ずらしチェビシェフ多項式による関数の フーリエ展開 (FCHB3S/D) 第1種チェビシェフ多項式による開区間上で 与えられた関数のフーリエ展開 (FCHBOS/D)	170
AQDCCS/D } AQDCOS/D }	クレンショウ・カーチス法による自動積分 (閉じた積分公式) (AQDCCS/D) クレンショウ・カーチス法による自動積分 (開いた積分公式) (AQDCOS/D)	177
VCOSS/D } VSINS/D }	cosine級数の値 (VCOSS/D) sine級数の値 (VSINS/D)	181

VCHB1S/D	DCHB3S/D	}	チェビシェフ級数の値 (VCHB1S/D)	
DCHB1S/D	ICHB3S/D		チェビシェフ級数の微係数 (DCHB1S/D)	
ICHB1S/D			チェビシェフ級数の不定積分の値 (ICHB1S/D)	
VCHB3S/D			ずらしチェビシェフ級数の値 (VCHB3S/D)	
			ずらしチェビシェフ級数の微係数 (DCHB3S/D)	
			ずらしチェビシェフ級数の不定積分の値 (ICHB3S/D)	185
VCHB2S/D	}		第2種チェビシェフ級数の値 (VCHB2S/D)	
ICHB2S/D			第2種チェビシェフ級数の不定積分の値 (ICHB2S/D)	190
DRCH1S/D	}		第1種チェビシェフ級数の導関数 (DRCH1S/D)	
DRCH3S/D			ずらしチェビシェフ級数の導関数 (DRCH3S/D)	
IICH1S/D			第1種チェビシェフ級数の不定積分 (IICH1S/D)	
IICH3S/D			ずらしチェビシェフ級数の不定積分 (IICH3S/D)	192
			数 値 積 分	
数値積分ルーチンの選び方				195
GASNS/D	}		ガウス型数値積分	196
GSLNS/D				
GSHNS/D				
TGLEGS/D/Q	}		ガウスの数値積分公式による重率と分点の値	198
TGLAGS/D/Q				
TGHERS/D/Q				
TGLOBS/D				
AQNN5S/D	}		ニュートン・コーツ5 (7, 9) 点則に基づく 適応型自動積分	200
AQNN7S/D				
AQNN9S/D				
DEFINS/D	}		二重指数関数型公式による有限区間積分	203
IMTDES/D				
QDAPBS/D			等差数列的に標本点を増す補間型積分法	206
HINFAS/D	}		二重指数関数型公式による半無限区間積分	208
HINFES/D				
INFINS/D			二重指数関数型公式による無限区間積分	210
TRAPZS/D			台形則による無限区間積分	212
ROMBGS/D			ロンバーグ積分	214
MQPRRS/D			直積型公式による多重数値積分	216
MQNCDS/D			ニュートン・コーツ則の直積による 多重数値積分 (データ入力)	219
MQFSRS/D			完全対称則による多重数値積分	221
AQMDS/D			等差数列的に標本数を増す補間型積分に 基づく自動多重積分	224

AQNDS/D } AQ3DS/D } AQ2DS/D } AQ1DS/D }	自動多重数値積分	228
--	----------------	-----

常 微 分 方 程 式

RK4S/D/Q/C/B	4 次の古典的ルンゲ・クッタ法による 連立1階常微分方程式の初期値問題の 数値解法	231
RKF4AS/D	ルンゲ・クッタ・フェールベルク4次法に よる連立1階常微分方程式の初期値問題の 解法	234
RKM4AS/D	ルンゲ・クッタ・メルルツィ4次法による 連立1階常微分方程式の初期値問題の解法	237
ODEBSS/D	有理補外法による連立1階常微分方程式の 初期値問題の解法	240

特 殊 関 数

SINHP/DSINHP/QSINHP } COSHP/DCOSHP/QCOSHP } TANHP/DTANHP/QTANHP } COTHP/DCOTHP/QCOTHP }	引数 $\frac{\pi}{2}x$ に対する三角関数	243
ASINH/DASINH/QASINH } ACOSH/DACOSH/QACOSH } ATANH/DATANH/QATANH }	逆双曲線関数	245
EXP1/DEXP1/QEXP1 } ALOG1/DLOG1/QLOG1 }	関数 $e^x - 1$ 及び $\log(1+x)$	247
ALANGV/DLANGV	ランジュバン関数	248
CABS1/CDABS1/CQABS1	複素数の実部と虚部の絶対値和	249
FCTRL/DFCTRL/QFCTRL } FFCTR/DFCTR/QFFCTR }	階乗及び二重階乗	250
COMB/DCOMB	二項係数	251
ZETNO/DZETNO/QZETNO	リーマン・ゼータ関数	252
BERNO/DBERNO/QBERNO	ベルヌーイ数	253
GAMCO/DGAMCO/QGAMCO	$1/\Gamma(x)$ のテーラー級数展開係数	254
RGAMA/DRGAMA	ガンマ関数の逆数	255
CGAMMA/CDGAMA	複素変数のガンマ関数	256
DIGAM/DDIGAM	ディガンマ関数	257
EI/DEI } SI/DSI } CI/DCI }	指数, 正弦及び余弦積分	258
FRESS/DFRESS } FRESC/DFRESC }	フレネル正弦及び余弦積分	260
CND/DCND } CNDC/DCNDC }	累積正規分布関数とその余関数	262

ACND/DACND }
 ACNDC/DACNDC }
 AERF/DAERF }
 AERFC/DAERFC }
 PN/DPN }
 PNM/DPNM }
 QNOME/DQNOME
 CELI1/DCELI1 }
 CELI2/DCELI2 }
 ICEILS/D
 JACELS/D

累積正規分布関数とその余関数の逆関数 263
 誤差関数及びその余関数の逆関数 264
 ルジャンドル多項式及び
 ルジャンドル陪多項式 266
 楕円関数のノーム 268
 第1種及び第2種完全楕円積分 269
 第1種及び第2種不完全楕円積分 271
 ヤコビの楕円関数 sn, cn, dn 273

ベ ッ セ ル 関 数

$\overset{0}{\circ}$
 BJ0/DJ0 }
 BJ1/DJ1 }
 BY0/DY0 }
 BY1/DY1 }
 $\overset{0}{\circ}$
 BI0/DI0 }
 BI1/DI1 }
 BK0/DK0 }
 BK1/DK1 }
 SJ0/DSJ0 }
 SJ1/DSJ1 }
 SY0/DSY0 }
 SY1/DSY1 }
 SI0/DSI0 }
 SI1/DSI1 }
 SK0/DSK0 }
 SK1/DSK1 }
 BJ0I0/DJ0I0 }
 BJ0I1/DJ0I1 }
 BY0I0/DY0I0 }
 BY0I1/DY0I1 }
 BI0I0/DI0I0 }
 BI0I1/DI0I1 }
 BK0I0/DK0I0 }
 BK0I1/DK0I1 }
 $\overset{0}{\circ}$
 BJN/DJN }
 BYN/DYN }
 BJF/DJF
 SJN/DSJN }
 SYN/DSYN }
 $\overset{0}{\circ}$
 BIN/DIN }
 BKN/DKN }
 BIF/DIF

0次及び1次のベッセル関数 274
 0次及び1次の変形ベッセル関数 276
 0次及び1次の球ベッセル関数 278
 0次及び1次の変形球ベッセル関数 280
 ベッセル関数の積分 282
 変形ベッセル関数の積分 284
 整数次のベッセル関数 286
 非整数次の第1種ベッセル関数 288
 整数次の球ベッセル関数 290
 整数次の変形ベッセル関数 292
 非整数次の第1種変形ベッセル関数 294

SIK/DSIK } SKN/DSKN }	整数次の変形球ベッセル関数	296
BESJNC/B } BESINC/B }	複素変数のベッセル関数	298
AI/DAI } AIP/DAIP } BI/DBI } BIP/DBIP }	エァリ関数とその導関数	299
線 形 計 画 法		
LIPS/D	クリス・クロス法を用いた線形計画 問題解法ルーチン	302
SIMPLX/SIMPLD	シンプレックス法を用いた線形計画法	304
多 重 精 度 演 算		
M-TYPE	誤差評価の可能な多重精度演算 (M型)	308
S-TYPE	誤差評価の可能な多重精度演算 (S型)	313
E-TYPE	誤差評価の可能な多重精度演算 (E型)	318
I-TYPE	誤差評価の可能な多重精度演算 (I型)	324
MMFUNC	多重精度の初等関数 (M型)	330
そ の 他		
RANDOM/DRANDM	一様乱数の生成	337
BITLOGIC	4バイトデータ間のビット毎の論理演算	338
ROUND/DROUND	実数の0捨1入	339

CENT.NLB&C

手引の利用に当たって

「ライブラリー・プログラム利用の手引」は、当センターに登録されている利用者及びセンター作成のライブラリー・プログラムを利用するためのもので、「数値計算編」と「サービス・プログラム、応用プログラム編」の二編に分かれています。本「数値計算編」は、数値計算の広汎な分野のための基本的でしかも重要なプログラムを収録しており、質量共に誇るに足るものと自負するものであります。

この手引に収められているプログラムの大部分はそのソース・プログラムが公開されています。ソース・プログラムのリスト、カード又はデータ・セットへのコピーを希望される場合は、次のような手続きを、利用者自身で実行してください。

なお、本書のライブラリー・プログラムを利用して得られた成果を、論文や報告として発表される場合には、当該プログラムについて一言言及していただくことができれば幸甚に存じます。

① TSS の場合は次のコマンドを入力します。

```
NLIBRARY ELM (ライブラリー名) [CARD(ON)] [DS(ON)] [SLAVE(ON)]
```

ソース・リストのみが欲しい場合は〔 〕内を省略できます。SLAVE(ON)を指定すると、プログラムのスレーブ・ルーチンがすべて出力されます。

② BATCH の場合は次のジョブを実行します。

```
//コメント JOB 利用者番号, パスワード
// EXEC NLIBRARY, ELM = プログラム名 [, CARD = ON]
//           [, DS = ON] [, SLAVE = ON]
[/ / FT99 F001 DD DSN = データ・セット名……]
//
```

また、プログラムの利用例題も次のような手続きで入手できます。

① TSS の場合

```
EXAMPLE NAME (ライブラリー名) [CARD(ON)] [DS(ON)]
```

② BATCH の場合

```
//コメント JOB 利用者番号, パスワード
// EXEC EXAMPLE, NAME = プログラム名 [, CARD = ON] [, DS = ON]
[/ / FT99 F001 DD DSN = データ・セット名, ……]
//
```

なお、当センターで利用できるライブラリー・プログラムとしては、富士通開発の科学技術計算ライブラリー SSL II, アプリケーション・プログラム、及び各種のソフトウェア会社より購入したプログラムがあります。センターニュースの巻末に掲げた一覧表を参照してください。

凡 例

この手引書に解説されているライブラリー・プログラムは大別して、関数副プログラムとサブルーチン副プログラムに分けられる。その各々に対して、いくつかの通則のようなものがあり、本文では記述を簡単にするためにそれを利用している。したがって本書を使用される前に、必ず以下の説明を読まねんことを希望する。

(I) 関数副プログラム

(1) 関数名と型

実数型の関数名は FORTRAN の暗黙の型指定の規約に従っている。

例 BJ0, ACND

倍精度実数型の関数名、4倍精度実数型の関数名は（存在する場合には）、対応する実数型の関数名の先頭にそれぞれD及びQを加える。ただし、いくつかの例外がある。

例 SINHP, DSINHP, QSINHP

例外の例 ALOG1, DLOG1, QLOG1

しかしながら、倍精度及び4倍精度用の関数名がそれぞれD及びQで始まることは厳格に守られている。ここで注意を喚起したいことは、これらの関数を使用する場合には、それを引用しているすべてのプログラム単位で、それらの関数名に適切な型宣言を行わねばならないことである。

例 DOUBLE PRECISION DCOSHP, DJ1
REAL*8 DCELI1, DCELI2
REAL*16 QSINHP, QASINH

倍精度及び4倍精度の関数名の頭字が必ずD及びQであることを利用すれば、他の変数も適当に考慮した上で IMPLICIT 文を用いるのが便利である。

例 IMPLICIT REAL*8(D)
IMPLICIT REAL*8(A-H,O-Z)

このようにすれば、個々の関数名を一々宣言する労が省ける。

(2) 関数値の精度

原則として使用桁数一杯の精度を目標として作成されている。しかしながら、原理的なあるいは技術的な困難¹⁾のために、この目標は必ずしも完全に達成されているとは云えない。特に二変数の関係や複素数の関数ではその傾向が強い。

(3) 引数の制限

(a) 定義域が限られている。

例 ALOG1

この関数は $\log(1+x)$ を計算する。したがって、 $x > -1$ という制限がつく。

1) 二宮市三；“数学ソフトウェアの現状と問題点”，情報処理，Vol. 23，pp. 109 - 117（1982）。

(b) 特異点が存在する。

例 TANHP

この関数は $\tan \frac{\pi}{2}x$ を計算する。したがって、 x が奇整数の場合は特異点となり計算できない。

(c) 関数値がオーバーフローとなる。

例 BIO

この関数は変形ベッセル関数 $I_0(x)$ のためのもので、大きな x に対しては、標準関数 EXP を引用して e^x を計算する。したがって、EXP のオーバーフロー限界 $252 \log_e 2 \approx 174.673$ がこの関数の引数の上限となる。

(d) 関数値が無意味となる。

例 BJO

この関数はベッセル関数 $J_0(x)$ のためのもので、大きな x に対して、標準関数 SIN 及び COS を引用している。したがって、SIN, COS での引数制限、 $|x| \leq 2^{18}\pi \approx 8.23 \cdot 10^5$ がこの関数の引数の制限となる。

このような例は非常に多い。注意すべきことは、 $2^{18}\pi$ という値自体が必然的な値ではないということ、及び制限以内でも、この制限に近づくにつれて関数値の意味のある桁数は次第に減少しているということである。

なお、関数値がアンダーフローする場合は、これを 0 として特別な処置は取らない。

(4) エラー処理

引数が制限を破った場合には、エラーとしてメッセージを印刷し、関数値を一様に 0 として計算を続行する。メッセージには関数名、引数値、関数値 (0) 及びエラーの理由が印刷される。

例 ALOG1 ERROR ARG=-0.2000000 E+01 VAL=0.0 ARG.LT.-1

エラーが続出して計算が全く意味のなくなる場合を考慮して、エラー処理プログラムでエラーの回数をカウントし、それがあらかじめ定められたある制限を超えた場合に計算を停止するようにしている。しかしながら、このような処置はすべての利用者が常時希望するとは限らないので、メッセージの印刷をも含めてその採否を利用者の自由意志にゆだねることとする。そのためにサブルーチン FNERST が作られており、次のように利用できる。

CALL FNERST (IABORT, MSGPRT, LIMERR)

引 数	型 と 種 類	属 性	内 容
IABORT	整 数 型	入 力	IABORT = 0 計算を停止させない。 IABORT \neq 0 計算を停止させる。
MSGPRT	整 数 型	入 力	MSGPRT = 0 メッセージを印刷しない。 MSGPRT \neq 0 メッセージを印刷する。
LIMERR	整 数 型	入 力	エラーの回数の上限。

なお、このサブルーチンをコールしないときは、標準値として

IABORT = 1, MSGPRT = 1, LIMERR = 10

が設定される。

(II) サブルーチン副プログラム

(1) サブルーチン名と型

サブルーチン名の頭字には型の意義はない。同じ目的を持ち、型だけが異なる一群のサブルーチンは名前の末尾の1字によって区別される。その原則は次の通りである。

単精度用	S
倍精度用	D
4倍精度用	Q
複素数用	C
倍精度複素数用	B

ただし、いくつかの例外もある。

例 LEQLUS/D/Q/C/B
RK4S/D/Q/C/B
GJMKNKS/D/Q

例外の例 FFTR/FFTRD
MINVSP/MINVDP

(2) 引数

引数の属性として次の4種類を区別する。

入力……………サブルーチンコールに先立って、利用者が設定すべきデータである。特に断らない限り、サブルーチン脱出時そのまま保存されている。関数名やサブルーチン名が引数になるときもこの部類に入れる。この場合には、それらの名前をEXTERNAL宣言することを忘れないように注意する。

出力……………サブルーチン内で作成される、利用者にとって意味のあるデータである。

入出力……………領域節約のために入力と同じ場所に出力が行われる場合をいう。入出力引数が単純変数である場合には、実引数として定数を入れることは厳に慎むべきである。例えばLEQLUSの入出力引数INDに定数1を入れてコールすると、正常終了の場合IND=0と出力されるが、この場合には定数1がすべて0に変えられてしまうことになる。

作業領域……………サブルーチン内の計算の必要上利用者に要求される領域のことで、サブルーチン脱出時のその内容は利用者にとって意味のないものである。

各サブルーチン群に対して、その引数の型、属性及び内容が説明されているが、簡単のためにこれらの説明は単精度用のものだけに行われ、外のものに対しては適当に型だけを読み替えるようにしてある。

ある引数を全く使わないような仕方ではコールする場合には、その引数に相当する領域を用意する必要がなく、また、その場所に何を書いてもよい。異なる引数に同じ領域を割り当てることができるのは、SVDSの場合のようにその旨が積極的に指摘してあるときに限る。引数に対して特殊の要求がされている例(FT 235 R)もある。

数値積分ルーチンや微分方程式解法ルーチンでは関数ルーチンやサブルーチンを利用者側で作成

することが要求される。この場合にその引数の数、型及び順序は指定されている通りにしなければならない。規定の引数以外の副引数が必要な場合は、これらを COMMON 領域にとり、主プログラムとの連絡を図るようにする。実例は個々のルーチンの解説を参照されたい。

```

SUBROUTINE MATHLIB(
    N,NR,NR1,NR2,NR3,NR4,NR5,NR6,NR7,NR8,NR9,
    NR10,NR11,NR12,NR13,NR14,NR15,NR16,NR17,
    NR18,NR19,NR20,NR21,NR22,NR23,NR24,NR25,
    NR26,NR27,NR28,NR29,NR30,NR31,NR32,NR33,
    NR34,NR35,NR36,NR37,NR38,NR39,NR40,
    NR41,NR42,NR43,NR44,NR45,NR46,NR47,NR48,
    NR49,NR50,NR51,NR52,NR53,NR54,NR55,NR56,
    NR57,NR58,NR59,NR60,NR61,NR62,NR63,NR64,
    NR65,NR66,NR67,NR68,NR69,NR70,NR71,NR72,
    NR73,NR74,NR75,NR76,NR77,NR78,NR79,NR80,
    NR81,NR82,NR83,NR84,NR85,NR86,NR87,NR88,
    NR89,NR90,NR91,NR92,NR93,NR94,NR95,NR96,
    NR97,NR98,NR99,NR100)

```

END SUBROUTINE

このルーチンでは、共通変数として、NR1-NR100 の変数を用いて、各変数の型、次元、並びに初期値を指定する。変数の型は、`REAL`、`INTEGER`、`COMPLEX`、`CHARACTER` の 4 種類に限定する。また、変数の次元は、1、2、3 の 3 種類に限定する。初期値は、`0`、`1`、`2`、`3`、`4`、`5`、`6`、`7`、`8`、`9`、`10`、`11`、`12`、`13`、`14`、`15`、`16`、`17`、`18`、`19`、`20` の 21 種類に限定する。

このルーチンでは、共通変数として、NR1-NR100 の変数を用いて、各変数の型、次元、並びに初期値を指定する。変数の型は、`REAL`、`INTEGER`、`COMPLEX`、`CHARACTER` の 4 種類に限定する。また、変数の次元は、1、2、3 の 3 種類に限定する。初期値は、`0`、`1`、`2`、`3`、`4`、`5`、`6`、`7`、`8`、`9`、`10`、`11`、`12`、`13`、`14`、`15`、`16`、`17`、`18`、`19`、`20` の 21 種類に限定する。

このルーチンでは、共通変数として、NR1-NR100 の変数を用いて、各変数の型、次元、並びに初期値を指定する。変数の型は、`REAL`、`INTEGER`、`COMPLEX`、`CHARACTER` の 4 種類に限定する。また、変数の次元は、1、2、3 の 3 種類に限定する。初期値は、`0`、`1`、`2`、`3`、`4`、`5`、`6`、`7`、`8`、`9`、`10`、`11`、`12`、`13`、`14`、`15`、`16`、`17`、`18`、`19`、`20` の 21 種類に限定する。

このルーチンでは、共通変数として、NR1-NR100 の変数を用いて、各変数の型、次元、並びに初期値を指定する。変数の型は、`REAL`、`INTEGER`、`COMPLEX`、`CHARACTER` の 4 種類に限定する。また、変数の次元は、1、2、3 の 3 種類に限定する。初期値は、`0`、`1`、`2`、`3`、`4`、`5`、`6`、`7`、`8`、`9`、`10`、`11`、`12`、`13`、`14`、`15`、`16`、`17`、`18`、`19`、`20` の 21 種類に限定する。

連立一次方程式ルーチンの選び方

名大プログラム・ライブラリーには、係数行列の型、性質及び構造に即して多数のすぐれた連立一次方程式解法サブルーチンが用意されているので、これらを以下に示す指針に基づいて慎重に選択すれば、精度、速度及び記憶容量のすべての面で多大の効果を上げることができる。簡単のために推奨ルーチンの名前は単精度用のもので代表させることにする。また、*印のついたものは、アセンブリ言語で書かれており、特にこれを推奨する。

(A) 実係数

1. 非対称

- | | |
|------------|---------|
| (1) 密行列 | LEQLUS* |
| (2) 帯行列 | LEQBDS |
| (3) 三重対角行列 | TRIDGS |

2. 対 称

- | | |
|---------|--------|
| (1) 密行列 | BUNCHS |
| (2) 帯行列 | BUNCBS |

3. 正值対称

- | | |
|------------|------------------|
| (1) 密行列 | CHOLFS*, MCHLFS* |
| (2) 帯行列 | CHLBDS*, MCHLBS* |
| (3) 可変帯幅行列 | CHLVBS |
| (4) 三重対角行列 | TRDSPS, TDSPCS |

4. 一般連立一次方程式

LEQLSS, LSMNS

(B) 複素係数

- | | |
|--------|--------|
| 1. 密行列 | LEQLUC |
| 2. 帯行列 | LEQBDC |

連立一次方程式の係数の間及び解の間にはなほだしい大きさの違いがあるときは、一般によい精度は得られない。正規化、変数変換などの手段によって係数及び解が同じ程度の大きさになるようにしておくことが肝要である。

連立一次方程式 $Ax = b$ の解が理論的に $x = A^{-1}b$ と書かれるという理由で、連立一次方程式の解法に逆行列ルーチンを利用する人が多いようであるが、これは連立一次方程式ルーチンを用いる場合に比べて、計算時間が3倍もかかり、解の精度も著しく劣悪であるので是非とも止めていただきたい。

同じ係数の方程式を右辺列だけを変えて反復的に求解する場合には、一度だけ逆行列を計算して $x_i = A^{-1}b_i$, $i = 1, 2, \dots$ と計算するのは合理的に見えるが、これも名大ルーチンのすべてが具備している分解成分再利用の機能を利用する方が遥かに有利である。

要するに連立一次方程式の求解に逆行列を使用することは、逆行列の濫用以外の何物でもない。

参 考 文 献

- 1) 二宮市三; “数学ソフトウェアの現状と問題点”, 情報処理, Vol. 23, No. 2, pp. 109-117 (1982)。

MNORMS/D/Q/C/B

Normalization of a Matrix

行列の正規化

作成	二宮 市三 1977年4月
形式	サブルーチン 言語; FORTRAN サイズ; 20, 21, 20, 23, 24行

(1) 概要

与えられた行列の各行を 2^n の形の適当な数で除して、各行の行列要素の最大絶対値を1の程度にそろえる。

(2) 使用法

CALL MNORMS/D/Q/C/B (A, KA, N, M, S, ILL)

引数	型と種類*	属性	内容
A	実数型 2次元配列	入出力	正規化の対象となる行列。
KA	整数型	入力	Aの配列宣言における第1添字の値。KA \geq N
N	整数型	入力	Aの行数。N \geq 2
M	整数型	入力	Aの列数。M \geq N
S	実数型 1次元配列	出力	S(I) (I = 1, ..., N)には第1行を正規化するため2のべきの形の除数が入る。
ILL	整数型	出力	ILL = 0 : 正常終了。 ILL = 30000 : KA, N, Mに関する制限が破られたとき。 ある行がすべて0であるときはその行番号。

* MNORMD (MNORMQ, MNORMC, MNORMB) の場合には、実数型をすべて倍精度実数型 (4倍精度実数型, 複素数型, 倍精度複素数型) とする。

(3) 備考

- 連立一次方程式を解くための前処理として正規化を行う場合、右辺列を係数行列の右に合併する方式ならば、この拡大行列に対して本ルーチンを適用すれば、正規化された方程式を解いて得られた解がそのままとの方程式の解である。すなわち、後処理は不必要である。
- 逆行列を求める場合には、正規化された行列の逆行列の各列を同じ番号の行に対するスケール・ファクター S (I) で除したものが、もとの行列の逆行列である。
- 正規化により、対称行列は一般に非対称行列となる。

MNRSPS / D

Normalization of a Symmetric Positive Definite Matrix

対称正値行列の正規化

作成	二宮 市三	1977年4月
形式	サブルーチン 言語; FORTRAN サイズ; 20, 21行	

(1) 概要

与えられた対称正値行列の各行、各列を 2^n の形の適当な数で除して、対称正値性を保存したままで各行各列の要素の最大絶対値を1の程度にそろえる。

(2) 使用法

CALL MNRSPS/D (A, KA, N, M, S, ILL)

引数	型と種類*	属性	内容
A	実数型 2次元配列	入出力	正規化の対象となる行列。対角線を含み、右上半分だけが処理される。他の部分は保存される。
KA	整数型	入力	Aの配列宣言における第1添字の値。 $KA \geq N$
N	整数型	入力	Aの行数。 $N \geq 2$
M	整数型	入力	Aの列数。 $M \geq N$
S	実数型 1次元配列	出力	S(I) (I = 1, 2, ..., N)には第I行、第I列を正規化するために除した2のべきが入る。
ILL	整数型	出力	ILL = 0 : 正常終了。 ILL = 30000 : KA, N, Mに関する制限が破られたとき。対角要素が正でないとき、その要素の行番号が入る。

* MNRSPDの場合は実数型をすべて倍精度実数型とする。

(3) 備考

- 連立一次方程式を解くための前処理として正規化を行う場合、右辺列を係数行列の右に合併する方式ならば（例えば CHOLSK など）、合併行列に本ルーチンを適用すればよい。右辺列を分離する方式ならば（例えば CHOLFS など）、係数行列に本ルーチンを適用した後、右辺列の第I要素を、スケール因子 S(I) で割る必要がある。どちらにしても、連立一次方程式を解いた後、解の第I成分をスケール因子 S(I) で割ることが必要である。
- 逆行列を求めるための前処理として、本ルーチンを使用した後、逆行列の各行各列を対応するスケール因子で割る必要がある。

MNRMBS / D / Q / C / B

MNMBSS / D / Q

Normalization of Band Matrices

帯行列の正規化

作 成	二宮 市三 1982年5月
形 式	サブルーチン 言語; FORTRAN77 サイズ; 25, 26, 26, 26, 27, 25, 26, 26行

(1) 概 要

一般の帯行列用のサブルーチン MNRMBS/D/Q/C/B では、与えられた帯行列の各行を、その要素の最大絶対値に近い2のべきで除して正規化する。

対称正値帯行列用のサブルーチン MNMBSS/D/Q では、与えられた帯行列の各行及び各列を、対角要素の平方根に近い2のべきで除して正規化する。

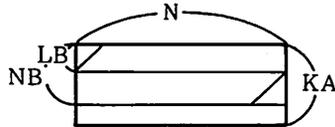


図 1

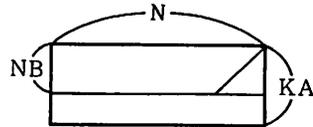


図 2

(2) 使用法

CALL MNRMBS/D/Q/C/B (A, KA, N, NB, LB, S, ILL)

CALL MNMBSS/D/Q (A, KA, N, NB, S, ILL)

引 数	型 と 種 類*	属 性	内 容
A	実 数 型 2 次 元 配 列	入 出 力	正規化の対象となる行列。 一般帯行列の場合は図1に示すように長方形化する。すなわち (I, J) 要素を $A(J-I+LB, I)$ に格納する。 正値対称行列の場合は図2に示すように長方形化する。すなわち (I, J) 要素を $A(I-J+1, J)$ に格納する。
KA	整 数 型	入 力	A の整合寸法 (配列宣言における第1添字)。 $KA \geq NB$
N	整 数 型	入 力	A の次数 (列数)。 $N \geq NB$
NB	整 数 型	入 力	一般行列の場合は全帯幅。 $NB \geq LB$ 正値対称行列の場合は半帯幅。 $NB \geq 1$
LB	整 数 型	入 力	A の左帯幅。 $LB \geq 1$
S	実 数 型 1 次 元 配 列	出 力	正規化因子。各行 (列) を除するために用いられた2のべきの形の実数。 大きさ N の1次元配列。
ILL	整 数 型	出 力	ILL = 0: 正常終了。 ILL = K: 第 K 段で正規化中止。

引 数	型 と 種 類*	属 性	内 容
			一般の場合：第 K 行がすべて 0 である。 正値対称の場合：第 K 対角要素が正でない。 ILL = 30000： 引数エラー。

* MNRMBD(Q, C, B)の場合は実数型を倍精度実数型(4倍精度実数型, 複素数型, 倍精度複素数型)とする。
 MNMBSD(Q)の場合は実数型を倍精度(4倍精度)実数型とする。

(3) 使用例

1. MNRMBS の使用例

N = 1000 の 5 項行列 (NB = 5, LB = 3) 方程式を, MNRMBS で正規化した後で, LEQBDS により求解する。ただし, 対角要素は $a_{jj} = 5j$, 非対角要素はすべて 1 とし, 定数項はすべての解が 1 となるようにとるものとする。

```

DIMENSION A(7,1000),S(1000),X(1000),MAX(1000)
N=1000
KA=7
NB=5
LB=3
EPS=1.E-6
DO 10 J=1,N
DO 20 I=1,5
20 A(I,J)=1.0
   A(3,J)=J*5
   X(J)=A(3,J)+4.0
   IF(J.LE.2) X(J)=X(J)-FLOAT(3-J)
   IF(J.GE.N-1) X(J)=X(J)-FLOAT(J+2-N)
10 CONTINUE
CALL MNRMBS(A,KA,N,NB,LB,S,IND)
DO 25 I=1,N
25 X(I)=X(I)/S(I)
   IND=0
CALL LEQBDS(A,KA,N,NB,LB,MB,X,N,1,MAX,EPS,IND)
EM=0.0
DO 30 I=1,N
EM=AMAX1(ABS(X(I)-1.0),EM)
30 CONTINUE
WRITE(6,600) EM
600 FORMAT(10X,E11.3)
STOP
END

```

2. MNMBSS の使用例

N = 1000, NB = 5 の正値対称帯行列を係数とする方程式を, MNMBSS で正規化した後で, CHLBDS により求解する。ただし, 対角要素は $a_{jj} = 10j$, 非対角要素はすべて 1 とし, 定数項はすべての解が 1 となるようにとるものとする。

```

DIMENSION A(5,1000),S(1000),X(1000)
N=1000
KA=5
NB=5
EPS=1.E-6
DO 10 J=1,N
A(1,J)=J*10
DO 20 I=2,5
20 A(I,J)=1.0
   X(J)=A(1,J)+8.0
   IF(J.LE.4) X(J)=X(J)-FLOAT(5-J)
   IF(J.GE.N-3) X(J)=X(J)-FLOAT(J+4-N)
10 CONTINUE
CALL MNMBSS(A,KA,N,NB,S,IND)
DO 25 I=1,N

```

```

25 X(I)=X(I)/S(I)
   IND=0
   DET=0.
   CALL CHLBDS(A,KA,N,NB,X,N,1,DET,EPS,IND)
   DO 27 I=1,N
27  X(I)=X(I)/S(I)
   EM=0.0
   DO 30 I=1,N
   EM=AMAX1(ABS(X(I)-1.0),EM)
30  CONTINUE
   WRITE(6,600) EM
600 FORMAT(10X,E11.3)
   STOP
   END

```

(4) 備 考

1. 帯行列を係数とする連立一次方程式の解法に本ルーチンを適用する場合には、連立一次方程式ルーチンをコールする前に右辺定数ベクトルの各成分を対応する正規化因子で割る必要がある。解がえられた後、一般行列の場合は何もしなくてよいが、対称正値行列の場合は、解の各成分を対応する正規化因子で割らねばならない。(使用例参照)

MDETS/D/Q/C/B

Calculation of Determinants

行列式の計算

作成	二宮 市三 1977年4月
形式	サブルーチン 言語; FORTRAN サイズ; 45, 34, 45, 34, 35行

(1) 概要

与えられた行列の行列式を計算する。

(2) 使用法

CALL MDETS/D/Q/C/B (A, KA, N, EPS, D, ILL)

引数	型と種類	属性	内容
A*	実数型 2次元配列	入力	行列式を計算すべき行列。保存されない。
KA	整数型	入力	Aの配列宣言における第1添字の値。KA ≥ N
N	整数型	入力	Aの次数。N ≥ 2
EPS**	実数型	入力	行列の特異性の判定常数。ピボット要素の絶対値がこの常数より小さいとき D = 0 とする。EPS > 0
D**	実数型	出力	行列式の値が入る。
ILL	整数型	出力	ILL = 30000: KA, N, EPS に対する制限が破られたとき。それ以外は 0。

* MDETD(Q, C, B) の場合には A と D は倍精度実数型 (4倍精度実数型, 複素数型, 倍精度複素数型) である。

** MDETD(Q, C, B) の場合には EPS は倍精度実数型 (4倍精度実数型, 実数型, 倍精度実数型) である。

(3) 性能

精度は問題に依存する。LU-分解法 (ドゥーリトル法) を用い、積和の計算に倍精度演算が使われているので精度がよい。速度は、連立一次方程式を解くのとほぼ同じだけの計算時間を要する。

(4) 備考

1. EPSの標準的な値は、行列 A の要素の代表的な大きさを a とすると、MDETS (MDETD) に対しては、 $a \times 10^{-6}$ ($a \times 10^{-16}$)。
2. 連立一次方程式を解き、同時に係数行列式をも計算したいときは、LEQLUS, LEQLUD を用いることを推奨する。

LEQLUS/D/Q/C/B

Solution of Linear Equations by LU-Decomposition Method

LU-分解法による連立一次方程式の解法

作成	二宮 市三 1977年4月		
形式	サブルーチン	言語; FORTRAN (LEQLUS/Dはアセンブラ)	サイズ; 293, 241, 74, 75, 76行

(1) 概要

係数行列を共有する複数個の連立一次方程式を、ピボット選択のための行交換を伴うLU-分解法によって解く。すなわち、行列方程式 $AX = B$ の解 $X = A^{-1}B$ を求める。

(2) 使用法

CALL LEQLUS/D/Q/C/B(A, KA, N, X, KX, M, DET, MAX, EPS, IND)

引数	型と種類	属性	内容
A*	実数型 2次元配列	入出力	係数行列を入れる。本ルーチンで処理されて、係数行列のLU-分解成分が入る。これを保存しておけば、後に同じ係数の方程式を解く必要が生じたとき、LU-分解を省略できて計算時間の節約になる (MAX 及び IND の項参照)。
KA	整数型	入力	Aの配列宣言における第1添字の値。 $KA \geq N$
N	整数型	入力	方程式の元数、すなわち、Aの行数である。 $N \geq 2$
X*	実数型 2次元配列	入出力	方程式の複数個の右辺列を入れる。本ルーチンによって処理されて、その各列に対応する解ベクトルが入る。
KX	整数型	入力	Xの配列宣言における第1添字の値。 $KX \geq N$
M	整数型	入力	右辺列の数、すなわち、行列Xの列数である。 $M \geq 0$ $M = 0$ のときには、係数行列のLU-分解のみを行い、方程式は解かない。 $M = 1$ のときは、Xに対する実引数は1次元配列でもよい。
DET*	実数型	入出力	DET $\neq 0$ なる値を入れると係数行列式が出力として与えられる。DET = 0 とすると、このままである。
MAX	整数型 1次元配列	出力	N個の要素を持つ1次元配列名で、LU-分解での行交換に関する情報を入れておく。後になって、同じ係数の方程式を解く必要が生ずる場合には、この情報を保存しておく再利用できる。
EPS**	実数型	入力	係数行列の特異性の判定常数。ピボット要素の絶対値がこの常数より小さいとき特異であると判定して、計算を中断する。保存される。EPS > 0
IND	整数型	入出力	入力変数としては、IND = 0 ならば新たにLU-分解を行って

引 数	型 と 種 類	属 性	内 容
			<p>方程式を解くことを意味し、IND ≠ 0 ならば以前に行った LU-分解の結果を再利用して、ただちに方程式を解くことを意味する。この場合には、A と MAX とが以前のコールの時のまま保存されていなければならない。</p> <p>出力変数としては、正常に計算が終了したとき 0、引数に関する制限条件が破られたために全く計算が行われなかったとき 30000、EPS による判定により LU-分解が K 段目で中断されたとき K の値。</p>

* LEQLUD (Q, C, B) の場合には A, X, DET は倍精度実数型（4 倍精度実数型、複素数型、倍精度複素数型）である。

** LEQLUD (Q, C, B) の場合には EPS は倍精度実数型（4 倍精度実数型、実数型、倍精度実数型）である。

(3) 性 能

LEQLUS/D はアセンブラで書かれているので、無駄がなく速度が速い。また、LEQLUS には倍精度演算が部分的に用いられているので、丸めの誤差が少なく、精度がよい。

(4) 備 考

1. 本ルーチンは、複数右辺列の同時処理、行列式の計算、LU-分解成分の再利用などの機能を有しているため、連立一次方程式の解、行列式及び逆行列のルーチンの働きを兼ねることができる。特に、LU-分解の再利用の機能が重要で、これを活用すれば、逆行列を計算する必要はほとんどなくなる。
2. 方程式の係数の絶対値の間に大きな差があるときは、MNORMS, MNORMD などによって、あらかじめ係数行列を正規化することが精度を確保する上で望ましい。
3. EPS の標準的な値は、係数行列の要素の絶対値の代表的な値を a とすると、LEQLUS(LEQLUD, LEQLUQ) に対しては $a \times 10^{-6}$ ($a \times 10^{-16}$, $a \times 10^{-30}$) が適当である。
4. 係数行列が対称正値の場合には、その場合の専門ルーチン CHOLFS, CHOLFD 等を利用する方が賢明である。
5. 引数 DET, IND は入出力両方に用いられるので、これらに対する実引数として定数を使ってはいけない。

GAUELS/D/Q/C/B

Solution of Linear Equations by LU-Decomposition

LU分解法による連立一次方程式の解法

作成	二宮 市三	1977年4月
形式	サブルーチン	言語; FORTRAN (GAUELS/Dはアセンブラ) サイズ; 180, 183, 53, 43, 44行

(1) 概要

係数行列を共有する複数個の連立一次方程式を、ピボット選択のための行交換を伴うガウスの消去法の一変形ドゥーリトル法によって解く。すなわち、行列方程式 $AX = B$ の解 $X = A^{-1}B$ を求める。

(2) 使用法

CALL GAUELS/D/Q/C/B (A, KA, N, M, EPS, ILL)

引数	型と種類	属性	内容
A*	実数型 2次元配列	入出力	係数行列の右に右辺列を付加した拡大行列を入れる。右辺列の位置に対応する解ベクトルが入る。
KA	整数型	入力	Aの配列宣言における第1添字の値。KA ≥ N
N	整数型	入力	Aの行数、すなわち、方程式の元数。N ≥ 2
M	整数型	入力	Aの列数、すなわち、方程式の元数に、同時に解くべき方程式の数を加えたもの。M > N
EPS**	実数型	入力	係数行列の特異性の判定常数。ピボット要素の絶対値がこの常数より小さいときは特異であると判定して、計算を中断する。EPS > 0
ILL	整数型	出力	ILL = 0 : 正常終了。 ILL = 30000 : KA, N, M, EPSに関する制限が破られたとき。 特異と判定されたときはピボット要素の番号。

* GAUELD (Q, C, B) の場合には A は倍精度実数型 (4倍精度実数型, 複素数型, 倍精度複素数型) である。

** GAUELD (Q, C, B) の場合には EPS は倍精度実数型 (4倍精度実数型, 実数型, 倍精度実数型) である。

(3) 備考

- 方程式の係数の絶対値の間に大きな差があるときは、MNORMS, MNORMD 等によってあらかじめ正規化することが、精度を保つ上で望ましい。
- EPS の標準的な値としては、係数行列の要素の絶対値の代表的な値を a とすると、GAUELS, (GAUELD, GAUELQ) に対しては $a \times 10^{-6}$ ($a \times 10^{-16}$, $a \times 10^{-30}$) が適当である。
- 連立一次方程式解法ルーチンとしては、特別の理由がなければ、精度がよく種々の機能をそなえた LEQLUS, LEQLUD 等の使用を推奨する。

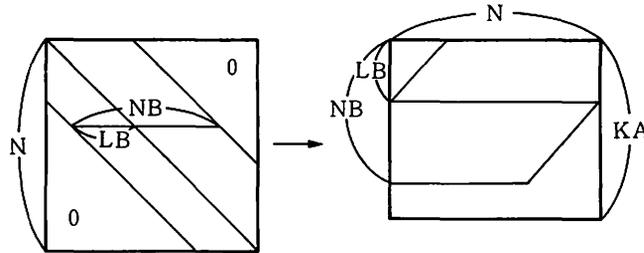
LEQBDS / D / Q / C / B

Solution of Linear Equations with Band Matrix of Coefficients by Gaussian Elimination
 ガウスの消去法による帯行列係数連立一次方程式の解法

作成	二宮 市三 1978年9月
形式	サブルーチン 言語; FORTRAN サイズ; 80, 80, 80, 81, 80行

(1) 概要

帯行列 A を係数行列とし、複数個の右辺列 B を持つ連立一次方程式 $AX = B$ の解 $X = A^{-1}B$ をピボット選択のための行交換を伴うガウスの消去法で求める。消去法によってえられる A の LU-分解成分の再利用の機能を持つ。



(2) 使用法

CALL LEQBDS/D/Q/C/B (A, KA, N, NB, LB, MB, X, KX, NX, MAX, EPS, IND)

引数	型と種類	属性	内容
A*	実数型 2次元配列	入出力	係数帯行列を長方形化して入れる。すなわち元の行列の I, J 要素を A (J-I+LB, I) に入れる (図示参照)。本ルーチンで処理されて LU-分解成分が入る。
KA	整数型	入力	A の配列宣言における第 1 添字の値。KA ≥ NB
N	整数型	入力	方程式の元数, すなわち A の列数である。N ≥ NB
NB	整数型	入力	全帯幅 (図示参照)。A の行数である。NB > LB
LB	整数型	入力	左帯幅 (図示参照)。LB ≥ 2
MB	整数型	出力	処理後の A の行数。MB ≤ KA でなければならない。
X*	実数型 2次元配列	入出力	右辺列を入れると、対応する場所に解ベクトルが出力される。
KX	整数型	入力	X の配列宣言における第 1 添字の値。KX ≥ N
NX	整数型	入力	X の列数。NX ≤ 0 のときは A の処理だけが行われる。
MAX	整数型 1次元配列	出力	N 個の要素をもつ 1 次元配列で、行交換に関する情報を入れておく。LU-成分の再利用のとき用いられる。
EPS**	実数型	入力	係数行列の特異性の判定常数。ピボット要素の絶対値が EPS より小さいとき特異であると判定して、計算を中断する。EPS > 0

引 数	型 と 種 類	属 性	内 容
IND	整 数 型	入 出 力	<p>入力としてはIND=0のとき新たに消去法を行って方程式を解くことを意味し、IND≠0のときは以前にえられたLU分解成分を再利用して、消去法を行わないで直ちに解を求めることを意味する。この場合にはAとMAXとが以前のコールのときのまま保存されていなければならない。</p> <p>出力としては、正常に計算が終了したとき0、引数に関する制限条件が破られたため全く計算が行われなかったとき30000、特異と判定されて消去がK段目で中止されたときKの値。</p>

* LEQBDD (Q, C, B) の場合には A, X を倍精度実数型 (4 倍精度実数型, 複素数型, 倍精度複素数型) とする。

** LEQBDD (Q, C, B) の場合には EPS を倍精度実数型 (4 倍精度実数型, 実数型, 倍精度実数型) とする。

(3) 使用例

1000元(N = 1000), 全帯幅 7 (NB = 7), 左帯幅 3 (LB = 3) の方程式についてのテストを行うためのプログラムを下に示す。

```

C      TEST FOR LEQBDS
      DIMENSION A(10,1000),MAX(1000),X(1000)
      N=1000
      NB=7
      LB=3
      KA=10
      EPS=1.0E-6
      NX=1
      C=1.
      DO 10 I=1,N
        A(1,I)=C
        A(2,I)=-C
        A(3,I)=0.
        A(4,I)=0.
        A(5,I)=C
        A(6,I)=0.
10     A(7,I)=-C
        DO 20 L=1,2
          IND=L-1
          DO 11 I=1,N
11      X(I)=0.0
          X(2)=-C*C
          X(N-2)=C*C
          X(N-3)=C*C
          CALL CLOCKM(K0)
          CALL LEQBDS(A,KA,N,NB,LB,MB,X,N,NX,MAX,EPS,IND)
          CALL CLOCKM(K1)
          KO=K1-K0
          AM=0.0
          DO 12 I=1,N
            AA=ABS(X(I)-C)
            IF(AA.LE.AM) GO TO 12
            AM=AA
            MM=I
12     CONTINUE
        WRITE(6,600) IND,KO,AM,MM,MB
600    FORMAT(//10X,'ILL =',I6,5X,'TIME =',I6,5X,'ERR =',1PE10.2,5X,
          *'IMAX =',I6,5X,'MB =',I6/)
20     CONTINUE
      STOP
      END

```

(4) 備考

1. 本ルーチンは、複数右辺列の同時処理、LU-分解成分の再利用の機能を有しているので、連立一次方程式解法ルーチンと逆行列ルーチンの働きを兼ねることができる。特に、LU-分解成分の再利用が重要で、これを活用すれば逆行列を計算する必要はほとんどなくなる。
2. ピボット選択のために、必要に応じて行の交換を行っているので、係数行列の列数は一般に始めの状態よりも増加する。したがって、 $MB \leq KA$ となるように KA をとっておかなければならない。最悪の場合でも $MB = NB + LB - 1$ であるから $KA = NB + LB - 1$ とっておけば安全である。
3. 方程式の係数の大きさに、はなはだしい差があるときには、各方程式の係数の最大絶対値が1の程度になるように、あらかじめ係数行列を正規化することが望ましい。
4. EPS の標準的な値は、係数行列の要素の代表的な大きさを a とすると、LEQBDS (LEQBDD) に対しては $a \times 10^{-6}$ ($a \times 10^{-16}$) 程度が適当である。
5. 係数行列が正値対称の場合には、その場合の専門ルーチン CHLBDS 等を用いるのが賢明である。
6. 引数 IND は入出力両方に用いられるので、これに対する実引数として定数を使ってはならない。

BUNCHS / D

Solution of Linear Equations with Symmetric Matrix of Coefficients by Bunch's Method バンチの方法による対称行列係数連立一次方程式の解法

作成	二宮 市三 1981年4月
形式	サブルーチン 言語; FORTRAN サイズ; 200, 200行

(1) 概要

BUNCHS (D) は必ずしも正値でない対称行列 A を係数とし、複数個の右辺列 B を持つ連立一次方程式 $AX = B$ の解 $X = A^{-1}B$ をバンチの $U^T D U$ 分解法で求めるための単（倍）精度用のサブルーチン。 $U^T D U$ 分解成分の再利用の機能を持つ。

(2) 使用法

CALL BUNCHS/D (A, KA, N, X, KX, M, CHG, EPS, IND)

引数	型と種類*	属性	内容
A	実数型 2次元配列	入出力	係数行列の対角線を含む右上半分を入力する。本ルーチンで処理されてバンチの分解成分が出力される。左下半分は保存される。
KA	整数型	入力	Aの整合寸法（配列宣言における第1添字の値）。 $KA \geq N$
N	整数型	入力	方程式の元数。 $N \geq 1$
X	実数型 2次元配列	入出力	右辺列を入力する。対応する場所に解ベクトルが出力される。
KX	整数型	入力	Xの整合寸法。 $KX \geq N$
M	整数型	入力	Xの列数。 $M \leq 0$ のときはAの分解だけが行われる。
CHG	実数型 1次元配列	出力	大きさN以上の1次元配列。ピボッティングの情報と 2×2 対角ブロックの行列式が出力される。
EPS	実数型	入力	分解の途中でピボット要素の大きさが $\ A\ \cdot EPS$ よりも小さくなったとき、係数行列が特異であると判定して計算を中断する。 $EPS \leq 0.0$ を与えると標準値 u が用いられる。ただし、 $u = 2^{-20}$ (単精度), $u = 2^{-52}$ (倍精度)。
IND	整数型	入出力	入力引数としては次の意味を持つ。 IND = 0: 新たにバンチの分解から始めて方程式を解く。 IND \neq 0: 直前に計算した分解成分を再利用し、方程式の求解だけを行う。この場合AとCHGの内容が保存されていないなければならない。 出力引数としては次の意味を持つ。 IND = 0: 正常終了。 IND = K: 分解のK段目で特異であると判定された。 IND = 30000: 入力引数が制限を犯した。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 計算法

係数行列 A が対称正値の場合は、単位上三角行列 U と対角行列 D とによって、改訂コレスキー分解

$$A = U^T D U$$

が可能であるが、 A が対称であっても正値でない場合には、この分解は一般に不可能である。しかし D を、 2×2 の小行列をも対角ブロック要素として許すブロック対角行列とすれば、上の分解は可能である。バンチは、適当に行と列を交換して、分解 $A = U^T D U$ を、数値的に安定に行う算法を考案した。^{1),2)} 本ルーチンは、バンチの算法 A に基づいている。

この分解を用いれば $AX = B$ の解 $X = A^{-1}B$ は、前進代入 $Y = U^{-T}B$ と後退代入 $X = U^{-1}D^{-1}Y$ によって求められる。

(4) 備考

1. EPS の標準的な値は、BUNCHS(D) の場合 10^{-6} (10^{-16}) である。EPS ≤ 0.0 を与えれば標準値 2^{-20} (2^{-52}) が用いられる。
2. IND は入出力引数であるから、実引数として定数を用いてはいけない。
3. 同じ係数行列に対して、右辺列のみを変えて反復的に求解を行う場合には、本ルーチンの持つ分解成分の再利用の機能が極めて有用である。逆行列による方法に比べて、精度、速度及び記憶容量のいずれの点においても優れている。
4. 右辺列の数 M が 1 のときは、 X に相当する実引数は 1 次元配列でもよい。ただし、 $KX \geq N$ の条件は成り立つようにしておく必要がある。

参考文献

- 1) J.R. Bunch et al.; "Decomposition of a Symmetric Matrix" Number. Math., Bd.27, pp.95-109 (1976).
- 2) J.R. Bunch et al.; "Some Stable Methods for Calculating Inertia and Solving Symmetric Linear Systems", Math. Comp., Vol.31, No.137, pp.163-179 (1977).

BUNCBS / D

Solution of Linear Equations with Symmetric Band Matrix of Coefficients by Bunch's Method

バンチの方法による対称バンド行列係数連立一次方程式の解法

作成	二宮 市三 1981年4月
形式	サブルーチン 言語 ; FORTRAN サイズ ; 200, 200行

(1) 概要

BUNCBS (D) は、必ずしも正値でない対称帯行列 A を係数とし、複数の右辺列 B を持つ連立一次方程式 $AX = B$ の解 $X = A^{-1}B$ を、バンチの LDL^T 分解法で求めるための単 (倍) 精度用のサブルーチン。 LDL^T 分解成分の再利用の機能を持つ。

(2) 使用法

CALL BUNCBS/D (A, KA, N, NB, X, KX, M, CHG, EPS, IW, IND)

引数	型と種類*	属性	内容
A	実数型 2次元配列	入出力	対称帯行列の対角線を含む左下半分を図のように長方形化して入力する。すなわち行列の I, J 要素を A (I - J + 1, J) に入れる。本ルーチンで処理されて、バンチの分解成分が出力される。
KA	整数型	入力	A の整合寸法 (A の配列宣言における第 1 添字の値)。ピボットングのために A の帯幅は一般に増大するのでその分を見越して、KA は十分大きく取っておく必要がある。KA ≥ NB
N	整数型	入力	方程式の元数 (A の列数)。N ≥ 1
NB	整数型	入出力	A の半帯幅 (行数) を入力する。処理後の半帯幅が出力される。NB ≥ 2
X	実数型 2次元配列	入出力	右辺列を入力する。対応する場所に解ベクトルが出力される。
KX	整数型	入力	X の整合寸法。KX ≥ N
M	整数型	入力	X の列数。M ≤ 0 のときは A の分解だけが行われる。
CHG	実数型 1次元配列	出力	大きさ N 以上の 1 次元配列。ピボットングの情報と 2 × 2 対角ブロックの行列式が出力される。
EPS	実数型	入力	分解の途中でピボット要素の大きさが A · EPS よりも小さくなったとき、係数行列が特異であると判定して計算を中断する。EPS ≤ 0.0 を与えると標準値 u が用いられる。ただし、u = 2 ⁻²⁰ (単精度), u = 2 ⁻⁵² (倍精度)。
IW	整数型 1次元配列	作業領域	大きさ N 以上の 1 次元配列。
IND	整数型	入出力	入力引数としては次の意味を持つ。 IND = 0: 新たにバンチの分解から始めて方程式を解く。 IND ≠ 0: 直前に計算した分解成分を再利用し、方程式

引 数	型 と 種 類*	属 性	内 容
			の求解だけ行う。この場合AとCHGの内容が保存されていなければならない。 出力引数としては次の意味を持つ。 IND = 0 : 正常終了。 IND = K : 分解のK段目で特異と判定されたか又は帯幅がKAを越えた。 IND = 30000 : 入力引数が制限を犯した。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 計算法

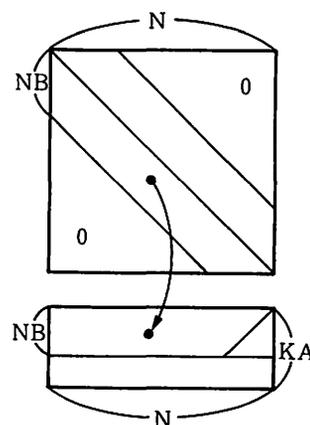
係数行列 A が対称正値の場合は、単位下三角行列 L と対角行列 D によって、改訂コレスキー分解

$$A = LDL^T$$

が可能であるが、 A が対称であっても正値でない場合には、この分解は一般に不可能である。しかし D を、 2×2 の小行列をも対角ブロック要素として許すブロック対角行列とすれば、上の分解は可能である。バンチは、適当に行と列を交換して、分解 $A = LDL^T$ を、数値的に安定に行う算法を考案した。^{1), 2)}

本ルーチンは、バンチの算法 D に基づいている。

この分解を用いれば $AX = B$ の解 $X = A^{-1}B$ は、前進代入 $Y = L^{-1}B$ と後退代入 $X = L^{-T}D^{-1}Y$ によって求められる。



(4) 備 考

1. EPS の標準的な値は、BUNCBS (D) の場合 10^{-6} (10^{-16}) である。EPS ≤ 0.0 を与えれば標準値 2^{-20} (2^{-52}) が用いられる。
2. IND は入出力引数であるから、実引数として定数を用いてはいけない。
3. 同じ係数行列に対して、右辺列のみを変えて反復的に求解を行う場合には、本ルーチンの持つ分解成分の再利用の機能が極めて有用である。逆行列による方法に比べて、精度、速度及び記憶容量のいずれの点においても優れている。
4. 右辺列の数 M が 1 のときは、 X に相当する実引数は 1 次元配列でもよい。ただし、 $KX \geq N$ の条件は成り立つようにしておく必要がある。

参 考 文 献

- 1) J.R. Bunch et al.; "Decomposition of a Symmetric Matrix" Number. Math., Bd.27, pp.95-109 (1976).
- 2) J.R. Bunch et al.; "Some Stable Methods for Calculating Inertia and Solving Symmetric Linear Systems", Math. Comp., Vol.31, No.137, pp.163-179 (1977).

CHOLFS / D / Q

MCHLFS / D / Q

Solution of Symmetric Positive Definite Linear Equations by Cholesky and Modified Cholesky Method (Full Matrix)

コレスキー法及び改訂コレスキー法による対称正値連立一次方程式の解法（密行列）

作成	二宮 市三 1981年4月
形式	サブルーチン 言語; アセンブラ (CHOLFQ, MCHLFQ は FORTRAN) サイズ; 207, 217, 49, 179, 176, 51行

(1) 概要

CHOLFS (D, Q) (MCHLFS(D, Q)) は対称正値行列 A を係数行列とし、複数個の右辺列 B を持つ方程式 $AX = B$ の解 $X = A^{-1}B$ を (改訂) コレスキー分解法で求めるための単 (倍, 四倍) 精度用のサブルーチン。(改訂) コレスキー分解成分の再利用の機能をもつ。

(2) 使用法

CALL { CHOLFS/D/Q
MCHLFS/D/Q } (A, KA, N, X, KX, M, DET, EPS, IND)

引 数	型 と 種 類*	属 性	内 容
A	実 数 型 2 次 元 配 列	入 出 力	係数行列の対角線を含む右上半分を入力する。本ルーチンで処理されて (改訂) コレスキー分解成分が出力される。左下半分は保存される。
KA	整 数 型	入 力	Aの整合寸法 (配列宣言における第1添字の値)。KA ≥ N
N	整 数 型	入 力	方程式の元数。N ≥ 1
X	実 数 型 2 次 元 配 列	入 出 力	右辺列を入力する。対応する場所に解ベクトルが出力される。
KX	整 数 型	入 力	Xの整合寸法。KX ≥ N
M	整 数 型	入 力	Xの列数。M ≤ 0 のときは、(改訂) コレスキー分解だけが行われる。
DET	実 数 型	入 出 力	DET ≠ 0.0 を入力すると係数行列式が出力される。 DET = 0.0 を入力するとそのまま 0.0 が出力される。
EPS	実 数 型	入 力	係数行列の正値性を判定するための定数。コレスキー分解の途中で対角要素が EPS よりも小さくなったとき、正値でないとして計算を中断する。
IND	整 数 型	入 出 力	入力引数としては次の意味を持つ。 IND = 0: 新たにコレスキー分解から始めて方程式を解く。 IND ≠ 0: 直前に計算したコレスキー分解成分を再利用して方程式の求解だけを行う。 出力引数としては次の意味を持つ。

引数	型と種類*	属性	内容
			IND=0: 正常に計算が行われた。 IND=K: コレスキー分解のK段目で対角成分がEPSよりも小さくなったので計算を中止した。 IND=30000: 入力引数が制限を犯した。

* 倍（四倍）精度用のサブルーチンの場合は、実数型をすべて倍（四倍）精度実数型とする。

(3) 計算法

1. コレスキー分解法

上三角行列 U とその転置行列 U^T とによって

$$A = U^T U$$

と分解する。解 $X = A^{-1}B$ は前進代入 $Y = U^{-T}B$ と後退代入 $X = U^{-1}Y$ によって求められる。

2. 改訂コレスキー分解法

単位上三角行列 U ，その転置行列 U^T 及び対角行列 D によって

$$A = U^T D U$$

と分解する。解 $X = A^{-1}B$ は，前進代入 $Y = U^{-T}B$ と後退代入 $X = U^{-1}D^{-1}Y$ によって求められる。

3. CHOLFS, MCHLFS ではすべての積和計算に部分的倍精度計算が用いられているので，丸めの誤差の影響が少ない。

(4) 備考

1. EPS の標準的な値としては，係数行列の要素の代表的な大きさを a とすると， $\{\text{CHOLFS}(D,Q)\}$ $\{\text{MCHLFS}(D,Q)\}$ に対しては， $10^{-6}a$ ($10^{-16}a$, $10^{-30}a$) 程度が適当である。
2. DET, IND は入出力引数であるから，実引数として定数を用いてはいけない。
3. 同じ係数行列に対して，右辺列のみを変えて反復的に求解を行う場合には，本ルーチンの持つ（改訂）コレスキー分解成分の再利用の機能が極めて有用である。逆行列による方法に比べて，記憶容量，精度及び速度のいずれの点においても優れている。
4. 右辺列の数 M が 1 のときは， X に相当する実引数は 1 次元配列でもよい。ただし， $KX \geq N$ でなければならない。

CHOLCS / D / Q

MCHLCS / D / Q

Solution of Symmetric Positive Definite Linear Equations by Cholesky and Modified Cholesky Method (Full Matrix, Compact Mode)

コレスキー法及び改訂コレスキー法による対称正値連立一次方程式の解法（密行列，圧縮表現）

作成	二宮 市三 1981年4月
形式	サブルーチン 言語；アセンブラ (CHOLCQ, MCHLCQ は FORTRAN) サイズ；203, 217, 60, 180, 179, 96行

(1) 概要

CHOLCS (D, Q) (MCHLCS (D, Q)) は圧縮表現された対称正値行列 A を係数行列とし，複数個の右辺列 B を持つ方程式 $AX = B$ の解 $X = A^{-1}B$ を（改訂）コレスキー分解法で求めるための単（倍，四倍）精度用のサブルーチン。（改訂）コレスキー分解成分の再利用の機能をもつ。

(2) 使用法

CALL { CHOLCS/D/Q
MCHLCS/D/Q } (A, N, X, KX, M, DET, EPS, IND)

引 数	型 と 種 類*	属 性	内 容
A	実 数 型 1 次 元 配 列	入 出 力	係数行列の対角線を含む右上半分を列方向に1次元化して入力する。すなわち元の行列の I, J 要素を A ((J-1)*J/2+I) に入れる。本ルーチンで処理されて（改訂）コレスキー分解成分が出力される。
N	整 数 型	入 力	方程式の元数。N ≥ 1
X	実 数 型 2 次 元 配 列	入 出 力	右辺列を入力する。対応する場所に解ベクトルが出力される。
KX	整 数 型	入 力	Xの整合寸法。KX ≥ N
M	整 数 型	入 力	Xの列数。M ≤ 0 のときは，（改訂）コレスキー分解だけが行われる。
DET	実 数 型	入 出 力	DET ≠ 0.0 を入力すると係数行列式が出力される。 DET = 0.0 を入力するとそのまま 0.0 が出力される。
EPS	実 数 型	入 力	係数行列の正値性を判定するための定数。コレスキー分解の途中で対角要素が EPS よりも小さくなったとき，正値でないとして計算を中断する。
IND	整 数 型	入 出 力	入力引数としては次の意味を持つ。 IND = 0： 新たにコレスキー分解から始めて方程式を解く。 IND ≠ 0： 直前に計算したコレスキー分解成分を再利用して方程式の求解だけを行う。

引 数	型 と 種 類*	属 性	内 容
			出力引数としては次の意味を持つ。 IND = 0 : 正常に計算が行われた。 IND = K : コレスキー分解のK段目で対角成分が EPS よりも小さくなったので計算を中止した。 IND = 30000 : 入力引数が制限を犯した。

* 倍（四倍）精度用のサブルーチンの場合は、実数型をすべて倍（四倍）精度実数型とする。

(3) 計算法

1. コレスキー分解法

上三角行列 U とその転置行列 U^T とによって

$$A = U^T U$$

と分解する。解 $X = A^{-1}B$ は前進代入 $Y = U^{-T}B$ と後退代入 $X = U^{-1}Y$ によって求められる。

2. 改訂コレスキー分解法

単位上三角行列 U 、その転置行列 U^T 及び対角行列 D によって

$$A = U^T D U$$

と分解する。解 $X = A^{-1}B$ は、前進代入 $Y = U^{-T}B$ と後退代入 $X = U^{-1}D^{-1}Y$ によって求められる。

3. CHOLCS, MCHLCS ではすべての積和計算に部分的倍精度計算が用いられているので、丸めの誤差の影響が少ない。

(4) 備 考

1. EPS の標準的な値としては、係数行列の要素の代表的な大きさを a とすると、 $\left\{ \begin{array}{l} \text{CHOLCS}(D, Q) \\ \text{MCHLCS}(D, Q) \end{array} \right\}$ に対しては、 $10^{-6}a$ ($10^{-16}a$, $10^{-30}a$) 程度が適当である。
2. DET, IND は入出力引数であるから、実引数として定数を用いてはいけない。
3. 同じ係数行列に対して、右辺列のみを変えて反復的に求解を行う場合には、本ルーチンの持つ（改訂）コレスキー分解成分の再利用の機能が極めて有用である。逆行列による方法に比べて、記憶容量、精度及び速度のいずれの点においても優れている。
4. 右辺列の数 M が 1 のときは、 X に相当する実引数は 1 次元配列でもよい。ただし、 $KX \geq N$ でなければならない。

CHOLSK / CHOLSD

Solution of Symmetric Positive Definite Linear Equations by Cholesky Method

コレスキー法による対称正値な連立一次方程式の解法

作 成	二宮 市三	1977年4月
形 式	サブルーチン 言語; FORTRAN サイズ; 43, 43行	

(1) 概 要

対称正値な係数行列を共有する複数個の連立一次方程式を、コレスキー分解法によって解く。すなわち、対称正値な行列 A に対する行列方程式 $AX = B$ の解 $X = A^{-1}B$ を求める。

(2) 使用法

CALL CHOLSK (A, KA, N, M, EPS, IND)

CALL CHOLSD (A, KA, N, M, EPS, IND)

引 数	型 と 種 類*	属 性	内 容
A	実 数 型 2 次 元 配 列	入 出 力	対称正値な係数行列の右に複数個の右辺列を付加した拡大行列を入れる。係数行列の部分は、対角線を含んで右上半分だけ入れてあればよい。本ルーチンで処理されて同じ場所に係数行列のコレスキー分解成分が入り、右辺列部分には対応する解ベクトルが入る。係数行列の左下半分は保存される。コレスキー分解成分は、これを保存しておく、同じ係数行列の別の方程式を解く場合、コレスキー分解が省略できて計算時間の節約となる。
KA	整 数 型	入 力	Aの配列宣言における第1添字の値。KA ≥ N
N	整 数 型	入 力	Aの行数。方程式の元数でもある。N ≥ 2
M	整 数 型	入 力	Aの列数。方程式の元数と右辺列の数との和。M ≥ N。M = Nのときは係数行列のコレスキー分解のみが行われる。
EPS	実 数 型	入 力	Aの正値性の判定常数。Aのピボット要素がこのEPSより小さいとき正値でないと判定して、計算を中断する。 EPS > 0
IND	整 数 型	入 出 力	入力として、コレスキー分解成分の再利用を行うかどうかを指示する。すなわち、IND = 0ならば再利用は行わず、通常の計算を行うことを意味し、IND ≠ 0ならば再利用を行うことを意味する。 出力としては、ルーチン内での計算の状況を示す。KA, N, M, EPSに関する制限が破られたとき30000, EPSによる判定で正値でないと判定されたときはそのピボットの番号、正常に計算が行われたとき0。

* CHOLSD の場合には、実数型はすべて倍精度実数型とする。

(3) 備 考

1. 引数 IND は入出力両方に使われるので、これに対する実引数として定数を書いてはいけない。

2. 同じ係数行列で右辺だけが異なる方程式を多数解く必要がある場合には、本ルーチンのコレスキー分解成分再利用の機能を用いると、計算時間が節約できて合理的である。
3. EPS の標準的な値としては、行列要素の代表的な大きさを a とするとき、CHOLSK (CHOLSD) に対しては $a \times 10^{-6}$ ($a \times 10^{-16}$) が適当である。
4. 本ルーチンと同様で、更に多くの機能を持つ CHOLFS、CHOLFD や、帯行列用の同様なルーチン CHLBDS、CHLBDがあるので、適宜使用されたい。

CHLBDS / D / Q

MCHLBS / D / Q

Solution of Symmetric Positive Definite Linear Equations by Cholesky and Modified Cholesky Method (Band Matrix)

コレスキー法及び改訂コレスキー法による対称正値連立一次方程式の解法（帯行列）

作成	二宮 市三 1981年4月
形式	サブルーチン 言語; アセンブラ (CHLBDQ, MCHLBQ は FORTRAN) サイズ; 233, 239, 64, 202, 199, 71行

(1) 概要

CHLBDS (D, Q) (MCHLBS(D, Q)) は対称正値帯行列 A を係数行列とし、複数個の右辺列 B を持つ方程式 $AX = B$ の解 $X = A^{-1}B$ を (改訂) コレスキー分解法で求めるための単 (倍, 四倍) 精度用のサブルーチン。コレスキー分解成分の再利用の機能を有する。

(2) 使用法

CALL { CHLBDS/D/Q
MCHLBS/D/Q } (A, KA, N, NB, X, KX, M, DET, EPS, IND)

引 数	型 と 種 類*	属 性	内 容
A	実 数 型 2 次 元 配 列	入 出 力	係数行列の対角線を含んで左下半分の帯の部分を実数型に整形して入力する。すなわち行列の I, J 要素を A (I-J+1, J) に入れる。本ルーチンで処理されて (改訂) コレスキー分解成分が出力される。図参照。
KA	整 数 型	入 力	A の整合寸法 (配列宣言における第 1 添字の値)。KA ≥ NB
N	整 数 型	入 力	方程式の元数 (A の列数)。N ≥ 1
NB	整 数 型	入 力	帯幅 (A の行数)。1 ≤ NB ≤ N
X	実 数 型 2 次 元 配 列	入 出 力	右辺列を入力する。対応する場所に解ベクトルが出力される。
KX	整 数 型	入 力	X の整合寸法。KX ≥ N
M	整 数 型	入 力	X の列数。M ≤ 0 のときは、(改訂) コレスキー分解だけが行われる。
DET	実 数 型	入 出 力	DET ≠ 0.0 を入力すると係数行列式が出力される。 DET = 0.0 を入力するとそのまま 0.0 が出力される。
EPS	実 数 型	入 力	係数行列の正値性を判定するための定数。コレスキー分解の途中で対角要素が EPS よりも小さくなったとき、正値でないとして計算を中段する。EPS > 0
IND	整 数 型	入 出 力	入力引数としては次の意味を持つ。 IND = 0 : 新たにコレスキー分解から始めて方程式を解く。 IND ≠ 0 : 直前に計算したコレスキー分解成分を再利用

引数	型と種類*	属性	内容
			<p>し、方程式の求解だけを行う。</p> <p>出力引数としては次の意味を持つ。</p> <p>IND=0： 正常に計算が行われた。</p> <p>IND=K： コレスキー分解のK目目で対角成分がEPSより小さくなったので計算を中止した。</p> <p>IND=30000： 入力引数が制限を犯した。</p>

* 倍（四倍）精度用のサブルーチンの場合は、実数型をすべて倍（四倍）精度実数型とする。

(3) 計算法

1. コレスキー分解法

下三角行列 L とその転置 L^T によって

$$A = LL^T$$

と分解する。解 $X = A^{-1}B$ は前進代入 $Y = L^{-1}B$ と後退代入 $X = L^{-T}Y$ によって求められる。

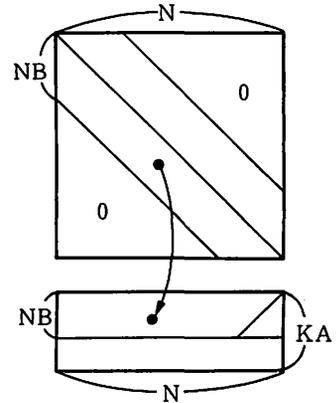
2. 改訂コレスキー分解法

単位下三角行列 L 、その転置 L^T 及び対角行列 D によって

$$A = LDL^T$$

と分解する。解 $X = A^{-1}B$ は前進代入 $Y = L^{-1}B$ と後退代入 $X = L^{-T} \cdot D^{-1}Y$ によって求められる。

3. CHLBDS, MCHLBS ではすべての積和計算に部分的倍精度計算が用いられているので、丸めの誤差の影響が少ない。



(4) 備考

- EPS の標準的な値としては、係数行列の要素の代表的な大きさを a とすると、 $\{MCHLBS(D, Q)\}$ に対しては、 $10^{-6}a$ ($10^{-16}a$, $10^{-30}a$) 程度が適当である。
- DET, IND は入出力引数であるから、実引数として定数を用いてはいけない。
- 同じ係数行列に対して、右辺列のみを変えて反復的に求解を行う場合には、本ルーチンの持つ(改訂)コレスキー分解成分の再利用の機能が極めて有用である。逆行列による方法に比べて、記憶容量、精度及び速度のいずれの点においても優れている。
- 右辺列の数 M が 1 のときは、 X に相当する実引数は 1 次元配列でもよい。ただし、 $KX \geq N$ でなければならない。

CHLVBS / D

Solution of Symmetric Positive Definite Linear Equations by Cholesky Method (Band Matrices with Variable Bandwidth, Compact Mode)

コレスキー法による対称正値な連立一次方程式の解法 (可変帯幅帯行列, 圧縮表現)

作成	二宮 市三	1977年4月
形式	サブルーチン 言語; FORTRAN サイズ; 94, 94行	

(1) 概要

対称正値可変帯幅帯行列 A を係数行列とし, 複数個の右辺列 B を持つ方程式 $AX = B$ の解 $X = A^{-1}B$ をコレスキー分解法で求める。すなわち, 上三角行列 U とその転置 U^T によって $A = U^T U$ と分解し, $X = U^{-1}(U^{-T}B)$ と計算する。 $C = U^{-T}B$ は下三角行列 U^T に対する前進代入法で, $X = U^{-1}C$ は上三角行列 U に対する後退代入法で計算する。また, コレスキー分解成分の再利用の機能を持っている。

(2) 使用法

CALL CHLVBS/D (A, NB, X, KX, N, M, EPS, IND)

引数	型と種類*	属性	内容
A	実数型 1次元配列	入出力	正値対称の係数行列の対角線を含んで右上半分の純粋に0である部分を除いたものを, 図示の要領で一列に並べる。本ルーチンで処理されて, コレスキー分解成分が同じ構造で入る。
NB	整数型 1次元配列	入力	N個の要素を持つ1次元配列。係数行列の右上半分の各列のバンド幅を入れる (図示参照)。
X	実数型 2次元配列	入出力	右辺列を入れる。対応する場所に解ベクトルが出力される。
KX	整数型	入力	Xの配列宣言における第1添字の値。 $KX \geq N$
N	整数型	入力	方程式の元数, すなわち, Xの行数である。 $N \geq 2$
M	整数型	入力	Xの列数。 $M \geq 0$ $M = 0$ のときは, コレスキー分解だけが行われる。 $M = 1$ のときは, Xに対する実引数は1次元配列でもよい。
EPS	実数型	入力	係数行列の正値性を判定するための常数。コレスキー分解の途中で対角要素がEPSより小さくなったとき正値でないとして, 計算を中断する。 $EPS > 0$
IND	整数型	入出力	入力引数として次の意味を持つ。 IND = 0: 新たにコレスキー分解を行って方程式を解く。 IND \neq 0: コレスキー分解成分の再利用により方程式を解く。 出力引数としては次の意味を持つ。 IND = 0: 正常に計算が行われた。 IND = 30000: 入力引数に関する制限が破られた。 IND = K: K段目で対角要素がEPSより小となったので, 計算を中断した。

引数	型と種類*	属性	内容
			この引数は入出力両用であるから、実引数として定数を用いてはいけない。

* CHLVBD の場合には実数型をすべて倍精度実数型とする。

	1	2	4	7					
		3	5	8	11				
			6	9	12	15	19	...	
				10	13	16	20	...	
					14	17	21	...	
						18	22	...	
							23	...	
<hr/>									
NB	1	2	3	4	4	4	5	...	

(3) 備考

本ルーチンが持つコレスキー分解成分再利用の機能を活用すれば、逆行列を計算する必要はほとんどなくなり、非常に合理的である。なぜなら、逆行列を計算すると帯行列ではなくなるために、帯行列であるための有利さが失われてしまう。また、計算速度においても、精度においても、コレスキー分解の再利用の方が逆行列を経由するよりも優れているのである。

PRCGFS / D

Solution of Linear Equations with Positive Symmetric Coefficient Matrix by Conjugate Gradient Method with Preconditioning

前処理付き共役勾配法による対称正値連立一次方程式の解法

作成	都田 艶子, 鳥居 達生 1982年2月		
形式	サブルーチン	言語 ;	FORTTRAN サイズ ; 85, 86行

(1) 概要

係数行列 A が正定値対称のフルマトリックスである連立一次方程式 $Ax = b$ を, 前処理して収束条件をよくして共役勾配法により解く。

近似解ベクトルがわかっている, それを補正したいとき用いるとよい。1度 PRCGFS/D を呼び出したあと, 前処理部分をスキップして再計算するための入口名として RECGFS/D を備えている。

(2) 使用法

CALL PRCGFS/D (A, NA, N, B, X, OMEGA, EPS, NMAX, W, IDUMP)

引 数	型 と 種 類	属 性	内 容
A	実 数 型 2 次 元 配 列	入 力	係数行列。保存されない。
NA	整 数 型	入 力	Aの整合寸法。NA ≥ N
N	整 数 型	入 力	方程式の元数。
B	実数型1次元配列	入 出 力	近似解ベクトル。大きさN。
X	実 数 型 1 次 元 配 列	入 出 力	入力: 右辺例。 出力: 解ベクトル。
OMEGA	実 数 型	入 力	反復法の収束の加速因子。1 ≤ OMEGA < 2。ただし, OMEGA として, 範囲外の値が入力されたときは OMEGA = 1 とみなして, 計算を実行する。
EPS	実 数 型	入 力	収束判定定数。残差の2乗和が EPS**2 より小のとき収束とする。ただし, EPS が小さすぎるときは, EPS として, 8 · u · b とみなす。u は丸め誤差の単位である。
NMAX	整 数 型	入 力	最大反復回数。理論的には, たかだかNでよいが, 与えたNMAX が大きすぎるとき, 3 · N/2 で置き換える。
W	倍精度実数型 1 次 元 配 列	作 業 領 域	大きさ N * 5。
IDUMP	整 数 型	入 出 力	入力引数としては次の意味を持つ。 IDUMP ≤ 0 : 途中印字無し。 IDUMP = 1 : 途中, 反復ごとの残差のノルムと (p, Ap) の印字。 IDUMP ≥ 2 : 反復ごとの近似解, 残差, A-直交系ベクトルの印字。 出力引数としては次の意味を持つ。

引 数	型 と 種 類	属 性	内 容
			入力と同じ： 正常終了。 IDUMP = NMAX : NMAX回の反復で収束せず。 IDUMP = 30000 : 入力引数のエラー。

(3) 計算法

$A = M^{-1}N$ と正則分割する。ただし、 M は A と同じ性質、正定値対称で、簡単に逆行列が求まるものであるようにする。この M を用いて前処理した系、

$$M^{-1}Ax = M^{-1}b$$

を考える。そして、

$$\bar{A} = M^{-\frac{1}{2}}AM^{-\frac{1}{2}}, \bar{b} = M^{-\frac{1}{2}}b, y = M^{\frac{1}{2}}x$$

とし $\bar{A}x = \bar{b}$

と書き直すと、これはもとの系に同値な正定値の系となっている。この系に対して共役勾配法のアルゴリズムを適用する。行列 M の構成は、ここでは(野寺, 高橋)¹⁾による以下のやり方によった。

$$A = L_0 + D + L_0^T$$

と分解、ただし、

L_0 : 下半三角行列 (対角要素 0)

D : 対角行列

$D^{-\frac{1}{2}}$ を A の両側から作用させる

$$A' = D^{-\frac{1}{2}}AD^{-\frac{1}{2}} = L'_0 + I + L'^0_T$$

(4) 使用例

```

C      MAIN FOR PRGFS
      DOUBLE PRECISION SU,W
      DIMENSION B(100),A(100,100),X0(100),X(100),W(500)
      NR=5
      NW=6
      EPS=1.E-5
      NA=100
      N=100
      XX=1.0E8+1.
      DO 10 I=1,N
      DO 10 J=1,N
      IJ=IABS(I-J)
10     A(I,J)=FLOAT(N-IJ)
      XI=12345678.0
      DO 20 I=1,N
      X0(I)=0.0
      X(I)=4.*XI/1.E8-2.
      XI=AMOD(23.*XI,XX)
20     CONTINUE
      DO 40 IK=1,N
      SU=0.
      DO 30 I=1,N
30     SU=SU+A(IK,I)*X(I)
40     B(IK)=SU
      WRITE(NW,1000)N
1000  FORMAT(1H1,15H EXAMPLE 3-6 N=,I4)
      IDUMP=0
      OMEGA=1.00
      CALL CLOCKM(JTIME1)

```

```

NMAX=100
CALL PRGFS(A,NA,N,B,XO,OMEGA,EPS,NMAX,W,IDUMP)
CALL CLOCKM(JTIME2)
JT=JTIME2-JTIME1
WRITE(NW,1020)IDUMP,JT
DO 50 I=1,N
RES=X(I)-XO(I)
WRITE(NW,1010) I,X(I),XO(I),RES
50 CONTINUE
1010 FORMAT(15,2E15.6,E11.3)
1020 FORMAT(7H IDUMP=,15,3X,'TIME =',15)
1030 FORMAT(2H X//(D23.15))
STOP
END

```

```

EXAMPLE 3-6 N= 100
IDUMP=      0   TIME = 130
 1 -0.150617E+01 -0.150611E+01 -0.572E-04
 2  0.135802E+01  0.135791E+01  0.116E-03
 3 -0.765456E+00 -0.765381E+00 -0.755E-04
 4 -0.160547E+01 -0.160547E+01 -0.572E-05
 5 -0.925886E+00 -0.925903E+00  0.170E-04

```

参 考 文 献

- 1) T. Nodera and H. Takahasi; "Preconditioned Conjugate Gradient Algorithm for Solving Biharmonic Equation" 4th IMACS, International Symposium (1981).

CGHTCS / D

Solution of Linear Equations with Positive Symmetric Coefficients Matrix by Conjugate Gradient Method (Compact Mode)

共役勾配法による対称正値連立一次方程式の解法（圧縮モード）

作成	都田 艶子, 鳥居 達生	1982年2月
形式	サブルーチン	言語 ; FORTRAN サイズ ; 55, 56行

(1) 概要

対称正値行列 A を係数行列とし、その右上半分を一列化して入力した場合の共役勾配法での解法ルーチンである。

(2) 使用法

CALL CGHTCS/D (A, NA, N, B, X, EPS, NMAX, W, IDUMP)

引数	型と種類	属性	内容
A	実数型 1次元配列	入力	係数行列の右上半分を1次元配列になおして入力。要素(I, J)(I ≤ J)を1次元配列のJ*(J-1)/2+I番目の要素とする。
NA	整数型	入力	係数行列を1次元配列にしたときのベクトルの長さ。
N	整数型	入力	方程式の次元数。
B	実数型 1次元配列	入力	方程式の右辺列。
X	実数型 1次元配列	入出力	入力: 近似解ベクトル(初期値)。 出力: 補正された解ベクトル。
EPS	実数型	入力	収束判定定数。小さすぎるとき、EPSとして $8 \cdot u \cdot \ b\ $ とみなす。uは丸め誤差の単位である。
NMAX	整数型	入力	最大反復回数。大きすぎる値を入力したときは $3 \cdot N/2$ とみなす。
W	倍精度実数型 1次元配列	作業領域	大きさ $N \times 3$ 。
IDUMP	整数型	入出力	入力引数としては次の意味を持つ。 IDUMP < 0 : 途中結果印字無し。 IDUMP = 1 : 反復ごとの残差(P, AP)の印字。 IDUMP > 2 : 反復ごとの近似解, 残差, A-直交系ベクトルの印字。 出力引数としては次の意味を持つ。 入力と同じ: 正常終了。 IDUMP = 3*N: 3*N回反復しても収束せず。 IDUMP = 30000: 入力パラメータ・エラー。

(3) 計算法

連立方程式 $Ax = b$ の解を、残差 $r = b - Ax$ としたとき、誤差関数 $\varphi(x) = (r, A^{-1}r)$ を最小とするよ

うにして求める。共役勾配法 Original 版における補正ベクトル (A -直交系) の計算式を $|r_{i+1}|^2$ で両辺割り、一種の正規化を行った形の公式を用いる(高橋版)。このとき、計算式は簡潔となるが、残差の減少速度、計算時間等はあまりかわりない。算法の式は以下で与えられる。

初期値 $x_0 = 0$, $r_0 = b - Ax_0$, $P_0 = r_0 / |r_0|^2$

$$x_i = 1 / (P_i, AP_i)$$

$$x_{i+1} = x_i + \alpha_i P_i$$

$$r_{i+1} = r_i - \alpha_i AP_i$$

収束判定 $|r_{i+1}|^2 < (\text{EPS})^2$ か?

$$P_{i+1} = P_i - r_{i+1} / |r_{i+1}|^2$$

$i = 0, 1, \dots$

(4) 使用例

```

C      MAIN FOR CGHTCS
      DIMENSION AS(5050),X(100),B(100),A(100),X0(100)
      DOUBLE PRECISION W(300),SU
      NR=5
      NW=6
      EPS=0.1E-4
      XX=0.1E8+1.
      N=100
      XI=12345678.00
      DO 10 I=1,N
      X0(I)=0.0
      X(I)=4.*XI/1.E8-2.
      XI=AMOD(23.*XI,XX)
10     CONTINUE
      NT=0
      DO 40 I=1,N
      SU=0.0D0
      DO 20 J=1,N
      IJ=IABS(I-J)
      A(J)=FLOAT(N-IJ)
20     SU=A(J)*X(J)+SU
      B(I)=SU
      DO 30 J=1,I
      NT=NT+1
30     AS(NT)=A(J)
40     CONTINUE
      WRITE(NW,1000)N
1000  FORMAT(1H1,15H EXAMPLE 3-6 N=,I4)
      IDUMP=0
      MAXN=100
      CALL CLOCKM(JTIME1)
      CALL CGHTCS(AS,NT,N,B,X0,EPS,MAXN,W,IDUMP)
      CALL CLOCKM(JTIME2)
      JTIME=JTIME2-JTIME1
      WRITE(6,1010) IDUMP,JTIME
1010  FORMAT(1H , ' IDUMP =',I5,3X,' TIME(MSEC)=' ,I5)
      DO 50 I=1,N
      RES=X(I)-X0(I)
      WRITE(NW,1020) I,X(I),X0(I),RES
1020  FORMAT(I5,2E13.5,E11.3)
      50 CONTINUE
1030  FORMAT((1H ,5(1PE13.5)))
      STOP
      END

      EXAMPLE 3-6 N= 100
      IK= 26 ZANSA= 0.53581611D-02
      IDUMP = 0 TIME(MSEC)= 565
      1 -0.15062E+01 -0.15031E+01 -0.303E-02
      2 -0.18420E+01 -0.18510E+01 0.899E-02

```

3 -0.19655E+01 -0.19646E+01 -0.883E-03
4 -0.16055E+01 -0.16043E+01 -0.119E-02

(5) 備考

共役勾配法は、係数行列が大次元でスパースである時、収束が速いという特徴をもつ。フルマトリクスの係数行列の場合、他の方法か前処理つき共役勾配法 (PRCGFS/D) を使用の方が望ましい。

参考文献

- 1) 戸川隼人; 共役勾配法, 教育出版 (1977)。

TRIDGS / D

Solution of Tridiagonal Equations

三項方程式の解法

作成	二宮 市三 1977年4月
形式	サブルーチン 言語 ; FORTRAN サイズ ; 39, 40行

(1) 概要

三項方程式、すなわち、三重対角行列を係数行列とする連立一次方程式を、ピボット選択のための行交換を伴うガウスの消去法で解く。

$$\begin{bmatrix} B_1 & C_1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ A_1 & B_2 & C_2 & 0 & \cdots & 0 & 0 & 0 \\ 0 & A_2 & B_3 & C_3 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & B_{n-2} & C_{n-2} & 0 \\ 0 & 0 & 0 & 0 & \cdots & A_{n-2} & B_{n-1} & C_{n-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & A_{n-1} & B_n \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ \vdots \\ X_{n-2} \\ X_{n-1} \\ X_n \end{bmatrix} = \begin{bmatrix} D_1 \\ D_2 \\ D_3 \\ \vdots \\ D_{n-2} \\ D_{n-1} \\ D_n \end{bmatrix}$$

(2) 使用法

CALL TRIDGS/D (A, B, C, D, N, EPS, ILL)

引数	型と種類*	属性	内容
A	実数型 1次元配列	入力	係数行列のN-1個の下対角線要素を左上から右下への順に入れる。保存されない。
B	実数型 1次元配列	入力	係数行列のN個の対角線要素を左上から右下への順に入れる。保存されない。
C	実数型 1次元配列	入力	係数行列のN-1個の上対角線要素を左上から右下への順に入れる。保存されない。
D	実数型 1次元配列	入出力	方程式の右辺列を入れると、本ルーチンにより計算されたベクトルが入る。
N	整数型	入力	方程式の元数。保存される。N ≥ 3
EPS	実数型	入力	方程式の特異性の判定常数。ピボット要素の絶対値がこの常数より小さいとき特異と判定して、計算を中断する。保存される。EPS > 0
ILL	整数型	出力	ILL=0: 正常終了。 ILL 30000: N, EPS に関する制限が破られたとき。特異と判定されたときはピボット要素の番号。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 性能

精度は問題に依存するので、一概に言えない。計算時間は元数に比例するに過ぎない。同じ問題を一

般の連立一次方程式サブルーチン (LEQLUS, CHOLFS, GAUELS 等) で解くと、計算時間は元数の 3 乗に比例するので、非常に時間がかかる。

(4) 備考

1. 三項方程式は一般の連立方程式ルーチンでも解けるが、本ルーチンを用いるのが計算時間の上から合理的である。
2. EPS の標準的な値としては、係数行列の要素の絶対値の代表的な値を a とするとき、TRIDGS (TRIDGD) に対しては $a \times 10^{-6}$ ($a \times 10^{-16}$) が適当である。
3. 対称正値三項方程式の場合は、そのための専門ルーチン TRDSPS, TRDSPD 又は TDSPCS, TDSPCD を用いる方が有利である。

TRDSPS / D

TDSPCS / D

Solution of Symmetric Positive Definite Tridiagonal Equations

対称正値三項方程式の解法

作 成	二宮 市三 1977年4月
形 式	サブルーチン 言語 ; FORTRAN サイズ ; 29, 29, 45, 46行

(1) 概 要

対称正値三項方程式、すなわち、対称正値三重対角行列を係数行列とする連立一次方程式を、平方根を用いないコレスキー分解法によって解く。

TRDSPS/D は $C(N) = 0$ の場合を扱い、TDSPCS/D は $C(N) \neq 0$ 、すなわち巡回型の三項方程式を扱う。両者とも、複数右辺列の同時処理、行列式の計算及びコレスキー分解成分の再利用の機能を備えている。

(2) 使用法

CALL TRDSPS/D (B, C, N, X, KX, M, DET, EPS, IND)

CALL TDSPCS/D (B, C, D, N, X, KX, M, DET, EPS, IND)

引 数	型 と 種 類*	属 性	内 容
B	実 数 型 1 次 元 配 列	入 出 力	係数行列の対角線要素を入力すると、コレスキー分解成分の対角線要素が出力される。
C	実 数 型 1 次 元 配 列	入 出 力	係数行列の副対角線要素を入力すると、コレスキー分解成分の副対角線要素が出力される。副対角線要素の番号のつけ方は図示のとおり。
D	実 数 型 1 次 元 配 列	出 力	係数行列のコレスキー分解成分が出力される。
N	整 数 型	入 力	方程式の元数。配列 B, C, D の要素の数でもある。 $N \geq 3$
X	実 数 型 2 次 元 配 列	入 出 力	M本の右辺列を一つの行列Xに合併して入力すると、それぞれの列に対応する解ベクトルが出力される。
KX	整 数 型	入 力	Xの配列宣言における第1添字の値。 $KX \geq N$
M	整 数 型	入 力	Xの列数。 $M=0$ のとき係数行列のコレスキー分解のみを行う。
DET	実 数 型	入 出 力	$DET \neq 0$ を入力すると係数行列式の値が出力される。 $DET = 0$ を入力すると、そのまま出力される。
EPS	実 数 型	入 力	係数行列の非正値性の判定常数。ピボット要素がこの常数よりも小となったとき正値でないと判定して計算を中断する。 $EPS > 0$
IND	整 数 型	入 出 力	入力引数としては次の意味を持つ。 IND = 0: 新たにコレスキー分解を行い、方程式を解く。 IND \neq 0: 以前に計算してB, C, Dに保存されているコレスキー成分を再利用して方程式を解く。

引数	型と種類*	属性	内容
			出力引数としては次の意味を持つ。 IND=0: 正常に計算が終了した。 IND=30000: 入力引数に関する制限が破られた。 IND=1: コレスキー分解の I 段目で正值でないと判定されたので、計算を中断した。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 性能

一般に、計算時間は元数の 1 乗に比例するのみである。同じ問題を普通の連立一次方程式ルーチン (LEQLUS, CHOLFS, GAUELS 等) で解くと、計算時間は元数の 3 乗に比例するので、非常に時間がかかる。

$$\begin{bmatrix}
B_1 & C_1 & 0 & \vdots & \vdots & \vdots & 0 & C_n \\
C_1 & B_2 & C_2 & \vdots & \vdots & \vdots & 0 & 0 \\
0 & C_2 & B_3 & \vdots & \vdots & \vdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & & & \vdots & \vdots \\
\vdots & \vdots & \vdots & & & & \vdots & \vdots \\
\vdots & \vdots & \vdots & & & & \vdots & \vdots \\
0 & 0 & 0 & \vdots & \vdots & \vdots & B_{n-1} & C_{n-1} \\
C_n & 0 & 0 & \vdots & \vdots & \vdots & C_{n-1} & B_n
\end{bmatrix}$$

(4) 備考

1. 係数が正值対称でない三項方程式には、TRIDGS/D が用意されている。
2. 同じ方程式を右辺列を変更しながら繰り返し解く場合には、本ルーチンのコレスキー分解成分の再利用の機能を用いると合理的である。

LEQLSS / D

Least Square and Minimum Norm Solutions of General Simultaneous Linear Equations by Householder Transformation

ハウスホルダー変換による一般連立一次方程式の最小二乗解及び最小ノルム解

作 成	二宮 市三 1979年3月
形 式	サブルーチン 言語; FORTRAN サイズ; 94, 94行

(1) 概 要

m 行 n 列で ($m \geq n \geq 1$) 階数が n の行列 A と m 行 l 列の行列 B が与えられたとき、優決定 (over-determined) の連立一次方程式

$$AX = B$$

の最小二乗解 (n 行 l 列)

$$X = (A^T A)^{-1} A^T B$$

を、 A をハウスホルダー変換によって三角化することにより計算する。また、同様な行列 A と n 行 k 列の行列 B が与えられたとき、劣決定 (underdetermined) の連立一次方程式

$$A^T X = B$$

の最小ノルム解 (m 行 k 列)

$$X = A(A^T A)^{-1} B$$

を、同様の方法により計算する。

(2) 使用法

CALL LEQLSS/D (A, KA, N, X, KX, NX, EPS, R, Q, ISW, ILL)

引 数	型 と 種 類*	属 性	内 容
A	実 数 型 2 次 元 配 列	入 力	係数行列。ハウスホルダー変換により三角化される。
KA	整 数 型	入 力	Aの配列宣言における第1添字の値。KA ≥ M
M	整 数 型	入 力	Aの行数。M ≥ N
N	整 数 型	入 力	Aの列数。N ≥ 1
X	実 数 型 2 次 元 配 列	入 出 力	右辺行列Bを入れて本ルーチンを引用すると解行列Xが出力される。M行NX列の2次元配列。
KX	整 数 型	入 力	Xの配列宣言における第1添字の値。KX ≥ M
NX	整 数 型	入 力	Xの列数。NX ≤ 0とするとAの三角化だけが行われる。
EPS	実 数 型	入 力	Aの階数落ちの判定定数ε。三角化の過程で対角要素の絶対値が A ₁ ・εより小さいとき階数落ちと判定し、処理を中断する。EPS ≤ 0とすると標準値として丸めの誤差の最小単位が設定される。
R	実 数 型 1 次 元 配 列	出 力	大きさNXの1次元配列。各解ベクトル(Xの各列)の残差ノルム又はノルムが出力される。

引 数	型 と 種 類*	属 性	内 容
Q	実 数 型 1 次 元 配 列	作 業 領 域	大きさNの1次元配列。
ISW	整 数 型	入 力	ISW ≥ 0のときは最小二乗解を、ISW < 0のときは最小ノルム解を求める。また、 ISW ≤ 1のときはAの三角化を行い、 ISW > 2のときはAの三角化を省略する。
ILL	整 数 型	入 力	ILL = 0 : 正常終了。 ILL = 20000 : 階数落ち。 ILL = 30000 : 入力変数エラー。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 性 能

最小二乗解（最小ノルム解）を求める従来からの方法では、係数行列 $A(A^T)$ から $A^T A$ を作り、これを係数とする連立一次方程式を解くので、方程式の条件が悪化することが多く精度の良い解を求めることが困難である。

一方、本サブルーチンでは、 $A^T A$ を作ることなくハウスホルダーの直交変換 H により、 A 自体を上三角行列 $U = HA$ に変形するので条件が悪化することはない。したがって、計算量の点では従来の方法にやや劣るが、解の精度はすぐれている。

M 次のヒルベルト行列 ($a_{ij} = 1/(i+j-1)$) の始めの N 列を A とし、厳密解のすべての成分が1となるように B を与えた場合の数値解の精度についての実験結果は下表のとおりである。

	最 小 二 乗 解 N = 10		最 小 ノ ル ム 解 N = 10	
	M = 10	M = 20	M = 10	M = 20
LEQLSD	5 桁	8 桁	6 桁	4 桁
従 来 の 方 法 CHOLFD使用	IND = 8 中 断	IND = 9 中 断	IND = 8 中 断	IND = 9 中 断

(4) 使 用 例

上述の実験において、最小二乗解を求めるために用いたプログラムを次に示す。

```

1      IMPLICIT REAL*8 (A-H,O-Z)
2      DIMENSION A(20,10),X(20),Q(10)
3      M=20
4      N=10
5      EPS=1.0D-17
6      ISW=0
7      DO 10 I=1,M
8          X(I)=0.0D0
9          DO 10 J=1,N
10             A(I,J)=1.0D0/DFLOAT(I+J-1)
11             10 X(I)=A(I,J)+X(I)
12             CALL LEQLSD(A,M,M,N,X,M,1,EPS,R,Q,ISW,ICON)
13             WRITE(6,600) ICON,R,(X(I),I=1,N)
14             600 FORMAT(1H1,110,D25.17/(1H ,10X,D25.17))
15             STOP
16             END

```

(5) 計算法

m 行 n 列 ($m \geq n \geq 1$) の行列 M , m 次のベクトル v を始めの n 行の部分と残りの $m-n$ 行の部分に分けて

$$M = \begin{pmatrix} M_1 \\ M_2 \end{pmatrix} \quad v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

というふうを書くことにする。簡単のために右辺がベクトル b の場合について説明する。

1. 最小二乗解

残差 $r = Ax - b$ のノルム $\|r\|_2$ を最小にする。 A の左からハウスホルダーの直交変換 H を乗じて、

$$HA = \begin{pmatrix} U_1 \\ O_2 \end{pmatrix}$$

の形にする。ただし U_1 は右上三角行列、 O_2 は零行列である。

$$Hr = HAx - Hb$$

であり、 $\|Hr\|_2 = \|r\|_2$ であるから、 $\|Hr\|_2$ を最小にすればよい。

$$Hb = \begin{pmatrix} \tilde{b}_1 \\ \tilde{b}_2 \end{pmatrix}$$

とすると、

$$Hr = \begin{pmatrix} U_1 x - \tilde{b}_1 \\ -\tilde{b}_2 \end{pmatrix}$$

となる。したがって、最小二乗解は、上三角行列についての後退代入法により $x = U_1^{-1} \tilde{b}_1$ として求められ、

$$\|r\|_2 = \|Hr\|_2 = \|\tilde{b}_2\|$$

である。

2. 最小ノルム解

$$A^T x = b$$

の無限個の解の中で $\|x\|_2$ 最小のものを求める。最小二乗解のときと同じ変換により、

$$(HA)^T Hx = b$$

となる。 $y = Hx$ とおくと、

$$(U_1^T O_2^T) \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = b$$

すなわち、 $U_1^T y_1 = b$

となる。この y の最小ノルム解は下三角行列 U_1^T についての前進代入法により

$$y = \begin{pmatrix} U_1^{-T} b \\ 0_2 \end{pmatrix}$$

で与えられる。ただし、 0_2 は $m-n$ 次の零ベクトルとする。この y から、

$$x = H^T y$$

によって x を求めれば、

$$\|x\|_2 = \|H^T y\|_2 = \|y\|_2$$

であるから、この x が最小ノルム解となる。

(6) 備考

(1) ISW(2)

1. A が共通で B が異なる問題を多数回解く場合には、最初の引用では $|ISW| \leq 1$ とし、そのときの A と Q とを保存しておいて、第2回目からは $|ISW| \geq 2$ として引用すれば、ハウスホルダー変換が省略できて都合がよい。
2. 本サブルーチンでは、 A の階数が n より小さい場合は取り扱うことができない。そのような場合には、特異値分解によるサブルーチン LSMNS/Dを用いるのがよい。

LSMNS / D

Least Square and Minimal Norm Solutions of General Simultaneous Linear Equations
by Singular Value Decomposition

特異値分解による一般連立一次方程式の最小二乗最小ノルム解

作 成	二宮 市三 1979年3月
形 式	サブルーチン 言語; FORTRAN サイズ194, 194行

(1) 概 要

m 行 n 列行列 A と m 行 l 列行列 B が与えられたとき,

$$\|Ax_i - b_i\|_2 \quad i = 1, 2, \dots, l$$

と

$$\|x_i\|_2 \quad i = 1, 2, \dots, l$$

を最小にするような n 行 l 列行列 X を求める。ただし,

$$B = [b_1, b_2, \dots, b_l]$$

$$X = [x_1, x_2, \dots, x_l]$$

である。このような最小二乗最小ノルム解を求めるには、まず A を特異値分解する。

$$A = U\Sigma V^T$$

ただし、 U は m 行 n 列、 Σ と V は n 行 n 列で

$$U^T U = V^T V = V V^T = I_n \text{ (} n \text{ 次単位行列)}$$

$$\Sigma = \text{diag}(q_1, q_2, \dots, q_n)$$

であり、 $q_1 \geq q_2 \geq \dots \geq q_n \geq 0$ は A の特異値、すなわち $A^T A$ の固有値の正平方根である。

このとき解 X は、

$$X = V\Sigma^*U^T B$$

で与えられる。ここに

$$\Sigma^* = \text{diag}(q_1^+, q_2^+, \dots, q_n^+)$$

$$q_i^+ = \begin{matrix} 1/q_i & q_i > 0 \\ 0 & q_i = 0 \end{matrix} \quad i = 1, 2, \dots, n$$

である。

(2) 使用法

CALL LSMNS/D (A, KA, M, N, B, KB, NB, Q, EPS, W, ILL)

引 数	型 と 種 類*	属 性	内 容
A	実 数 型 2 次 元 配 列	入 出 力	係数行列を入力すると、その始めのN行の部分に直交行列Vが出力される。大きさmax(M, N)行N列の配列。
KA	整 数 型	入 力	Aの配列宣言における第1添字の値。KA ≥ max(M, N)
M	整 数 型	入 力	Aの行数。M ≥ 1
N	整 数 型	入 力	Aの列数。N ≥ 1

引 数	型 と 種 類*	属 性	内 容
B	実 数 型 2 次 元 配 列	入 出 力	右辺行列を入力するとその始めのN行の部分に解行列Xが出力される。大きさ max (M, N) 行 NB 列の配列。
KB	整 数 型	入 力	B の配列宣言における第 1 添字の値。KB ≥ max (M, N)
NB	整 数 型	入 力	B の列数。NB ≥ 1
Q	実 数 型 1 次 元 配 列	出 力	A の特異値が減少順に出力される。大きさ N の 1 次元配列。
EPS	実 数 型	入 力	収束判定と特異値の零判定に用いられる定数 ε。A から両側ハウスホルダー変換でえられる 2 重対角行列を J とするとき $\ J\ _{\infty} \epsilon + u$ を J の非対角要素および特異値の零判定のしきい値とする。EPS ≤ 0.0 とすると ε として丸めの誤差の単位 u が用いられる。
W	実 数 型 1 次 元 配 列	作 業 領 域	大きさ N の 1 次元配列。
ILL	整 数 型	出 力	ILL=0: 正常終了。 ILL=20000: 30N 回以上反復しても特異値分解が収束しない。 ILL=30000: 入力引数が制限を破った。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 性 能

参考文献¹⁾の p. 418 所載の、8 行 5 列で $\sqrt{1248}$, 20, $\sqrt{384}$, 0, 0 という特異値をもつ階数 3 の行列 A を係数とし、8 行 3 列の行列 B を右辺とする問題を LSMNS で解いた。ESP = 10^{-6} ととり変換行列 V を A の上に重ね書きした。特異値 Q, 変換行列 V 及び三つの最小二乗最小ノルム解の精度は約 6 桁であった。

(4) 使用例

上述のテストを行うためのプログラムを次に示す。

```

1      DIMENSION A(8,5),B(8,3),Q(5),W(5),R(5)
2      M=8
3      N=5
4      KA=8
5      KB=8
6      NB=3
7      EPS=1,E-6
8      R(1)=SQRT(1248.)
9      R(2)=20.
10     R(3)=SQRT(384.)
11     R(4)=0.
12     R(5)=0.
13     READ(5,500) ((A(I,J),J=1,N),I=1,M)
14     500 FORMAT(5F4,0)
15     READ(5,510) ((B(I,J),J=1,NB),I=1,M)
16     510 FORMAT(3F4,0)
17     WRITE(6,600) M,N,NB,((A(I,J),J=1,N),I=1,M)
18     *,((B(I,J),J=1,NB),I=1,M)
19     600 FORMAT(1H1//10X,'M =',I2,2X,'N =',I2,2X,'NB =',I2//
20     *8(10X,1P5E13.5)/(10X,3E13.5))
21     CALL LSMNS(A,KA,M,N,B,KB,NB,Q,EPS,W,ICON)
22     WRITE(6,610) EPS,ICON,(Q(J),R(J),J=1,N),
23     *((A(I,J),J=1,N),I=1,N),((B(I,J),J=1,NB),I=1,N)
24     610 FORMAT(/10X,'EPS =',1PE10.2,2X,'ICON =',I6//

```

```
22      *5(10X,2E13.5)/ 5(10X,5E13.5)/(10X,3E13.5))
23      STOP
23      END
```

(5) 備考

1. 特異値分解の収束判定と特異値の零判定に用いられる定数 EPS の選択は慎重に行わなければならない。データとしての A や B がもっている精度に比べて小さすぎる EPS を与えることは、不必要に精密な計算を行う無駄を招き、本来 0 として棄てるべき特異値を意味のあるものとして誤認する危険を犯すことになる。また逆に大きすぎる EPS は、小さいけれども意味のある特異値を 0 として棄ててしまうという誤りの原因となる。
2. 同じ係数行列 A に対して最小二乗最小ノルム解を一度しか計算しない場合に、GINVS/D を用いて一般逆行列 A^+ を求めることは、 A^+ 自体が必要である場合を除いては、計算量の多い点で賢明ではない。是非とも本ルーチン LSMNS/D を使用すべきである。

参考文献

- 1) G.H. Golub, C. Reinsch; "Singular Value Decomposition and Least Squares Solutions", Numerische Mathematik, 14, pp.403-420 (1970).

MINVS/D/Q/C/B

Inversion of Matrices

行列の逆転

作成	二宮 市三 1977年4月
形式	サブルーチン 言語; FORTRAN サイズ; 96, 97, 96, 95, 96行

(1) 概要

LU-分解法によって、与えられた行列の逆行列を与えられた行列の位置に作る。この際、ピボット選択のために必要ならば、行の交換を行う。

(2) 使用法

CALL MINVS/D/Q/C/B(A, KA, N, EPS, ILL)

引数	型と種類	属性	内容
A*	実数型 2次元配列	入出力	与えられた行列を入れると、本ルーチンにより計算された逆行列が入る。
KA	整数型	入力	Aの配列宣言における第1添字の値。KA ≥ N
N	整数型	入力	Aの次数。2 ≤ N ≤ 1000
EPS**	実数型	入力	行列の特異性の判定常数。ピボット要素の絶対値がこの常数より小さいとき特異であると判定して、計算を中断する。EPS > 0
ILL	整数型	出力	ILL=0: 正常終了。 ILL=30000: KA, N, EPSに関する制限が破られたとき。特異と判定されたときはピボット要素の番号。

* MINVD (MINVQ, MINVC, MINVB) の場合、Aは倍精度実数型（4倍精度実数型、複素数型、倍精度複素数型）である。

** MINVD (MINVQ, MINVC, MINVB) の場合、EPSは倍精度実数型（4倍精度実数型、実数型、倍精度実数型）である。

(3) 計算法

1. ピボットニングに伴う行交換に対応する置換行列 P を A に施したものを、単位下三角行列 L と上三角行列に分解する。 $PA = LU$
2. L の位置に L^{-1} を作る。
3. U の位置に U^{-1} を作る。
4. $A^{-1} = U^{-1}L^{-1}P$ を A の位置に作る。

MINVS の場合、以上のすべての段階に必要な積和計算を部分的倍精度演算で行う。

(4) 備考

1. 行列の要素の絶対値の間に大きな差があるときは、MNORMS, MNORMD 等によって、あらかじめ正規化することが精度を保つ上で望ましい。この際に必要な後処理については、MNORMS に関する説明の項を参照されたい。

2. EPS の標準的な値としては、行列の要素の絶対値の代表的な値を a とすると、MINVS, (MINVD, MINVQ) に対しては $a \times 10^{-6}$ ($a \times 10^{-16}$, $a \times 10^{-30}$) が適当である。
3. A の逆行列を、 $A^{-1}B$ の形の行列積の計算のために計算することは、計算時間と精度との両方の立場から見て大変損である。ぜひとも連立一次方程式ルーチン LEQLUS, LEQLUD 等を使うべきである。
4. 対称正値行列の逆行列を求めるには、そのための専門ルーチン MINVSP, MINVDP 等を用いる方が賢明である。

MINVSP / MINVDP / MINVQP

Inversion of Symmetric Positive Definite Matrices

対称正値行列の逆転

作成	二宮 市三 1977年4月
形式	サブルーチン 言語; FORTRAN サイズ; 41, 41, 41行

(1) 概要

コレスキー分解法により、対称正値行列 A の逆行列を与えられた行列の位置に作る。

(2) 使用法

CALL MINVSP (A, KA, N, EPS, ILL)

CALL MINVDP (A, KA, N, EPS, ILL)

CALL MINVQP (A, KA, N, EPS, ILL)

引数	型と種類*	属性	内容
A	実数型 2次元配列	入出力	対称正値行列を入れると本ルーチンにより逆行列が出力される。対称性により、対角線を含んで右上半分だけを処理する。左下半分は保存される。
K	整数型	入力	Aの配列宣言における第1添字の値。KA ≥ N
N	整数型	入力	Aの次数。N ≥ 2
EPS	実数型	入力	行列Aの正値性の判定常数。ピボット要素がこの常数より小さいとき正値でないと判定して、計算を中断する。EPS > 0
ILL	整数型	出力	ILL=0: 正常終了。 ILL=30000: KA, N, EPSに関する制限が破られたとき正値でないと判定されたときはピボット要素の番号。

* MINVDP (MINVQP) の場合は、実数型をすべて倍(4倍)精度実数型とする。

(3) 計算法

1. A のコレスキー分解成分 U ，すなわち， $A = U^T U$ となるような上三角行列 U を A の右上三角部分に作る。
2. U の逆行列 U^{-1} ，すなわち， $UV = I$ となるような上三角行列 V を U の位置に作る。
3. A の逆行列 $A^{-1} = VV^T$ の右上半分を V の位置に作る。

MINVSP の場合、以上のすべての段階で必要なすべての積和計算を、部分的倍精度演算で行う。

(4) 備考

1. EPS の標準的な値としては、行列の要素の絶対値の代表的な値を a とすると、MINVSP (MINVDP, MINSQP に対しては $a \times 10^{-6}$ ($a \times 10^{-16}$, $a \times 10^{-30}$) が適当である。
2. A の逆行列を、 $A^{-1}B$ の形の行列積の計算のために計算するのは、計算時間と精度との両方の立場から見て大変損である。ぜひとも連立一次方程式ルーチン CHOLFS, CHOLFD 等を使うべきである。

GINVS / D

Generalized Inverses (Pseudo-inverses) by Singular Value Decomposition

特異値分解による一般化逆行列

作成	二宮 市三 1979年3月
形式	サブルーチン 言語; FORTRAN サイズ; 30, 30行

(1) 概要

m 行 n 列行列 A に対して, 次の関係を満足する n 行 m 列行列 X を一般化逆行列という。

$$AXA = A$$

$$XAX = X$$

$$(AX)^T = AX$$

$$(XA)^T = XA$$

A に対してこのような X は一義的に定まる。この X を A^+ と書く。いま, A の特異値分解

$$A = U\Sigma V^T$$

が与えられているとする。ただし U は m 行 n 列行列, Σ と V は n 行 n 列行列で

$$U^T U = V^T V = V V^T = I_n (n \text{ 次単位行列})$$

$$\Sigma = \text{diag}(q_1, q_2, \dots, q_n)$$

$$q_1 \geq q_2 \geq \dots \geq q_n \geq 0$$

であり, $q_i, i = 1, 2, \dots, n$ は A の特異値 ($A^T A$ の固有値の正平方根) である。

このとき A^+ は

$$A^+ = V \Sigma^+ U^T$$

で与えられる。ここに

$$\Sigma^+ = \text{diag}(q_1^+, q_2^+, \dots, q_n^+)$$

であって

$$q_i^+ = \begin{cases} 1/q_i & q_i > 0 \\ 0 & q_i = 0 \end{cases}$$

ととるものとする。

本サブルーチンの目的は A が与えられたとき, 特異値分解によって A^+ を求めることである。

(2) 使用法

CALL GINVS/D(A, KA, M, N, Q, V, KV, EPS, W, ILL)

引数	型と種類*	属性	内容
A	実数型 2次元配列	入出力	Aを入力とすると, Aの一般化逆行列の転置行列(A^+) ^T が出力される。
KA	整数型	入力	Aの配列宣言における第1添字の値。KA ≥ M
M	整数型	入力	Aの行数。M ≥ 1
N	整数型	入力	Aの列数。N ≥ 1

引数	型と種類*	属性	内容
Q	実数型 1次元配列	出力	Aの特異値が減少順に出力される。大きさNの1次元配列。
V	実数型 2次元配列	出力	特異値分解のための直交変換行列Vが出力される。N行N列の2次元配列。
KV	整数型	入力	Vの配列宣言における第1添字の値。KV ≥ N
EPS	実数型	入力	収束及び零判定のために用いられる定数ε。Aを両側ハウスホルダー変換でいったん2重対角行列Jに変換したときの J _∞ を使って ε J _∞ + u を特異値分解の収束判定及び特異値の零判定のしきい値とする。ただしuは丸めの誤差の単位である。EPS ≤ 0.0 とするとεとしてuが使われる。
W	実数型 1次元配列	作業領域	大きさNの1次元配列。
ILL	整数型	出力	ILL=0: 正常終了。 ILL=30000: 引数が制限を破った。 ILL=20000: 30N回反復してもAの特異値分解が収束しない。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 性能

参考文献¹⁾のp. 418 所載の、8行5列で $\sqrt{1248}$, 20, $\sqrt{384}$, 0, 0という特異値をもつ階数3の行列Aを係数とし、8行3列の行列Bを右辺とする問題に対し、先づGINVSによりA⁺を求め、A⁺Bとして最小二乗最小ノルム解を計算した。EPS = 10⁻⁶とした場合、特異値Q, 変換行列V, 一般逆行列A⁺, 解ベクトルA⁺Bの精度は約6桁であった。

(4) 使用例

上述のテストを行うためのプログラムを次に示す。

```

1      DIMENSION A(8,5),B(8,3),V(5,5),Q(5),W(8),R(5)
2      M=8
3      N=5
4      NB=3
5      KA=8
6      KV=5
7      EPS=1,E-6
8      R(1)=SQRT(1248.)
9      R(2)=20.
10     R(3)=SQRT(384.)
11     R(4)=0.
12     R(5)=0.
13     READ(5,500) ((A(I,J),J=1,N),I=1,M)
14     500 FORMAT(5F4,0)
15     READ(5,510) ((B(I,J),J=1,NB),I=1,M)
16     510 FORMAT(3F4,0)
17     WRITE(6,600) M,N,NB,((A(I,J),J=1,N),I=1,M)
18     * , ((B(I,J),J=1,NB),I=1,M)
19     600 FORMAT(1H1///10X,'M =',I2,2X,'N =',I2,2X,'NB =',I2//
20     *8(10X,1P5E13.5)/(10X,3E13.5))
21     CALL GINVS(A,KA,M,N,Q,V,KV,EPS,W,ICON)
22     DO 30 J=1,NB
23     DO 10 I=1,M
24     10 W(I)=B(I,J)
25     DO 30 I=1,N
26     S=0.
27     DO 20 K=1,M
28     20 S=A(K,I)*W(K)+S

```

```

27      30 B(I,J)=S
28          WRITE(6,610) EPS,ICON,(Q(J),R(J),J=1,N)
          *,((V(I,J),J=1,N),I=1,N),((A(I,J),I=1,M),J=1,N)
          *,((B(I,J),J=1,NB),I=1,N)
29      610 FORMAT(/10X,'EPS =',1PE10,2,2X,'ICON =',16//
          *5(10X,2E13,5)/5(10X,5E13,5)/5(10X,8E13,5)/
          */(10X,3E13,5))
30          STOP
31          END

```

(5) 備考

1. 特異値分解の収束判定と特異値の零判定に用いられる定数 EPS の選択は慎重に行わなければならない。データとしての A がもっている精度に比べて小さすぎる EPS を与えることは、不必要に精密な計算を行う無駄を招き、本来 0 として棄てるべき特異値を意味のあるものとして誤認する危険を犯すことになる。また逆に大きすぎる EPS は、小さいけれども意味のある特異値を 0 として棄ててしまうという誤りの原因となる。
2. 同じ係数行列 A に対して最小二乗最小ノルム解を一度しか計算しない場合に、本ルーチン GINVS/D を用いて一般逆行列 A^+ を求めることは、 A^+ 自体が必要である場合を除いては、計算量の多い点で賢明でない。是非とも LSMNS/D を使用すべきである。

参考文献

- 1) G.H. Golub, C. Reinsch; "Singular Value Decomposition and Least Squares Solutions", Numerische Mathematik, 14, pp.403-420 (1970).

固有値解析ルーチンの選び方

名大プログラム・ライブラリーには対象行列の型、性質、構造及び計算の性質に即して多数のすぐれた固有値解析ルーチンが用意されているので、これらを以下の指針に基づいて慎重に選択すれば、精度、速度及び記憶容量のすべての面で多大の効果を上げることができる。簡単のために推奨ルーチンの名前は単精度用のもので代表させることにする。

- | | |
|-------------------------------|--------|
| 1. 非対称 | HEQRVS |
| 2. 対 称 | |
| (1) 密行列 | |
| (a) 固有値, 固有ベクトルを全部求める | HOQRVS |
| (b) 固有値を全部, 固有ベクトルを全部又は一部分求める | HQRIS |
| (c) 固有値, 固有ベクトルを一部分求める | HOBSVS |
| (d) 固有値, 固有ベクトルを極く一部分求める | JENNFS |
| (2) 帯行列 | |
| (a) 固有値を全部求める | RHQRVS |
| (b) 固有値と固有ベクトルを極く一部分求める | JENNBS |
| 3. 対称行列一般問題 | |
| (1) 密行列 | |
| (a) 固有値, 固有ベクトルを全部求める | GHQRVS |
| (b) 固有値を全部, 固有ベクトルを一部分求める | GHQRIS |
| (c) 固有値, 固有ベクトルを一部分求める | GHBSVS |
| (2) 帯行列 | |
| (a) 固有値, 固有ベクトルを極く一部分求める | GJENBS |
| 4. 特異値分解 | SVDS |

なお以上の外に高速固有値解析パッケージ NICER がある。

HEQRVS / D

Eigenvalue Analysis for Real Nonsymmetric Matrices by Double QR Method

ダブル QR 法による実非対称行列の固有値解析

作 成	二宮 市三	1977 年 4 月	改訂 1981 年 4 月
形 式	サブルーチン 言語 ; FORTRAN サイズ ; 391, 391行		

(1) 概 要

実非対称行列を安定化基本変換によって上ヘッセンベルグ行列に移し、これにダブル QR 法を適用してすべての固有値を求め、これに対する固有ベクトルを要求された個数だけ逆反復法によって計算する。HEQRVS (D) は単 (倍) 精度用のサブルーチンである。

(2) 使用法

CALL HEQRVS/D (A, KA, N, E, F, G, H, NV, EPS, IW, W, IND)

引 数	型 と 種 類*	属 性	内 容
A	実 数 型 2 次 元 配 列	入 力	固有値解析の対象となる行列。本ルーチンで処理されて上ヘッセンベルグ型となる。
KA	整 数 型	入 力	A, G, H の整合寸法 (配列宣言における第 1 添字の値)。 $KA \geq N$
N	整 数 型	入 力	A の次数。G, H の行数。E, F の大きさでもある。 $N \geq 3$
E	実 数 型 1 次 元 配 列	出 力	固有値の実部。I 番目の固有値は $E(I) + iF(I)$ である。
F	実 数 型 1 次 元 配 列	出 力	固有値の虚部。I 番目の固有値は $E(I) + iF(I)$ である。
G	実 数 型 2 次 元 配 列	出 力	I 番目の固有ベクトルの実部が G の第 I 列に出力される。NV + 1 列分を用意しなければならない。
H	実 数 型 2 次 元 配 列	出 力	I 番目の固有ベクトルの虚部が H の第 I 列に出力される。また、作業領域としても使われるので A と同じ大きさの領域を用意しなければならない。
NV	整 数 型	入 力	求めるべき固有ベクトルの個数。共役固有ベクトルは対にして出力されるので、実際に出力されるベクトルは NV + 1 個となることがある。 $0 \leq NV \leq N$
EPS	実 数 型	入 力	$\ A\ \cdot \text{EPS} / N$ が QR の収束判定定数として用いられる。 $\text{EPS} > 0$
IW	整 数 型 1 次 元 配 列	作 業 領 域	大きさ 2N 以上の 1 次元配列。
W	実 数 型 1 次 元 配 列	作 業 領 域	大きさ 3N 以上の 1 次元配列。
IND	整 数 型	入 出 力	入力： 固有値の配列の方法を指定する。 IND = 0 : 計算されたままの順序にする。 IND > 0 : 絶対値の減少順にする。

引 数	型 と 種 類*	属 性	内 容
			IND < 0 : 絶対値の増大順にする。 出力: コンディションコード。 IND = 0 : 正常。 IND = 1 : Aの要素がすべて0である。 IND = 2 : QR法が100 N回反復しても収束しない。 IND = 30000 : 入力引数が制限を犯した。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 計算法

実行列 A を安定化基本変換 S すなわち行交換を伴うガウスの消去法によって、上ヘッセンベルグ行列 $H = S^{-1}AS$ に移す。 H のすべての固有値を、原点シフトを伴うダブル QR 法によって求める。

H の指定された個数の固有ベクトルを逆反復法で求める。これをまとめて U とする。 A の固有ベクトルは、 U から $V = SU$ として計算される。

(4) 備 考

1. 対称行列は、そのための専用ルーチン HOQRVS/D, HQRIS/D, HOBSVS/D など処理するのが合理的である。
2. 固有ベクトルを求めないとき ($NV = 0$) には、 G, H に対する領域は使用しないので用意する必要はなく、ここには何を書いてもよい。

HOQRVS/D

HOQRUS/D

Eigenvalue Analysis for Real Symmetric Matrices by Householder-QR Method

ハウスホルダー・QR法による実対称行列の固有値解析

作成	二宮 市三 1977年4月
形式	サブルーチン 言語; FORTRAN サイズ; 142, 141, 142, 141行

(1) 概要

実対称行列のすべての固有値と、必要ならば、すべての固有ベクトルを、ハウスホルダーの三重対角変換と原点移動を伴うQR法によって求める。

(2) 使用法

CALL HOQRVS/D (A, KA, N, E, F, EPS, ILL)

CALL HOQRUS/D (A, KA, N, E, F, EPS, ILL)

引数	型と種類*	属性	内容
A	実数型 2次元配列	入出力	実対称行列の対角線を含む右上半分を入れる。左下半分は何か入ってもよい。固有ベクトルを求める場合には、Aの各列に固有ベクトルが入る。すなわち、第I列にはE(I)に入っている固有値に対する固有ベクトルが長さ1に正規化されて入る。
KA	整数型	入力	Aの配列宣言における第1添字の値。KA ≥ N
N	整数型	入力	Aの次数。N ≥ 2
E	実数型 1次元配列	出力	N個の要素を持つ1次元配列名。固有値がHOQRVS/Dでは代数的な意味で、また、HOQRUS/Dでは絶対値の意味で、減少順に並べられる。
F	実数型 1次元配列	作業領域	N個の要素を持つ1次元配列名。
EPS	実数型	入力	QR法のための収束判定常数。非対角要素が A ・EPSより小となったら収束と判定する。EPS > 0
ILL	整数型	入出力	入力としては、ILL = 0ならば固有値のみを、ILL ≠ 0ならば固有値と固有ベクトルを計算することを意味する。出力としては、正常に計算が終了したとき0、入力引数に関する制限が破られたときは30000。この引数に対する実引数には、定数を用いてはいけない。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 性能

ヤコビ法に比べて高速であり、多重もしくは近接固有値に対しても支障なく使用できる。

(4) 備 考

1. 少ない所用メモリーで、固有値（と固有ベクトル）をすべて求める場合に最適である。
2. 固有値や固有ベクトルを全部でなくて、小部分だけ求めたい場合には、ハウスホルダー・ギブンス法（二分法）がよい。現在登録されているサブルーチンとしては、HOBSVS/Dがある。
3. 一般形の固有値問題 $Ax = \lambda Bx$ のためには、サブルーチン GHQRVS/D を推奨する。

参 考 文 献

- 1) 戸川隼人；“マトリクスの数値計算”，オーム社（1971）。

HQRIS / D

Eigenvalue Analysis of Symmetric Matrices by Householder-QR-Inverse Iteration Method ハウスホルダー・QR・逆反復法による実対称行列の固有値解析

作成	二宮 市三	1981年4月
形式	サブルーチン 言語 ; FORTRAN サイズ ; 198, 196行	

(1) 概要

実対称行列の固有値全部をハウスホルダー・QR法で求め、指定された固有ベクトルを逆反復法で計算する。

(2) 使用法

CALL HQRIS/D (A, KA, N, E, V, NV, EPS, W, ILL)

引数	型と種類*	属性	内容
A	実数型 2次元配列	入力	実対称行列の対角線を含む右上半分を入力する。本ルーチンによって処理される。左下半分は保存される。
KA	整数型	入力	AとVの整合寸法(配列宣言における第1添字の値)。KA ≥ N
N	整数型	入力	Aの次数。Vの行数でもある。N ≥ 2
E	実数型 1次元配列	出力	すべての固有値が大きさの順に出力される。NV > 0ならば減少順、NV < 0ならば増大順。
V	実数型 2次元配列	出力	第I列に固有値E(I)に対する固有ベクトルが長さ1に正規化されて出力される。
NV	整数型	入力	NV は求めるべき固有ベクトルの数を表す。NV > 0(NV < 0)ならば代数的に最大(最小)のものから減少(増大)順に数える。 NV ≤ N
EPS	実数型	入力	QR法の収束判定定数。三重対角化された行列をTとすると T ・EPSが判定に用いられる。EPS > 0
W	実数型 1次元配列	作業領域	大きさ6N以上の1次元配列。
ILL	整数型	出力	ILL = 0 : 正常終了。 ILL = 30000 : 入力引数が制限を犯した。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 計算法

対称行列 A を、ハウスホルダー変換 H によって三重対角行列 $T = H^T A H$ に移す。

Tの固有値全部をQR法によって計算する。Tの指定された数の固有ベクトルを逆反復法によって求める。これを行列にまとめてUとする。Aの固有ベクトルは $V = H U$ によって計算される。

(4) 備考

1. 少ない所用時間で固有値を全部求め、固有ベクトルを全部又は一部分求める場合に最適である。

2. 固有値を全体の 1/4 以下程度求める場合には、ハウスホルダー・二分法に基づく HOBVS/D を用いる方が有利である。

参 考 文 献

- 1) 別府良孝, 二宮市三; “標準固有値問題の行列解法の比較”, 名古屋大学大型計算機センターニュース, Vol: 11, No 3, pp. 265—274 (1980)。

HOBSVS / D

Eigenvalue Analysis for Real Symmetric Matrices by Householder-Bisection Method
 ハウスホルダー・二分法による実対称行列の固有値解析

作成	二宮 市三	1977年4月	改訂	1981年4月
形式	サブルーチン 言語 ; FORTRAN サイズ ; 173, 171行			

(1) 概要

実対称行列をハウスホルダーの鏡像変換により三重対角化し、得られた三重対角行列の固有値をスツルム列の理論を応用した二分法によって求め、固有ベクトルを逆反復法によって計算する。HOBSVS (D) は単(倍)精度用である。

(2) 使用法

CALL HOBSVS/D (A, KA, N, E, NE, V, NV, EPS, W, ILL)

引 数	型 と 種 類*	属 性	内 容
A	実 数 型 2 次 元 配 列	入 力	実対称行列の対角線を含む右上半分を入力する。本ルーチンによって処理される。左下半分は保存される。
KA	整 数 型	入 力	AとVの整合寸法(配列宣言における第1添字の値)。 $KA \geq N$
N	整 数 型	入 力	Aの次数。Vの行数でもある。 $N \geq 1$
E	実 数 型 1 次 元 配 列	出 力	固有値が大ききの順に出力される。 $NE > 0$ ならば減少順, $NE < 0$ ならば増大順。
NE	整 数 型	入 力	絶対値で求める固有値の数を表す。 $NE > 0$ ($NE < 0$) ならば 代数的に最大(最小)のものより減少(増大)順に数える。 $NE \neq 0$
V	実 数 型 2 次 元 配 列	出 力	第1列に固有値E(1)に対する固有ベクトルが長さ1に正規 化されて出力される。
NV	整 数 型	入 力	絶対値で求める固有ベクトルの数を表す。NEによって定ま る順に端の固有値から数える。 $0 \leq NV \leq NE $
EPS	実 数 型	入 力	二分法の収束判定定数。三重対角化された行列をTとすると $\ T\ \cdot EPS$ が判定に用いられる。 $EPS > 0$
W	実 数 型 1 次 元 配 列	作 業 領 域	大きさ $6N$ 以上の1次元配列。
ILL	整 数 型	出 力	ILL = 0 : 正常終了。 ILL = 30000 : 入力引数が制限を犯した。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 計算法

行列 A をハウスホルダーの変換 H によって、三重対角行列 $T = H^T A H$ に移す。

T の固有値を、指定された順に数えて、端から指定された数だけ、スツルム列の理論を応用した二分法で求める。端から数えて指定された数の固有値に対応する固有ベクトルを逆反復法によって求める。

これらを列とする行列を U とする。 U は T の固有ベクトルであるので、これを A の固有ベクトル V に変換する。 $V = HU$ 。

(4) 備 考

対称行列の固有値を全部求めるには、本ルーチンよりも QR 法によるルーチン HOQRVS/D を用いる方がよい。また、固有値を全部求め、固有ベクトルを全部又は一部分求める場合には、QR・逆反復法によるルーチン HQRIS/D を用いる方が合理的である。

参 考 文 献

- 1) 別府良孝, 二宮市三; "標準固有値問題の行列解法の比較", 名古屋大学大型計算機センターニュース, Vol. 11, No 3, pp. 265-274 (1980)。

JACOBS / D

Eigenvalue Analysis for a Real Symmetric Matrix by Threshold Jacobi Method 閾ヤコビ法による実対称行列の固有値解析

作成	二宮 市三 1977年4月
形式	サブルーチン 言語 ; FORTRAN サイズ ; 88, 88行

(1) 概要

与えられた実対称行列の固有値と固有ベクトル全体を閾ヤコビ法で計算する。

(2) 使用法

CALL JACOBS/D (A, KA, N, EPS, V, ILL)

引数	型と種類*	属性	内容
A	実数型 2次元配列	入出力	実対称行列。対角線を含んで右上半分だけ与えればよい。出力として対角線上に固有値が出る。左下半分は保存される。
KA	整数型	入力	AとVの配列宣言における第1添字の値。 $KA \geq N$
N	整数型	入力	AとVの次数。 $N \geq 2$
EPS	実数型	入力	収束判定常数。入力行列Aの非対角元の絶対値の平均値を基準とし、この値のEPS倍を収束判定の規準とする。 $EPS > 0$
V	実数型 2次元配列	出力	各列に、対応するAの対角元に入っている固有値に対する固有ベクトルが入る。
ILL	整数型	出力	ILL = 0 : 正常終了。 ILL = 30000 : K, N, EPSに関する制限が破られたとき。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 性能

対称行列の固有値と固有ベクトルを全部求める良い方法として、多重固有値や近接固有値などに強い点が長所とされているが、近年、同じ目的のために、同じような長所を持つさらに強力なハウスホルダーQR法が現れたため、存在価値が薄れてきた。

(4) 備考

1. 固有値は、Aの対角線上に代数的に最大のものから減少順に並べ換えられている。
2. EPSの標準的な値としては、JACOBS (JACOBD) に対して 10^{-6} (10^{-16}) くらいが適当である。
3. 10程度の小規模な問題以外は、同じ目的のためのハウスホルダーQR法によるサブルーチンHOQRVS, HOQRVDを用いた方が、計算時間が大幅に短縮されるので有利である。

RHQRVS / D

Eigenvalue Analysis for Real Symmetric Band Matrices by Rutishauser-QR Method ルティスハウザー QR 法による実対称帯行列の固有値解析

作 成	二宮 市三 1977年4月
形 式	サブルーチン 言語 ; FORTRAN サイズ ; 150, 152行

(1) 概 要

実対称帯行列 B を、ルティスハウザー・シュワルツの方法により三重対角化し、これに QR 法を適用して固有値解析を行う。

(2) 使用法

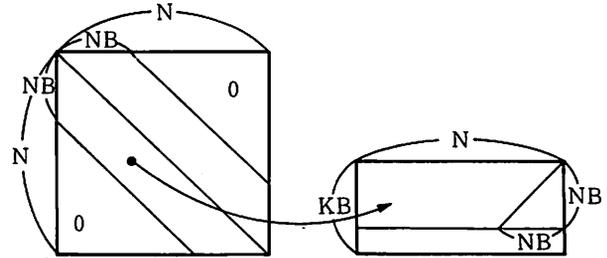
CALL RHQRVS/D (B, KB, N, NB, V, KV, E, F, EPS, IND)

引 数	型 と 種 類*	属 性	内 容
B	実 数 型 2 次 元 配 列	入 力	実対称帯行列の対角線を含み、左下半分を図示のように横長に長方形化して入力する。保存されない。
KB	整 数 型	入 力	Bの配列宣言での第1添字の値。KB ≥ NB
N	整 数 型	入 力	Bの次数(列数)。E, Fの大きさでもある。3 ≤ N
NB	整 数 型	入 力	Bの半帯幅(行数)。3 ≤ NB ≤ N
V	実 数 型 2 次 元 配 列	出 力	第J列にJ番目の固有値(J)に対応する固有ベクトルが、長さ1に規格化されて出力される。
KV	整 数 型	入 力	Aの配列宣言での第1添字の値。KV ≥ N
E	実 数 型 1 次 元 配 列	出 力	固有値が代数的に最大のものから減少順に整列されて出力される。
F	実 数 型 1 次 元 配 列 名	作 業 領 域	N個の要素を持つ配列名。
EPS	実 数 型	入 力	収束判定定数。Bを三重対角化したものをTとするとき、 T ・EPS/Nの形で用いられる。EPS > 0
IND	整 数 型	入 出 力	入力引数としては次の意味を持つ。 IND = 0 : 固有ベクトルを計算しない。 IND ≠ 0 : 固有ベクトルを全部計算する。 出力引数としては次の意味を持つ。 IND = 0 : 正常に計算が終了した。 IND = 30000 : 入力引数に関する制限が守られていない。 注意 : この引数に対する実引数には定数を用いてはいけない。

* 倍精度用のサブルーチンの場合、実数型をすべて倍精度実数型とする。

(3) 性能

ハウスホルダーの鏡像変換による三重対角化は非常にすぐれた方法であるが、一つの難点は、たとえ与えられた行列が実対称帯行列でも変換の途中でフル行列に広がってしまうことである。ルティスハウザー・



シュワルツの方法は、帯行列をその位置から拡がらせないで三重対角化するので上述の難点を克服できるが、帯幅を増すと計算量が多くなる。また、固有ベクトルを求めるためには、帯行列から三重対角行列への変換を表す直交行列を計算しなければならないが、そのためには、次数を N とするとき $N \times N$ の正方行列を新たに必要とする。以上の見地から、本ルーチンの存在理由は、帯幅の比較的小さい、大きな行列の固有値だけを計算することにあると考えられる。

(4) 備考

固有ベクトルを求めないときには、 V の所に何を書いてもよいが、 $KV \geq N$ は満たしておかなければならない。

RHBSVS / D

Eigenvalue Analysis for Symmetric Band Matrices by Rutishauser-Bisection Method

ルティスハウザー・二分法による対称帯行列の固有値解析

作成	二宮 市三	1977年4月	改訂; 1981年4月
形式	サブルーチン 言語; FORTRAN サイズ; 250, 250行		

(1) 概要

対称帯行列をルティスハウザー・シュワルツの方法で三重対角化し、これに二分法、逆反復法を適用して固有値解析を行う。

(2) 使用法

CALL RHBSVS/D (A, KA, N, NB, E, NE, V, KV, NV, VW, EPS, W, ILL)

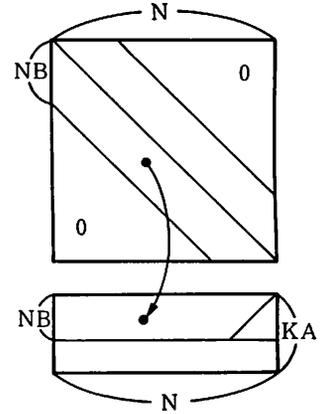
引数	型と種類*	属性	内容
A	実数型 2次元配列	入力	対称帯行列の対角線を含む左下半分を図のように長方形化して入力する。すなわち行列の I, J 要素を A (I-J+1, J) に入れる。本ルーチンで処理される。
KA	整数型	入力	Aの整合寸法(配列宣言における第1添字の値)。KA ≥ NB
N	整数型	入力	Aの次数。N ≥ 3
NB	整数型	入力	Aの半帯幅。NB ≥ 2
E	実数型 1次元配列	出力	固有値が大ききの順に出力される。NE > 0ならば減少順, NE < 0ならば増大順。
NE	整数型	入力	絶対値で求める固有値の数を表す。NE > 0 (NE < 0)ならば代数的に最大(最小)のものより減少(増大)順に数える。NE ≠ 0
V	実数型 2次元配列	出力	第1列に固有値E(1)に対する固有ベクトルが長さ1に正規化されて出力される。
KV	整数型	入力	V, VWの整合寸法。KV ≥ N
NV	整数型	入力	絶対値で求める固有ベクトルの数を表す。NEによって定まる順に端の固有値から数える。0 ≤ NV ≤ NE
VW	実数型 2次元配列	作業領域	大きさ N×N の2次元配列。固有ベクトルを計算しない場合(NV = 0)には不必要である。
EPS	実数型	入力	二分法の収束判定定数。三重対角化された行列をTとすると T · EPS が判定に用いられる。EPS > 0
W	実数型 1次元配列	作業領域	大きさ 6N 以上の1次元配列。
ILL	整数型	出力	ILL = 0 : 正常終了。 ILL = 30000 : 入力引数が制限を犯した。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 計算法

対称帯行列 A を、ルティスハウザー・シュワルツの直交変換 R によって、三重対角行列 $T = R^T A R$ に移す。 T の固有値問題 $Tu = \lambda u$ を二分法と逆反復法によって解く。 A の固有ベクトルは T の固有ベクトル u から $v = Ru$ として求められる。

ルティスハウザー・シュワルツの方法は帯行列の帯の内部だけで計算が行われるという利点を持っているが、その反面帯幅が大きくなると計算量が増大する。また、固有ベクトルを求めるには、変換行列 R を保存しなければならないが、そのためには次数を N とするとき、 $N \times N$ の正方行列を必要とする。以上の見地から、本ルーチンの存在理由は、帯幅の小さい大次元行列の固有値だけを計算することにある。



(4) 備考

1. 固有ベクトルを計算しない場合には V, VW のところには何を書いてもかまわないが、 $KV \geq N$ の条件は成り立つようにしなければならない。
2. 固有ベクトルを計算する場合に、もしも記憶容量を節約したいときには、 A と V との間に EQUIVALENCE の関係を設けることができる。なぜなら A と V とは同時に用いられることはないから。

GHQRVS / D

GHQRUS / D

Eigenvalue Analysis $Ax = \lambda Bx$ by Householder-QR Method

ハウスホルダー QR 法による $Ax = \lambda Bx$ 型の固有値解析

作成	二宮 市三 1977年4月
形式	サブルーチン 言語 ; FORTRAN サイズ ; 60, 60, 60, 60行

(1) 概要

実対称行列 A , 対称正値行列 B が与えられたとき, 固有値問題 $Ax = \lambda Bx$ の固有値全体と, 必要に応じて, 固有ベクトル全体を求める。 $B = U^T U$ とコレスキー分解を行い, これによって A を $\tilde{A} = U^{-T} A U^{-1}$ に変換し, 普通の固有値問題 $\tilde{A}y = \lambda y$ をハウスホルダー QR 法で解く。固有ベクトルが必要な場合には $x = U^{-1}y$ として \tilde{A} の固有ベクトル y を変換する。

(2) 使用法

CALL GHQRVS/D (A, B, KK, N, E, F, EPS, IND)

CALL GHQRUS/D (A, B, KK, N, E, F, EPS, IND)

引数	型と種類*	属性	内容
A	実数型 2次元配列	入出力	実対称行列全体 (右上半分でなくて) を入れる。ルーチン内で処理され, 右上半分に \tilde{A} が作られる。固有ベクトルを求める場合には, 各列に固有ベクトルが入る。ベクトルは, $x^T Bx = 1$ の意味で正規化されている。
B	実数型 2次元配列	入出力	対称正値行列を右上半分だけ入れる。ルーチン内で処理され, 右上半分に B のコレスキー分解成分 U が入る。左下半分は保存される。
KK	整数型	入力	A, B 両配列の宣言における第 1 添字の値。 $KK \geq N$
N	整数型	入力	A, B 両配列の次数。 $N \geq 2$
E	実数型 1次元配列	出力	N 個の要素を持つ 1 次元配列名。固有値が GHQRVS/D では代数的な意味で, GHQRUS/D では絶対値の意味で, 減少順に並べられる。
F	実数型 1次元配列	作業領域	N 個の要素を持つ 1 次元配列名。
EPS	実数型	入力	$ EPS $ は QR 法の収束判定常数であり, また, B のコレスキー分解のときの正値性判定常数でもある。 $EPS < 0$ としてコールすれば, B のコレスキー分解成分を再利用する。 $EPS \approx 0$
IND	整数型	入出力	入力引数としては次の意味を持つ。 IND = 0 : 固有値のみを計算する。 IND \approx 0 : 固有ベクトルをも計算する。 出力引数としては次の意味を持つ。 IND = 0 : 計算が正常に行われた。

引 数	型 と 種 類	属 性	内 容
			IND = 1: Bが正値でないと判定された。 IND = 30000: 入力引数についての制限が破られた。 この引数は入出力両用であるので、実引数として定数を用いてはいけない。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 計算法

対称正値行列 B を上三角行列 U によってコレスキー分解して

$$B = U^T U$$

とする。この U を用いて A から $\bar{A} = U^{-T} A U^{-1}$ を作ると、一般固有値問題 $Ax = \lambda Bx$ は、標準固有値問題 $\bar{A}\bar{x} = \lambda\bar{x}$ となる。この問題をハウスホルダー・QR法で解き、固有ベクトル x を $x = U^{-1}\bar{x}$ で求める。

(4) 備 考

1. 固有ベクトルを一部分だけ求める場合には、ハウスホルダー・QR・逆反復法 (GHQRIS/D) が有利なことがある。
2. B を一定にして、 A だけを変えて反復的に計算を行う場合は、 B のコレスキー分解成分を再利用するのがよい。引数 EPS の項を参照されたい。

GHQRIS / D

Eigenvalue Analysis of the Type $Ax = \lambda Bx$ by Householder-QR-Inverse Iteration Method ハウスホルダー・QR・逆反復法による $Ax = \lambda Bx$ 型の固有値解析

作成	二宮 市三 1981年4月
形式	サブルーチン 言語; FORTRAN サイズ; 250, 250行

(1) 概要

実対称行列 A , 実対称正値行列 B が与えられたとき, 固有値問題 $Ax = \lambda Bx$ の全部の固有値と一部分の固有ベクトルをハウスホルダー・QR・逆反復法によって求める。GHQRIS (D) は単 (倍) 精度用である。

(2) 使用法

CALL GHQRIS/D (A, B, KA, N, E, V, NV, EPS, W, ILL)

引 数	型 と 種 類*	属 性	内 容
A	実 数 型 2 次 元 配 列	入 出 力	実対称行列の対角線を含む右上半分を入力する。本ルーチンにより処理されて \tilde{A} となる (計算法参照)。左下半分は保存される。
B	実 数 型 2 次 元 配 列	入 力	実対称正値行列の対角線を含む右上半分を入力する。本ルーチンにより処理されてコレスキー分解成分 U となる (計算法参照)。左下半分は保存される。
KA	整 数 型	入 力	A, B, V の整合寸法 (配列宣言における第1添字の値)。 $KA \geq N$
N	整 数 型	入 力	A, B の次数。V の行数でもある。 $N \geq 2$
E	実 数 型 1 次 元 配 列	出 力	固有値が大ききの順に出力される。 $NV \geq 0$ ならば減少順, $NV < 0$ ならば増大順。
V	実 数 型 2 次 元 配 列	出 力	第1列に固有値 $E(1)$ に対する固有ベクトルが出力される。 $x^T Bx = 1$ の意味で正規化されている。
NV	整 数 型	入 力	$ NV $ は求めるべき固有ベクトルの数を表す。 $NV > 0$ ($NV < 0$) ならば代数的に最大 (最小) のものから減少 (増大) 順に数える。 $ NV \leq N$
EPS	実 数 型	入 力	QR法の収束判定定数。三重対角化された行列を T とすると $\ T\ \cdot EPS $ が判定に用いられる。 $EPS < 0$ のときBのコレスキー分解を省略する。 $EPS \neq 0$
W	実 数 型 1 次 元 配 列	作 業 領 域	大きさ $6N$ 以上の1次元配列。
ILL	整 数 型	出 力	ILL = 0 : 正常終了。 ILL = 1 : B が正値でない判定された。 ILL = 30000 : 入力引数が制限を犯した。

* 倍精度用のサブルーチンの場合, 実数型をすべて倍精度実数型とする。

(3) 計算法

対称正値行列 B を上三角行列 U によって、コレスキー分解して

$$B = U^T U \quad (U \text{ の対角成分はすべて正値とする})$$

とする。この U を用いて、 A から $\bar{A} = U^{-T} A U^{-1}$ を作ると、一般固有値問題 $Ax = \lambda Bx$ は、標準固有値問題 $\bar{A}\bar{x} = \lambda\bar{x}$ となる。この問題をハウスホルダー・QR・逆反復法で解き、固有ベクトル x を $x = U^{-1}\bar{x}$ で求める。

(4) 備考

1. 固有値を全体の 1/4 以下程度求める場合には、ハウスホルダー・二分法によるサブルーチン GH-BSVS/D を用いる方が有利である。
2. B を一定にして、 A だけを変えて反復的に計算を行う場合は、 B のコレスキー分解成分を再利用するのがよい。引数表の EPS の項を参照されたい。

参考文献

- 1) 別府良孝, 二宮市三; “標準固有値問題の行列解法の比較”, 名古屋大学大型計算機センターニュース, Vol. 11, No 3, pp. 265-274 (1980)。

GHBSVS / D

Eigenvalue Analysis of the Type $Ax = \lambda Bx$ by Householder-Bisection Method

ハウスホルダー・二分法による $Ax = \lambda Bx$ 型の固有値解析

作成	二宮 市三	1981年4月
形式	サブルーチン 言語; FORTRAN サイズ; 230, 230行	

(1) 概要

実対称行列 A , 実対称正値行列 B が与えられたとき, 固有値問題 $Ax = \lambda Bx$ の固有値と固有ベクトルをハウスホルダー・二分法により, 指定された数だけ求める。GHBSVS (D) は単 (倍) 精度用である。

(2) 使用法

CALL GHBSVS/D (A, B, KA, N, E, NE, V, NV, EPS, W, ILL)

引数	型と種類*	属性	内容
A	実数型 2次元配列	入力	実対称行列の対角線を含む右上半分を入力する。本ルーチンによって処理され \bar{A} となる (計算法参照)。左下半分は保存される。
B	実数型 2次元配列	入力	実対称正値行列の対角線を含む右上半分を入力する。本ルーチンによって処理され, コレスキー分解成分 U となる (計算法参照)。左下半分は保存される。
KA	整数型	入力	A, B, V の整合寸法 (配列宣言における第1添字の値)。 $KA \geq N$
N	整数型	入力	A, B の次数。V の行数でもある。 $N \geq 2$
E	実数型 2次元配列	出力	固有値が大きさの順に出力される。 $NE > 0$ ならば減少順, $NE < 0$ ならば増大順。
NE	整数型	入力	絶対値で求める固有型の数を表す。 $NE > 0$ ($NE < 0$) ならば 代数的に最大 (最小) のものより減少 (増大) 順に数える。 $NE \neq 0$
V	実数型 2次元配列	出力	第1列に固有値 $E(I)$ に対する固有ベクトルが $x^T Bx = 1$ の意味で正規化されて, 出力される。
NV	整数型	入力	絶対値で求める固有ベクトルの数を表す。 NE によって定まる 順に端の固有値から数える。 $0 \leq NV \leq NE $
EPS	実数型	入力	二分法の収束判定定数。三重対角化された行列を T とすると $\ T\ \cdot EPS $ が判定に用いられる。 $EPS < 0$ のとき B のコレスキー分解を省略する。 $EPS \neq 0$
W	実数型 1次元配列	作業領域	大きさ 6 N 以上の 1 次元配列。
ILL	整数型	出力	ILL = 0: 正常終了。 ILL = 1: B が正値でないと判定された。 ILL = 30000: 入力引数が制限を犯した。

* 倍精度用のサブルーチンの場合は, 実数型をすべて倍精度実数型とする。

(3) 計算法

対称正値行列 B を上三角行列 U によってコレスキー分解して

$$B = U^T U$$

とする。この U を用いて、 A から $\tilde{A} = U^{-T} A U^{-1}$ を作ると、一般固有値問題 $Ax = \lambda Bx$ は、標準固有値問題 $\tilde{A}\tilde{x} = \lambda\tilde{x}$ となる。この問題をハウスホルダー・二分法で解き、固有ベクトル x を $x = U^{-1}\tilde{x}$ で求める。

(4) 備考

1. 固有値を全部求める場合には、ハウスホルダー・QR法によるサブルーチン GHQRVS/D 又は GHQRIS/D を用いる方が有利である。
2. B を一定にして、 A だけを変えて反復的に計算を行う場合は、 B のコレスキー分解成分を再利用するのがよい。引数表の EPS の項を参照されたい。

参考文献

- 1) 別府良孝, 二宮市三; “標準固有値問題の行列解法の比較”, 名古屋大学大型計算機センターニュース, Vol. 11, No 3, pp. 265-274 (1980)。

JENNFS / S

JENNBS / D

GJENBS / D

Eigenvalue Analysis for Real Symmetric Matrices by Jennings' Simultaneous Iteration Method

ジェニングスの同時反復法による実対称行列の固有値解析

作成	二宮 市三	1981年4月
形式	サブルーチン	言語: FORTRAN サイズ: 141, 142, 151, 152, 184, 185行

(1) 概要

ジェニングスのベクトル加速法を伴う、ジェニングスの同時反復法により、実対称行列の一部の固有値と対応する固有ベクトルを求める。JENNFS/Dは標準型の固有値問題 $(A - \lambda I)x = 0$ を密行列 A に対して解くためのものであり、JENNBS/Dは同じ問題を帯行列 A に対して解くためのものである。GJENBS/Dは一般固有値問題 $(A - \lambda B)x = 0$ を A が帯行列、 B が正値帯行列の場合に解くためのものである。JENNFS (D), JENNBS (D), GJENBS (D)は単(倍)精度用のサブルーチンである。

(2) 使用法

CALL JENNFS/D (A, KA, N, L, M, V, E, C, W, EPS, ITER, ILL)

CALL JENNBS/D (A, KA, N, NB, L, M, V, KV, E, C, W, EPS, ITER, ILL)

CALL GJENBS/D (A, B, KA, N, NB, L, M, V, KV, E, C, W, EPS, ITER, ILL)

引数	型と種類*	属性	内容
A	実数型 1次元配列	入力	JENNFS/Dの場合は対称行列全体を入力する。JENNBS/D, GJENBS/Dの場合は帯行列の対角線を含む左下半分を図のように長方形化して入力する。すなわち行列のI, J要素をA(I-J+1, J)に入れる。固有値を絶対値の小さい方から求める場合は本ルーチンによりコレスキー分解される。
B	実数型 1次元配列	入力	帯行列をAと同様の方法によって入力する。固有値を絶対値の大きい方から求める場合は本ルーチンによりコレスキー分解される。
KA	整数型	入力	JENNFS/Dの場合はAとVの整合寸法。KA ≥ N JENNBS/Dの場合はAの整合寸法。 GJENBS/Dの場合はA, Bの整合寸法。 } KA ≥ NB
N	整数型	入力	A, Bの次数。N ≥ 2
NB	整数型	入力	A, Bの半帯幅。NB ≥ 2
L	整数型	入力	L は求める固有値と固有ベクトルの数を表す。L > 0ならば絶対値の大きい方から減少順。L < 0ならば絶対値の小さい方から増大順。1 ≤ L ≤ N
M	整数型	入力	試みのベクトルの本数。 L ≤ M ≤ N

引数	型と種類*	属性	内容
V	実数型 2次元配列	入出力	M本の固有ベクトルの初期ベクトルを入力する。最初の L 列に固有ベクトルが出力される。
KV	整数型	入力	Vの整合寸法。KV ≥ N
E	実数型 1次元配列	出力	固有値がLで指定された順序に出力される。
C	実数型 1次元配列	作業領域	大きさM ² 以上の1次元配列。
W	実数型 1次元配列	作業領域	JENNFS/D, JENNBS/Dの場合は大きさ3N以上, GJENBS/Dの場合は大きさ4N以上の1次元配列。
EPS	実数型	入力	収束判定定数。EPS > 0
ITER	整数型	入出力	入力： 反復数の上限。Nより小さいとき1000と設定する。 出力： 実際の反復数。
ILL	整数型	出力	ILL = 0： 正常。 ILL = 1： 反復数が上限を超えた。 ILL = 2： コレスキー分解が不可能であった。 ILL = 30000： 入力引数が制限を犯した。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 計算法

記述を簡単にするために、標準型問題を(標)、一般型問題を(般)、絶対値の大きい固有値を求める場合を(大)、絶対値の小さい固有値を求める場合を(小)と省略する。

- 固有ベクトルの初期値をVとする。Vはn行m列行列で、m個のベクトルを一つの行列にまとめたものである。mは求める固有値の数lに対して $l \leq m \leq n$ と取る(初期ベクトルのとり方については備考を参照のこと)。

(標, 小) Aを改訂コレスキー分解して $A = R^T D R$ とする。

(般, 小) Aをコレスキー分解して $A = \tilde{A}^T \tilde{A}$ とする。 $\tilde{A}V$ を作り、これを改めてVとする。

(般, 大) Bをコレスキー分解して $B = \tilde{B}^T \tilde{B}$ とする。 $\tilde{B}V$ を作り、これを改めてVとする。

- (標, 小) $U = R^{-1} D^{-1} R^{-T} V (= A^{-1} V)$ とする。

(標, 大) $U = AV$ とする。

(般, 小) $U = \tilde{A}^{-T} B \tilde{A}^{-1} V$ とする。

(般, 大) $U = \tilde{B}^{-T} A \tilde{B}^{-1} V$ とする。

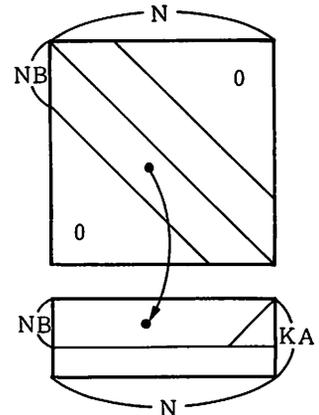
- $G = V^T U$ を作る。Gは $m \times m$ の対称行列である。

- Gを対角化して $P^T G P = Q$ とする。ただし、QはGの固有値

$\mu_1, \mu_2, \dots, \mu_m (|\mu_1| \geq |\mu_2| \geq \dots \geq |\mu_m|)$ を対角要素とする対角行列、Pは固有ベクトルを列とする直交行列である。

- $W = UP$ とする。

- $W^T W$ を作り、これをコレスキー分解して $W^T W = S^T S$ とする。Sは $m \times m$ 上三角行列である。



7. $V = WS^{-1}$ を作る。 V は $V^T V = I_m$ (m 次の単位行列) の意味の直交行列である。
8. 収束判定を行い(備考参照), 収束していない場合には, ジェニングスのベクトル加速を施し, その結果を改めて V として, 2.へもどる。
9. 収束した場合:
 - (小) $1/\mu_1, 1/\mu_2, \dots, 1/\mu_l$ を固有値とする。
 - (大) $\mu_1, \mu_2, \dots, \mu_l$ を固有値とする。
 - (標) V の初めの l 列を固有ベクトルとする。
 - (般, 小) $\bar{A}^{-1}V$ の初めの l 列を固有ベクトルとする。
 - (般, 大) $\bar{B}^{-1}V$ の初めの l 列を固有ベクトルとする。

(4) 備考

1. V の初期値は真の値に近いほど良いのは当然であるが, もしも, 初期値として利用できる情報がない場合には, 例えば単位行列の断片をとるのが一つの方法である。 m の選び方としては, 固有値を絶対値の順に並べたものを $\lambda_1, \lambda_2, \dots, \lambda_n$ とするとき, $|\lambda_m|/|\lambda_{m+1}| \gg 1$ (あるいは $|\lambda_m|/|\lambda_{m+1}| \ll 1$) となり, しかもなるべく l に近くとるのがよい。一回の反復あたりの計算量は大略 m に比例する。
2. 収束判定は, 固有ベクトルの成分について行われる。一般に固有値は固有ベクトルに比べて遥かに良い精度で収束する。特に固有値の分離の良い場合には約2倍の精度になる。したがって, 判定定数 EPS は, これを見越して幾分大きめにえらぶのが無難である。
3. 反復回数の上限 ITER の標準的な値は数百程度である。なお, この変数は入出力両用であるから実引数として定数を書いてはいけない。

参考文献

- 1) A. Jennings; "Matrix Computation for Engineers and Scientists", John Wiley, London, (1977).

SVDS / D

Singular Value Decomposition

特異値分解

作成	二宮 市三 1979年3月
形式	サブルーチン 言語 ; FORTRAN サイズ ; 205, 205行

(1) 概要

m 行 n 列 ($m \geq n \geq 1$) の行列 A を, m 行 n 列の直交行列 U , n 行 n 列の直交行列 V , n 行 n 列の対角行列 Σ により,

$$A = U \Sigma V^T$$

と分解する。ここに

$$U^T U = V^T V = V V^T = I_n (n \text{ 次の単位行列})$$

$$\Sigma = \text{diag}(q_1, q_2, \dots, q_n)$$

である。

U は, AA^T の大きい方から数えて n 個の固有値に対する正規直交固有ベクトル, V は $A^T A$ の正規直交固有ベクトルから成る。 Σ の対角要素は $A^T A$ の固有値の正平方根で

$$q_1 \geq q_2 \geq \dots \geq q_n \geq 0$$

となっている。 A の階数は 0 でない q_i の個数で与えられる。

(2) 使用法

CALL SVDS/D (A, KA, M, N, ISW, Q, U, KU, V, KV, W, ILL)

引 数	型 と 種 類*	属 性	内 容
A	実 数 型 2 次 元 配 列	入 力	特異値分解の対象となる行列。UまたはVを格納する領域として用いられない限り保存される。
KA	整 数 型	入 力	Aの配列宣言における第1添字の値。KA ≥ M
M	整 数 型	入 力	Aの行数。M ≥ N
N	整 数 型	入 力	Aの列数。N ≥ 1
ISW	整 数 型	入 力	0 ≤ ISW ≤ 3 ISW = 0 : U, Vともに計算しない。 ISW = 1 : Vのみ計算する。 ISW = 2 : Uのみ計算する。 ISW = 3 : U, Vともに計算する。
Q	実 数 型 1 次 元 配 列	出 力	特異値が大きい方から順に出力される。大きさNの1次元配列。
U	実 数 型 2 次 元 配 列	出 力	変換行列U。Aの上に重ね書きすることができる。M行N列の配列。
KU	整 数 型	入 力	Uの配列宣言における第1添字の値。KU ≥ M

引 数	型 と 種 類*	属 性	内 容
V	実 数 型 2 次 元 配 列	出 力	変換行列V。Aの上に重ね書きすることができる。 N行N列の配列。
KV	整 数 型	入 力	Vの配列宣言における第1添字の値。KV ≥ N
W	実 数 型 1 次 元 配 列	作 業 領 域	大きさNの1次元配列。
ILL	整 数 型	出 力	ILL=0: 正常終了。 ILL=20000: 30N回以上反復しても特異値分解が収束しない ILL=30000: 引数が制限を破った。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 性 能

参考文献¹⁾のp.418所載の、8行5列で $\sqrt{1248}$ 、20、 $\sqrt{384}$ 、0、0という特異値をもつ階数3の行列について実験した。SVDSによる特異値Qと変換行列V(その最後の2列は同次方程式 $Ax=0$ の二つの独立な解ベクトルである)の精度は約6桁であった。

(4) 使用例

上述のテストを行うためのプログラムを次に示す。

```

1      DIMENSION A(8,5),U(8,5),V(5,5),Q(5),W(5),R(5)
2      M=8
3      N=5
4      KA=8
5      KU=8
6      KV=5
7      EPS=1,E=-6
8      ISW=3
9      R(1)=SQRT(1248.)
10     R(2)=20.
11     R(3)=SQRT(384.)
12     R(4)=0.
13     R(5)=0.
14     READ(5,500) ((A(I,J),J=1,N),I=1,M)
15     500 FORMAT(5F4,0)
16     CALL SVDS(A,KA,M,N,ISW,Q,U,KU,V,KV,EPS,W,ICON)
17     WRITE(6,600) M,N,ISW,EPS,ICON,((A(I,J),J=1,N),I=1,M)
18     *,(Q(J),R(J),J=1,N),((U(I,J),J=1,N),I=1,M)
19     *,((V(I,J),J=1,N),I=1,N)
20     600 FORMAT(1H1//10X,'M =',I2,2X,'N =',I2,2X,'ISW =',I2
21     *,2X,'EPS =',1PE10.2,2X,'ICON =',I6//8(10X,5E13.5/)
22     *,5(10X,2E13.5//8(10X,5E13.5//5(10X,5E13.5//)
19     STOP
20     END

```

(5) 備 考

1. $M < N$ の場合は、Aの代わりに A^T を入れ、MとN、UとVとを交換して本サブルーチンを引用すれば、やはり $A = USV^T$ となる。
2. 特異値分解は行列Aが悪条件または階数落ちの場合に用いられる非常に有効な方法であるが、計算量が非常に多いのが欠点である。したがって、U、Vなどは必要でなければ計算しないようにするのが望ましい。
3. UまたはVはAの上に重ね書きすることができるので、Aを保存する必要がない場合にはUまたはVの所にAと同じものを書き、記憶容量を節約することが望ましい。

4. 一般化逆行列と最小二乗最小ノルム解のためにはそのための専用サブルーチンが用意されているので適当に取捨選択すべきである。

参 考 文 献

- 1) G.H. Golub, C. Reinsch; "Singular Value Decomposition and Least Squares Solutions", Numerische Mathematik, 14, pp.403-420, (1970).

NSHOUS / D

Nicer for Standard eigenvalue-problem by HOUseholder method

ハウスホルダー・二分割 QR・逆反復法による $Av = \lambda v$ 型の固有値解析

作成	別府 良孝, 二宮 市三 1981年12月	
形式	サブルーチン	言語; FORTRAN サイズ; 271, 272行

(1) 概要

密な実対称行列 A に関する標準固有値問題を準直接法で解く。

(2) 使用法

CALL NSHOUS/D(A, NMAX, N, NE, NV, EPS, IORD, E, V, ILL, W1, W2, W3, W4, W5,
W6, W7)

引数	型と種類*	属性	内容
A	実数型 2次元配列	入力	対角要素を含む右上半分に A_{ij} ($i \leq j$) を入力する。右上半分は変化するが、左下半分は保存される。
NMAX	整数型	入力	AとVの整合寸法。 $N \leq NMAX$
N	整数型	入力	Aの次数。 $2 \leq N$
NE	整数型	入力	求める固有値の数。 $0 < NE \leq N$ 。
NV	整数型	入力	求める固有ベクトルの数。 $0 \leq NV \leq NE \leq N$
EPS	実数型	入力	収束判定定数。標準値は 10^{-6} (NSHOUS) 又は、 10^{-10} (NSHOD)。)
IORD	整数型	入力	固有値の出力順序を指定する。 $IORD > 0$ ならば代数的減少順、 $IORD < 0$ ならば代数的増大順。
E	実数型 1次元配列	出力	I番目の固有値が、 $E(I)$ に出力される。 $IORD$ が正であれば、 $E(1) > E(2) > \dots > E(NE)$ 。 $IORD$ が負であれば、 $E(1) < E(2) < \dots < E(NE)$ 。
V	実数型 2次元配列	出力	$E(I)$ に対応する固有ベクトルが、 $v^T v = 1$ のごとく正規化されて第1列に出力される。
ILL	整数型	出力	$ILL = 0$: 正常終了。 $ILL = 300$: 引数異常。
W1	実数型 1次元配列	作業領域	寸法は N 以上必要。
W2	実数型 1次元配列	作業領域	寸法は N 以上必要。
W3	実数型 1次元配列	作業領域	寸法は N 以上必要。
W4	実数型 1次元配列	作業領域	寸法は N 以上必要。

引 数	型 と 種 類*	属 性	内 容
W 5	実 数 型 1 次 元 配 列	作 業 領 域	寸法はN以上必要。
W 6	実 数 型 1 次 元 配 列	作 業 領 域	寸法はN以上必要。
W 7	実 数 型 1 次 元 配 型	作 業 領 域	寸法はN以上必要。

(3) 計算法

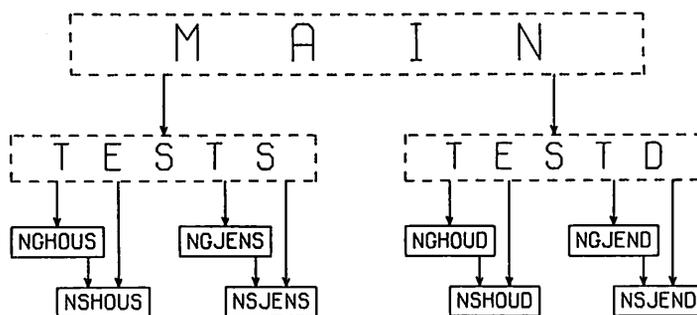
行列 A をハウスホルダー変換によって三重対角行列 T に変換してから、 $(NE/N) < 0.12$ ならば二分法によって NE 個の固有値を、 $(NE/N) \geq 0.12$ ならば無平方根QR法によって N 個の固有値を求め、ついで、逆反復法によって NV 個の固有ベクトルを求める。

(4) 備 考

1. NSHOUD は、SSL-II の DSEIG1 や DSEIG2 よりも 1.6 ~ 2.5 倍、N.LIB の HQRIID よりも 1.02 倍高速である。
2. 近似解が未知な場合に適している。
3. NSHOUS/D は、下図のごとく、NGHOUS/D、NSJENS/D、NGJENS/D と共に、高速固有値ルーチンパッケージ・NICER (Nagoya Iterative Computation Eigenvalue Routines) を構成する。
4. NICER は、北大、分子研、京大、九大などの計算センターや QCPE などのソフトウェアバンクにも登録されている。
5. NICER の効率的使用によって得られた研究成果を報ずる論文では、下記文献を適宜引用していただければ幸いである。

<NICER のエレメント構成>

破線で囲まれたものは、呼び出し用のエレメントである。



参 考 文 献

- 1) Y. Beppu and I. Ninomiya; "Manual of NICER", Quantum Chemistry Program Exchange (Indiana University), No.409 (1980).
- 2) Y. Beppu and I. Ninomiya; "NICER—Fast Eigenvalue Routines", Computer Physics Communications, Vol.23, pp.123–126 (1981).
- 3) Y. Beppu and I. Ninomiya; "HQR II—A Fast Diagonalization Subroutine", Computers and Chemistry, Vol.6, No.2, pp.87–91 (1982).

NGHOUS / D

Nicer for Generalized eigenvalue-problem by HOUSeholder method

双三角分解・ハウスホルダー・二分割QR・逆反復法による $Av = \lambda Bv$ 型の固有値解析

作成	別府 良孝, 二宮 市三 1981年12月	
形式	サブルーチン	言語; FORTRAN サイズ; 104, 105行

(1) 概要

密な実対称行列 A と密な実対称正値行列 B に関する一般固有値問題を準直接法で解く。

(2) 使用法

CALL NGHOUS/D (AB, NMAX, N, NE, NV, EPS, IORD, ICHO, BD, E, V, ILL, W 1,
W 2, W 3, W 4, W 5, W 6, W 7)

引 数	型 と 種 類*	属 性	内 容
AB	実 数 型 2 次 元 配 列	入 出 力	対角要素を含む右上半分に, $A_{ij} (i \leq j)$ を入力する。右上半分は変化する。左下半分に, $ICHO = 0$ ならば $B_{ij} (i > j)$ を, $ICHO = 1$ ならば B のコレスキー分解成分 L の非対角要素 $L_{ij} (i > j)$ を入力する。左下半分に, L_{ij} が出力される。
NMAX	整 数 型	入 力	AB と V の整合寸法。 $N \leq NMAX$
N	整 数 型	入 力	A と B の次数。 $2 \leq N$
NE	整 数 型	入 力	求める固有値の数。 $0 < NE \leq N$
NV	整 数 型	入 力	求める固有ベクトルの数。 $0 \leq NV \leq NE \leq N$
EPS	実 数 型	入 力	収束判定定数。標準値は 10^{-6} (NGHOUS) 又は 10^{-10} (NGH- OD)。
IORD	整 数 型	入 力	固有値の出力順序を指定する。 $IORD > 0$ ならば代数的減少順, $IORD < 0$ ならば代数的増大順。
ICHO	整 数 型	入 力	実対称正値行列 B の入力様式を指定する。AB と BD の項を参照のこと。
BD	実 数 型 1 次 元 配 列	入 出 力	BD (I) に, $ICHO = 0$ ならば B の対角要素 B_{ii} を入力し, $ICHO = 1$ ならば L の対角要素 L_{ii} の逆数を入力する。BD (I) に, L_{ii}^{-1} が出力される。
E	実 数 型 1 次 元 配 列	出 力	I 番目の固有値が, $E(I)$ に出力される。 $IORD$ が正であれば, $E(1) > E(2) > \dots > E(NE)$ 。 $IORD$ が負であれば, $E(1) < E(2) < \dots < E(NE)$ 。
V	実 数 型 2 次 元 配 列	出 力	$E(I)$ に対応する固有ベクトルが, $v^T B v = 1$ のごとく正規化されて第 I 列に出力される。
ILL	整 数 型	出 力	ILL = 0 : 正常終了。 ILL = 100 : B が非正値。 ILL = 300 : 引数異常。
W 1	実 数 型 1 次 元 配 列	作 業 領 域	寸法は, N 以上必要。

引 数	型 と 種 類*	属 性	内 容
W 2	実 数 型 1 次 元 配 列	作 業 領 域	寸法は、N以上必要。
W 3	実 数 型 1 次 元 配 列	作 業 領 域	寸法は、N以上必要。
W 4	実 数 型 1 次 元 配 列	作 業 領 域	寸法は、N以上必要。
W 5	実 数 型 1 次 元 配 列	作 業 領 域	寸法は、N以上必要。
W 6	実 数 型 1 次 元 配 列	作 業 領 域	寸法は、N以上必要。
W 7	実 数 型 1 次 元 配 列	作 業 領 域	寸法は、N以上必要。

(3) 計算法

まず、双三角分解法により、一般固有値問題 ($Av = \lambda Bv$) を標準固有値問題 ($\tilde{A}u = \lambda u$) に変換する。すなわち、行列 A を右上三角行列 R と左下三角行列 R^T の和に分解 ($A = R + R^T$) し、行列 B を左下三角行列 L と右上三角行列 L^T の積に分解 ($B = LL^T$) して、 $\tilde{A} = L^{-1}(R + R^T)L^{-T}$ を効率良く計算する。 A は実対称行列なので、 λ および正規直交ベクトル u を NSHOUS/D によって求め、ついで、広義直交ベクトル v を $v = L^{-T}u$ で求める。

(4) 備 考

1. NGHOUD は、SSL-II の DGSEG 2 よりも 1.3~2.0 倍、N.LIB の GHQRID や GHBSVD よりも 1.05 倍高速である。 L が既知な時、 $ICHO = 1$ とすると、約 5% 速くなる。
2. 近似解が未知な場合に適している。
3. 本ルーチンも、NICER を構成する。

NSJENS / D

Nicer for Standard eigenvalue-problem by JENnings method

ジェニングス法による $Av = \lambda v$ 型の固有値解析

作成	別府 良孝, 二宮 市三	1981年12月
形式	サブルーチン	言語; FORTRAN サイズ; 334, 335行

(1) 概要

密な実対称行列 A に関する標準固有値問題を同時反復法で解く。

(2) 使用法

CALL NSJENS/D (A, NMAX, N, NE, NV, EPS, ITER, ESHIFT, E, V, ILL, W 1, W2, U)

引 数	型 と 種 類*	属 性	内 容
A	実 数 型 2 次 元 配 列	入 力	対角要素を含む右上半分に A_{ij} ($i \leq j$) を入力する。 右上半分は、参照されるが変化しない。左下半分は、参照されず、変化もしない。
NMAX	整 数 型	入 力	A, V, U の整合寸法。 $N \leq NMAX$
N	整 数 型	入 力	A の次数。 $2 \leq N$
NE	整 数 型	入 力	求める固有値の数。絶対値最大のものより減少順に数える。 $0 < NE < N$
NV	整 数 型	入 力	求める固有ベクトルの数。 $0 < NE \leq NV < N$
EPS	実 数 型	入 力	収束判定定数。標準値は 10^{-6} (NSJENS) 又は 10^{-10} (NSJ - END)。
ITER	整 数 型	入 出 力	ジェニングス反復回数の上限 (標準値は 1 ~ 10) を、入力する。実際の反復回数が出力される。
ESHIFT	実 数 型	入 力	原点移動量 σ 。本ルーチン内では、 A ではなく $A' = A - \sigma I$ に関して同時反復法を適用しているため、 σ に近い固有成分は急速に希釈され、それ以外の固有成分は急速に濃縮される。標準値は、 $0.5 * (E(NV + 1) + E(N))$ の近似値。
E	実 数 型 1 次 元 配 列	入 出 力	絶対値順で I 番目に大きい固有値の近似値を、 $E(I)$ に入力する。絶対値順で I 番目に大きい固有値が、 $E(I)$ に出力される。 $ E(1) > E(2) > \dots > E(NE) $
V	実 数 型 2 次 元 配 列	入 出 力	NV 個の近似正規直交ベクトルを入力する。 $E(I)$ に対応する固有ベクトルが、 $v^T v = 1$ のごとく正規化されて出力される。
ILL	整 数 型	出 力	ILL = 0 : 正常終了。 ILL = 200 : 近似ベクトルの精度が悪くて収束未了。 ILL = 300 : 引数異常。
W 1	実 数 型 1 次 元 配 列	作 業 領 域	寸法は、N 以上必要。
W 2	実 数 型 1 次 元 配 列	作 業 領 域	寸法は、N 以上必要。

引 数	型 と 種 類*	属 性	内 容
U	実 数 型 2 次 元 配 列	作 業 領 域	行寸法は NMAX, 列寸法は NV 以上必要。

(3) 計算法

以下の手順で近似解を反復改良する。

1. 近似固有ベクトル行列 V_0 を用意する。
2. V_0 に $A' = A - \sigma I$ を乗算して X を求める。この時、ベキ乗法の原理により、 A' の絶対値の大きい固有成分が濃縮される。

$$X = A' V_0 = (A - \sigma I) V_0$$

3. $G = V_0^T X$ を作る。
4. G の固有値行列 E'_0 と固有ベクトル行列 W を求める。

$$W^T G W = E'_0$$

5. $Y = XW$ を作る。
6. $S = Y^T Y$ を作る。
7. S をコレスキー分解する。 $S = Z^T Z$
8. $V'_0 = YZ^{-1}$ を求める。 V'_0 は、 V_0 よりも V に近くなっている。
9. 許容精度に達していれば、 $V = V'_0$, $E = E'_0$ として計算を終える。逆に、未収束であれば、 $V_0 = V'_0$ として 2. にもどる。なお、 I は N 行 N 列の単位行列、 V_0 , X , Y , V'_0 は N 行 NV 列の行列、 G , W , S , Z は NV 行 NV 列の行列である。

(4) 備 考

1. NSJEND は、 $(ITER \times NV / N) < 0.5$ の時に、 NSHOUD より高速である。
2. N.LIB の JENNFS/D と同様、良好な近似解が既知な場合に適している。
3. 本ルーチンも NICER を構成する。

参 考 文 献

- 1) A. Jennings; "Matrix Computation for Engineers and Scientists", John Wiley and Sons, London (1977).
- 2) B. Parlett; "The Symmetric Eigenvalue Problem", Prentice-Hall, New Jersey (1980).
- 3) 別府良孝, 二宮市三; "高速固有値ルーチンパッケージ・NICER", 京都大学大型計算機センター広報, Vol. 13, No 5, pp. 378-386 (1980).

NGJENS / D

Nicer for Generalized eigenvalue-problem by JENnings method

双三角分解・ジェニングス法による $Av = \lambda Bv$ 型の固有値解析

作成	別府 良孝, 二宮 市三	1981年12月
形式	サブルーチン	言語; FORTRAN サイズ; 89, 90行

(1) 概要

密な実対称行列 A と密な実対称正値行列 B に関する一般固有値問題を同時反復法で解く。

(2) 使用法

CALL NGJENS/D (AB, NMAX, N, NE, NV, EPS, BD, IUUV, ITER, ESHIFT, E, V, U, ILL,
W 1, W 2)

引 数	型 と 種 類*	属 性	内 容
AB	実 数 型 2 次 元 配 列	入 出 力	対角要素を含む右上半分に, $A_{ij} (i \leq j)$ を入力する。右上半分は変化する。 左下半分に, B のコレスキー分解成分 L の非対角要素 $L_{ij} (i > j)$ を入力する。左下半分は, 変化しない。
NMAX	整 数 型	入 力	AB, V, U の整合寸法。 $N \leq NMAX$
N	整 数 型	入 力	A と B の次数。 $2 \leq N$
NE	整 数 型	入 力	求める固有値の数。絶対値最大のものより減少順に数える。 $0 < NE < N$
NV	整 数 型	入 力	求める固有ベクトルの数。 $0 < NE \leq NV < N$
EPS	実 数 型	入 力	収束判定定数。標準値は 10^{-6} (NSJENS) 又は 10^{-10} (NSJEND)。
BD	実 数 型 1 次 元 配 列	入 力	BD(1) に, L の対角要素 L_{ii} の逆数 L_{ii}^{-1} を入力する。保存される。
IUV	整 数 型	入 力	初期ベクトルの入力参照。様式を指定する。IUV = 0 ならば配列 V に入力されている近似広義直交ベクトル v_0 が, IUV = 1 ならば配列 U に入力されている近似正規直交ベクトル u_0 が, 同時反復法の初期ベクトルとして参照される。IUV = 0 の時の配列 U の内容, 及び, IUV = 1 の時の配列 V の内容は, 参照されない。
ITER	整 数 型	入 出 力	ジェニングス反復回数の上限(標準値は 1~10)を入力する。実際の反復回数が出力される。
ESHIFT	実 数 型	入 力	原点移動量 σ 。本ルーチン内では, \bar{A} ではなく $\bar{A}' = \bar{A} - \sigma \cdot I$ に関して同時反復法を適用しているので, σ に近い固有成分は急速に希釈され, それ以外の固有成分は急速に濃縮される。標準値は, $0.5 * (E(NV + 1) + E(N))$ の近似値。
E	実 数 型 1 次 元 配 列	入 出 力	絶対値順で I 番目に大きい固有値の近似値を, E(1) に入力する。絶対値順で I 番目に大きい固有値が, E(1) に出力される。 $ E(1) > E(2) > \dots > E(NE) $

引 数	型 と 種 類*	属 性	内 容
V	実 数 型 2 次 元 配 列	入 出 力	IUV=0 ならば NV 個の近似広義直交ベクトル v_0 を, IUV=1 ならば任意の量を入力する。E(I) に対応する広義直交ベクトルが, $v^T B v = 1$ のごとく正規化されて第 I 例に出力される。
U	実 数 型 2 次 元 配 列	入 出 力	IUV=0 ならば任意の量を, IUV=1 ならば NV 個の近似正規直交ベクトル u_0 を入力する。E(I) に対応する正規直交ベクトルが, $u^T u = 1$ のごとく正規化されて第 I 例に出力される。
ILL	整 数 型	出 力	ILL=0 : 正常終了。 ILL=100 : L_{ij} の入力エラー。 ILL=200 : 近似ベクトルの精度が悪くて収束未了。 ILL=300 : 引数異常。
W1	実 数 型 1 次 元 配 列	作 業 領 域	寸法は, N 以上必要。
W2	実 数 型 1 次 元 配 列	作 業 領 域	寸法は, N 以上必要。

(3) 計算法

以下の手順で近似解を反復改良する。

1. 双三角分解法 ($A = R + R^T$, $B = LL^T$) により $A = L^{-1}(R + R^T)L^{-T}$ を求め, $Av = \lambda Bv$ を $\tilde{A}u = \lambda u$ に変換する。
2. 初期正規直交ベクトル u_0 を用意する。IUV = 0 ならば, 配列 V に入力されている近似広義直交ベクトル v_0 に L^T をかけて u_0 を求める。IUV = 1 ならば, 配列 U に入力されている u_0 をそのまま用いる。
3. $Au = \lambda u$ を, u_0 を初期ベクトルとして, NSJENS/D によって解く。
4. $v = L^{-T}u$ を計算する。
5. NE 個の λ を配列 E に, NV 個の v を配列 V に, NV 個の u を配列 V に出力する。

(4) 備 考

1. NGJEND は, $(\text{ITER} \times \text{NV} / \text{N}) < 0.4$ の時に, NGHOUD より高速である。初期正規直交ベクトル u_0 が, 既知な時, IUV = 1 とすると, 約 5% 速くなる。
2. 良好な近似解が既知な場合, すなわち, よく似た実対称行列をいくつも対角化したい場合に適している。使用例と使用結果を以下に示す。

<NICER の使用例>

B が同一で A が異なる一般固有値問題 $Av = \lambda Bv$ を10回解くプログラム。この例は、分子の波動関数を逐次近似法で求める場合に対応している。

```

C   ITERATIVE COMPUTATION OF A*V=B*V*E   BY NICER
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION AB(10,10),BD(10)
      DIMENSION E(10),V(10,10)
      DIMENSION W1(10),W2(10),W3(10),W4(10),W5(10),W6(10),W7(10)
      DIMENSION U(10,10)
      NMAX=10
      N=8
      EPS=1.E-10

C
      DO 10 I=1,N
      AB(I,I)=7.2
      BD(I)=N+1-I
      DO 10 J=1,N
      IF(J.GT.I) AB(I,J)=3.0 / (FLOAT(I-J))**2
      IF(I.GT.J) AB(I,J)=N+1-MAX(I,J)
10  CONTINUE
      CALL NGHOUO(AB,NMAX,N,N,N,EPS,1,0,BD,E,V,ILL,W1,W2,W3,W4,W5,W6,W7)
      WRITE(6,100) ILL
100  FORMAT(1H1 //20X,4HILL=,I7)
      WRITE(6,200) ( E(I),I=1,N )
200  FORMAT(1H / 10(2X,10E12.3 /) )
      WRITE(6,300) ( ( V(I,J),J=1,N ),I=1,N )
300  FORMAT(1H / 10( 8F12.3/ ) )

C
      DO 1000 K=2,10
      FK=0.5*FLOAT(K-1)
      DO 20 I=1,N
      AB(I,I)=7.2 + FK
      DO 20 J=1,N
      IF(J.GT.I) AB(I,J)=(3.0+FK) / (FLOAT(I-J))**2
20  CONTINUE
      NE=N/2
      NV=N/2
      ESHIFT=0.5*( E(NV+1)+E(N) )
      ITER=10
      CALL NGJEND(AB,NMAX,N,NE,NV,EPS,BD,0,ITER,ESHIFT,E,V,U,ILL,W1,W2)
      WRITE(6,400) K, ITER,ILL
400  FORMAT(1H /10X,3H K=,I2,3X,5HITER=,I5,6H ILL=,I5)
      WRITE(6,200) ( E(I),I=1,NE )
      WRITE(6,500) ( ( V(I,J),J=1,NV ),I=1,N )
500  FORMAT(1H / 10( 4F12.3/ ) )
1000 CONTINUE
      STOP
      END

```

<NICER の使用結果の一部>

```

      ILL=      0
0.126D+02  0.116D+02  0.109D+02  0.102D+02  0.952D+01  0.729D+01  0.345D+01  0.465D+00
  0.057    0.355    -0.040    -0.595    -0.260    0.545    0.364    0.131
  0.005    -0.723    -0.464    0.838    0.708    -0.216    0.083    0.067
 -0.266    0.049    1.083    0.061    -0.816    -0.277    0.017    0.076
  0.169    0.941    -0.398    -0.209    0.828    -0.420    -0.133    0.062
  0.448    -0.632    -0.617    -0.069    -0.978    -0.087    -0.218    0.052
 -0.567    -0.450    0.562    -0.664    0.786    0.197    -0.247    0.039
 -0.511    0.652    -0.420    0.837    -0.440    0.440    -0.217    0.027
  1.225    -0.156    0.584    -0.006    0.295    0.204    -0.083    0.007

```

PACKAGE-NAME : NICER(NAGOYA ITERATIVE COMPUTATION EIGENVALUE ROUTINES)
 (VERSION-1,LEVEL-3) MODIFIED ON MARCH 1981
 REFERENCE : Y.BEPPU AND I.NINOMIYA;QUANTUM CHEMISTRY PROGRAM EXCHANGE,NO.409(1980)

```

      K= 2  ITER= 4  ILL= 0
0.131D+02  0.117D+02  0.112D+02  0.988D+01
  0.099    0.203    -0.456    -0.444
 -0.124    -0.663    0.340    0.973
 -0.202    0.399    0.808    -0.616
  0.328    0.718    -0.665    0.094
  0.281    -0.856    -0.399    0.183
 -0.666    -0.245    0.281    -0.790
 -0.323    0.718    -0.104    0.946
  1.184    -0.266    0.532    -0.339

```

```

      K= 3  ITER= 3  ILL= 0
0.136D+02  0.120D+02  0.115D+02  0.935D+01
  0.093    -0.151    -0.511    -0.297
 -0.172    -0.235    0.773    0.897
 -0.119    0.740    0.311    -0.800
  0.383    0.190    -0.922    0.192
  0.151    -0.952    0.158    0.173
 -0.697    -0.013    0.279    -0.742
 -0.199    0.647    -0.371    0.966
  1.156    -0.118    0.592    -0.404

```

```

      K= 4  ITER= 6  ILL= 0
0.143D+02  0.126D+02  0.117D+02  0.918D+01
  0.067    0.322    0.395    0.126
 -0.180    -0.077    -0.829    0.498
 -0.041    -0.770    0.024    -0.884
  0.392    0.137    0.923    0.164
  0.046    0.866    -0.425    -0.054
 -0.701    -0.128    -0.245    -0.393
 -0.108    -0.555    0.471    0.969
  1.135    0.074    -0.620    -0.288

```

QUADRS / D / Q

CUBICS / D / Q

QUARTS / D / Q

Solution of Low Order Polynomial Equations with Real Coefficients

低次実係数代数方程式の解法

作成	二宮 市三	1977年4月
形式	サブルーチン	言語 ; FORTRAN サイズ ; 24, 27, 27, 40, 41, 41, 46, 47, 47行

(1) 概要

QUADRS (D, Q), CUBICS (D, Q), QUARTS (D, Q) は実数係数の二次、三次及び四次代数方程式のすべての根を計算するための単 (倍, 4 倍) 精度用サブルーチン。

(2) 使用法

$$\text{CALL } \left\{ \begin{array}{l} \text{QUADRS/D/Q} \\ \text{CUBICS/D/Q} \\ \text{QUARTS/D/Q} \end{array} \right\} (\text{A, X, Y, ILL})$$

引数	型と種類*	属性	内容
A	実数型 1次元配列	入力	代数方程式の係数を最高次の係数から順に入力する。 $A(1) \neq 0$
X	実数型 1次元配列	出力	代数方程式の根の実部が出力される。
Y	実数型 1次元配列	出力	代数方程式の根の虚部が出力される。
ILL	整数型	出力	ILL = 0 : 正常終了。 ILL = 30000 : $A(1) = 0$ のとき。

* 倍 (4 倍) 精度用のサブルーチンの場合は、実数型をすべて倍 (4 倍) 精度実数型とする。

(3) 計算法

1. 二次方程式は絶対値の小さい根のみを根の公式で計算し、他の根は2根の積と係数の関係を用いて求める。
2. 三次方程式はカルダノの方法による。
3. 四次方程式はフェラリの方法による。

(4) 備考

五次以上の代数方程式は GJMNS/D/Q を用いて解くことができる。

GJMNKS / D / Q

Solution of Polynomial Equations with Real Coefficients by Garside-Jarrat-Mack Method

ガーサイド・ジャラット・マック法による実係数代数方程式の解法

作成	二宮 市三	1977年4月
形式	サブルーチン 言語 ; FORTRAN サイズ ; 128, 130, 130行	

(1) 概要

GJMNKS (D, Q) は実係数代数方程式のすべての根を求めるための単(倍, 4倍)精度用サブルーチン。複素係数の代数方程式の解法として定評のある Garside-Jarrat-Mack¹⁾の方法を実係数の方程式の解法として, 二宮・門脇²⁾が改良したものをサブルーチン化したものである。原方法のもつ頑丈さと複素計算の実数化によるスピード・アップが組み合わされており, また, 三次, 四次の方程式の解法としてカルダノ法, フェラリ法が取り入れられているので, 汎用の代数方程式ルーチンとして推奨できる。

(2) 使用法

CALL GJMNKS/D/Q (A, N, X, Y, ILL)

引数	型と種類*	属性	内容
A	実数型 1次元配列	入力	代数方程式の係数を最高次の係数から順に入力する。保存されない。A(1) ≠ 0
N	整数型	入力	代数方程式の次数。N ≥ 1
X	実数型 1次元配列	出力	代数方程式の根の実部が出力される。根は, おおむね絶対値の小さいものから求められ, X(N), X(N-1) … と逆順に格納される。
Y	実数型 1次元配列	出力	代数方程式の根の虚部が出力される。根の計算順序と格納の仕方はXに同じ。
ILL	整数型	出力	ILL = 0 : 正常終了。 ILL = 30000 : N < 1 又は A(1) = 0 のとき。 ILL = K : 減次されたK次方程式を処理中に200回反復しても未収束のとき。

* 倍(4倍)精度用のサブルーチンの場合, 実数型をすべて倍(4倍)精度実数型とする。

(3) 計算法

重根や近接根などに対しても, 原理的には収束率が変わらないという原方法の良さを受けついでいて, ほとんどの実際的な方程式に対して, 途中で収束不能となることなく, 方程式の条件が許す限りの精度で根が計算される。この意味で非常に頑丈であると言える。また, 原方法では, 実係数の方程式に対しても複素計算を必要としたが, これを実計算で行えるように改良したため, 計算量は半減し, スピードが倍加された。

(4) 備考

1. 代数方程式は悪条件となることが多いので, よほど低次の場合を除いては, 倍精度用ルーチン

GJMKNKD を用いるのが安全である。

2. 二次、三次及び四次方程式は本ルーチンによるよりもそれぞれ専門のルーチンを用いて解く方が有利である。

参 考 文 献

- 1) G.R. Garside, P. Jarrat and C. Mack; "A New Method for Solving Polynomial Equations," *Computer Journal*, Vol.11 (1968).
- 2) 二宮市三, 門脇幸平; "実係数代数方程式の一解法", 情報処理学会第16回大会講演予稿集, p. 445 (1975)。

BRENDS / D

Solution of Nonlinear Equations by Brent's Method

ブレントの方法による非線型方程式の解法

作成	二宮 市三 1980年4月
形式	サブルーチン 言語; FORTRAN サイズ; 50, 51行

(1) 概要

解の存在区間が与えられたとき、与えられた非線型方程式のその区間内の一つの解を Brent の方法で求める。

(2) 使用法

CALL BRENDS/D (AO, BO, FUN, EPS, MAX, X, FX, N, ILL)

引 数	型 と 種 類*	属 性	内 容
AO	実 数 型	入 力	解の存在する区間の左端。
BO	実 数 型	入 力	解の存在する区間の右端。
FUN	実 数 型 関数副プログラム	入 力	解くべき方程式を $f(x) = 0$ とするとき、 $f(x)$ に相当する関数副プログラム。これに対する実引数としての関数副プログラムを使用者が用意しなければならない。
EPS	実 数 型	入 力	解の精度判定定数。
MAX	整 数 型	入 力	関数計算回数の上限。MAX \geq 3
X	実 数 型	出 力	解の近似値。
FX	実 数 型	出 力	解の近似値での $f(x)$ の値。
N	整 数 型	出 力	関数計算回数。
ILL	整 数 型	出 力	ILL = 0 : 正常終了。 ILL = 20000 : 関数計算回数 N が MAX を超えた。 ILL = 30000 : AO と BO での関数値が同符号のとき、 又は MAX < 3。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 計算法

参考文献¹⁾を参照のこと。

(4) 使用例

方程式 $f(x) = \cos x - x = 0$ の $(0, \pi/2)$ の内部の解を計算するためのプログラムを示す。

```

C      TEST FOR BRENTS
      EXTERNAL FUN
      A=0.0
      B=1.5708
      MAX=100
      EPS=1.E-5
      CALL BRENDS(A,B,FUN,EPS,MAX,X,Y,N,ILL)
      WRITE(6,600) A,B,EPS,X,Y,N,ILL
  
```

```

600 FORMAT(1H ,2E13.5,E11.3,E13.5,E11.3,2I6)
STOP
END
C FUNCTION SUBPROGRAM
FUNCTION FUN(X)
FUN=COS(X)-X
RETURN
END

```

(5) 備考

1. Brentの方法は二分法を基礎にしているので確実に解を求めることができる。
2. 必要に応じて、一次又は二次の逆補間を援用しているので収束が速い。
3. 関数 $f(x)$ は連続であればよく、滑らかさは必要ではない。
4. 代数方程式は専用のサブルーチン GJMKNKS/D 等を用いた方が有利である。

参考文献

- 1) P.R. Brent; "Algorithms for Minimization without Derivatives," Prentice Hall, London, pp.47-60 (1973).

BROYDS/D

Solution of Systems of Nonlinear Equations by Broyden's Method

ブロイデンの方法による非線形連立一次方程式の解法

作成	二宮 市三 1977年4月
形式	サブルーチン 言語; FORTRAN サイズ; 59, 71行

(1) 概要

初期値が与えられたとき、非線形連立方程式 $f_i(x_1, \dots, x_n) = 0$ ($i = 1, 2, \dots, n$) を Broyden の反復法で解くためのサブルーチン副プログラムである。

(2) 使用法

CALL BROYDS/D (X, N, H, KH, FN, LF, NF, EPS, FM, ILL)

引 数	型 と 種 類*	属 性	内 容
X	実 数 型 1 次 元 配 列	入 出 力	初期ベクトルを入力すると解ベクトルが出力される。
N	整 数 型	入 力	方程式の元数。 $0 < N \leq 100$
H	実 数 型 2 次 元 配 列	作 業 領 域	$N \times N$ の大きさを要する。
KH	整 数 型	入 力	Hの配列宣言における第1添字の値。 $KH \geq N$
FN	サブルーチン	入 力	位置ベクトルXが与えられたとき、N個の方程式の値から成るベクトルYを計算するためのFN(X, Y)の形のサブルーチン。この引数に対する実引数は、本ルーチンを引用するプログラム単位でEXTERNAL宣言を必要とする。
LF	整 数 型	入 力	関数サブルーチン引用回数の上限。 $LF > N+1$
NF	整 数 型	出 力	関数サブルーチン引用回数。
EPS	実 数 型	入 力	収束判定常数。 $EPS > 0$
FM	実 数 型	出 力	方程式の残差2乗平均の平方根。
ILL	整 数 型	出 力	ILL = 0 : 正常終了。 ILL = 1 : NF > LF となっても収束しないとき。 ILL = 30000 : 入力引数が条件を満足しないとき。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 計算法

参考文献¹⁾を参照のこと。

(4) 使用例

Freudenstein-Roth の問題 (解: $X(1) = 5.0$, $X(2) = 4.0$) を解くためのプログラムの主要部を示す。

```

DIMENSION H(2,2),X(2)
EXTERNAL FREUDE
KH=2
N=2
LF=1000
EPS=1.E-5
X(1)=15.0
X(2)= 3.0
CALL BROYDS(X,N,H,KH,FREUDE,LF,NF,EPS,FM,ILL)
:
END

SUBROUTINE FREUDE(X,F)
DIMENSION X(2),F(2)
F(1)=X(1)-13.+(5.-X(2))*X(2)-2.)*X(2)
F(2)=X(1)-29.+(X(2)+1.)*X(2)-14.)*X(2)
RETURN
END

```

(5) 備 考

1. 非線形連立方程式は一般に多くの解を持つので、得られた解が果たして目的の解であるかどうかは検討の必要がある。目的の解に確実に収束させるためには、良い初期値が是非とも必要である。
2. 最小値（又は最大値）が極値である場合の最小化問題は、勾配ベクトルに関して非線形連立方程式となり、逆に、非線形連立方程式 $f_i = 0$ ($i = 1, \dots, n$) は $\sum f_i^2$ を考えれば最小化問題となる。これを考慮して、それぞれの場合に適する計算法及びプログラムを選択すべきである。
3. 本ルーチンは、その中で逆行列ルーチン MINVS/D を引用している。

参 考 文 献

- 1) C.G. Broyden; "A Class of Methods for Solving Nonlinear Simultaneous Equations", *Math. Comp.*, Vol.19, pp.577-593 (1965).

MINSXS / D

Minimization of Functions by Simplex Method

シンプレックス法による関数の最小化

作 成	二宮 市三	1977 年 7 月
形 式	サブルーチン 言語 ; FORTRAN サイズ ; 98, 99行	

(1) 概 要

初期値が与えられたとき、多変数関数の最小点を Nedler-Mead のシンプレックス法で求めるためのサブルーチン。計算には関数値だけが必要である。また、関数は連続であればよく、滑らかさは必要ではない。

(2) 使用法

CALL MINSXS/D (X, N, P, KP, FUNC, LF, NF, EPS, FM, ILL)

引 数	型 と 種 類*	属 性	内 容
X	実 数 型 1 次 元 配 列	入 出 力	最小点に対する初期値を入力すると、最小点が出力される。
N	整 数 型	入 力	変数の数, すなわち, X の成分の数。 $0 < N \leq 100$
P	実 数 型 2 次 元 配 列	作 業 領 域	N 行 N+1 列。この各列にシンプレックスを形成する N+1 個の点の座標が入る。初期シンプレックスは、初期値 X と X の各座標成分を10%増したN個の点（その成分が0なら0.1だけ増す）で作る。しかし、使用者が本ルーチンを引用する前に、適当にPの位置に作ってもよい。この選択は引数NFを通じて行う。
KP	整 数 型	入 力	P の配列宣言における第1添字の値。 $KP \geq N$
FUNC	実 数 型 関数副プログラム	入 力	最小化の対象となる関数。これに対する実引数としての関数は、FUNC (X) の形の関数副プログラムとして、使用者が用意する。この関数名は EXTERNAL 宣言を要する。
LF	整 数 型	入 力	関数の計算回数の上限。 $LF > N$
NF	整 数 型	入 出 力	入力としては、 $NF \geq 0$ ならば初期シンプレックスの作り方をルーチンに任せることを、 $NF < 0$ ならば使用者が自ら作ったことを意味する。出力としては、関数の計算回数となる。入出力変数であるから、実引数として定数を書いてはいけない。
EPS	実 数 型	入 力	収束判定常数。 $EPS > 0$
FM	実 数 型	出 力	関数の最小値。
ILL	整 数 型	出 力	ILL = 0 : 正常終了。 ILL = 1 : $NF > LF$ となっても収束しないとき。 ILL = 30000 : 入力引数が制限条件を満たさないとき。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 計算法

参考文献¹⁾を参照のこと。

(4) 使用例

Rosenbrock の問題 (最小点: $X(1) = 1.0$, $X(2) = 1.0$) を解くためのプログラムの主要部を示す。

```
DIMENSION P(2,3),X(2)
EXTERNAL ROSEN
N=2
KP=2
NF=1
LF=1000
EPS=1.E-5
X(1)=-1.2
X(2)=1.0
CALL MINSXS(X,N,P,KP,ROSEN,LF,NF,EPS,FM,ILL)
:
FND

FUNCTION ROSEN(X)
DIMENSION X(2)
ROSEN=100.*(X(1)*X(1)-X(2))**2+(1.-X(1))**2
RETURN
END
```

(5) 備考

1. 本ルーチンで得られるものは、一般に局所的な最小値に過ぎない。真の最小値を確実に求めるためには、良い初期値が必要である。
2. 収束が遅いので、あまり高次元の問題には適しない。
3. 実引数としての関数の名前は、本ルーチンを引用するプログラム単位で、EXTERNAL 宣言しなければならない。
4. 関数が滑らかで、勾配の計算が容易な場合は、DFP 法のサブルーチン FLPOWS を用いる方が有利である。
5. 非線形連立方程式 $f_1 = 0$, $f_2 = 0$, ..., $f_n = 0$ を、本ルーチンを用いて最小化問題として扱うとき、 $F = \sum f_i^2$ とするよりは $F = \sum |f_i|$ とする方がよい。

参考文献

- 1) J.A. Nedler & R. Mead; "A Simplex Method for Function Minimization", *Computer Journal*, Vol.7, pp.308-312 (1965).

FLPOWS / D

Minimization of Functions by Davidon-Fletcher-Powell Method

ダビドン-フレッチャー-パウエル法による関数の最小化

作成	二宮 市三 1977年7月
形式	サブルーチン 言語; FORTRAN サイズ; 92, 105行

(1) 概要

初期値が与えられたとき、多変数関数の最小点を Davidon-Fletcher-Powell の方法で求めるためのサブルーチン副プログラム。関数値の外に勾配ベクトルの値を必要とする。

(2) 使用法

CALL FLPOWS/D (X, N, B, KB, FUNC, GRAD, LF, NF, FLB, EPS, FM, ILL)

引数	型と種類*	属性	内容
X	実数型 1次元配列	入力	初期値ベクトルを入力すると、解ベクトル（最小点）が出力される。
N	整数型	入力	変数の数、すなわち、Xの成分の数。 $0 < N \leq 100$
B	実数型 2次元配列	作業領域	$N \times N$ の大きさを要する。始めに単位行列が作られ、反復に伴う更新 (Updating)により、最小点におけるヘッセ行列の逆に収束する。
KB	整数型	入力	Bの配列宣言における第1添字の値。 $KB \geq N$
FUNC	実数型 関数副プログラム	入力	最小化の対象となる関数。FUNC (X) の形の関数副プログラムとして使用者が用意する。実引数名は EXTERNAL 宣言を必要とする。
GRAD	サブルーチン	入力	関数 FUNC の勾配ベクトル G を計算するためのサブルーチン。GRAD (X, G) の形のサブルーチンとして使用者が用意する。実引数名は EXTERNAL 宣言を必要とする。
LF	整数型	入力	関数引用回数の上限。 $LF \geq N$
NF	整数型	出力	関数引用回数 (FUNC と GRAD の引用回数は同じ)。
FLB	実数型	入力	関数の最小値の下限。
EPS	実数型	入力	収束判定常数。 $EPS > 0$
FM	実数型	出力	関数の最小値。
ILL	整数型	出力	ILL = 0 : 正常終了。 ILL = 1 : $NF > LF$ となっても収束しないとき。 ILL = 30000 : 入力引数が条件を満足しないとき。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 計算法

参考文献^{1), 2)}を参照のこと。

(4) 使用例

Rosenbrock の問題を (最小点: $X(1) = 1.0$, $X(2) = 1.0$) 解くためのプログラムの主要部を示す。

```
DIMENSION X(2),B(2,2)
EXTERNAL ROSEN,GROSEN
N=2
KB=2
LF=1000
FLB=0.
EPS=1.E-5
X(1)=-1.2
X(2)=1.0
CALL FLPOWS(X,N,B,KB,ROSEN,GROSEN,LF,NF,FLB,EPS,FM,ILL)
:
FND

FUNCTION ROSEN(X)
DIMENSION X(2)
ROSEN=100.*(X(1)*X(1)-X(2))**2+(X(1)-1.)**2
RETURN
FND

SUBROUTINE GROSEN(X,G)
DIMENSION X(2),G(2)
G(2)=200.*(X(2)-X(1)*X(1))
G(1)=2.*(X(1)-1.-G(2)*X(1))
RETURN
FND
```

(5) 備考

1. 本ルーチンで得られるものは、一般に局所的な最小値に過ぎない。真の最小値を確実に求めるためには、是非とも良い初期値が必要である。
2. 勾配ベクトルの計算が不可能又は非常に困難な場合は、勾配ベクトルの計算を必要としない方法によるべきである。

参考文献

- 1) R. Fletcher & M.J.D. Powell; "A Rapidly Convergent Descent Method for Minimization", *Computer Journal*, Vol.6, pp.163-168 (1963).
- 2) P.J. Reddy, H.J. Zimmermann & Asghar Hussain; "Numerical Experiments on DFP-Method, A Powerful Function Minimization Technique", *Journal of Computational & Applied Mathematics*, Vol.1, pp.255-265 (1975).

NOLLS1

Subroutine for Nonlinear Least Squares by a Quasi-Newton Method

準ニュートン法による非線形型最小二乗法サブルーチン

作成	田辺 國士, 上田 澄江	1981年3月
形式	サブルーチン	言語; FORTRAN サイズ; 772行

(1) 概要

n 個の変数 $x_i, i = 1, \dots, n$, について非線形な関数 $f_j(x_1 \dots x_n), j = 1, \dots, m$, に対して

$$\sum_{j=1}^m f_j^2(x_1 \dots x_n)$$

を最小にする $x_i, i = 1, \dots, n$, を求める。特に非線形性の強い問題に有効である。

利用者は $f_j(x_1, \dots, x_n), j = 1, \dots, m$, の値を計算するサブルーチン (MODEL F) を用意するだけでよい。できれば, f_j の x_i についての1次微係数 $\partial f_j / \partial x_i$ の値を計算するサブルーチン (MODEL D) も用意した方が精度のよい結果が得られる。

(2) 使用法

CALL NOLLS1 (MAXM, MAXN, M, N, X, ITMAX, NFEMAX, FTOL, XTOL, LDERIV, NPRINT, FF2, F, DF, ITER, NFE, NDE, INFORM, XO, DX, FO, DFO, H, SL, D, S, Y, R, W1, W2, W3, W4)

引数	型と種類*	属性	内容
MAXM	整数型	入力	DFの整合寸法(配列宣言における第1添字の値)。 MAXM \geq N
MAXN	整数型	入力	Hの整合寸法。MAXN \geq N
M	整数型	入力	非線形な関数 $f_j(x)$ の数, m 。
N	整数型	入力	未知パラメータ x_i の数, n 。
X	実数型 1次元配列	入出力	未知パラメータ x_i の初期値を入れると最終値が出力される ($i = 1, 2, \dots, n$)。
ITMAX	整数型	入力	反復回数の上限。
NFEMAX	整数型	入力	関数評価回数の上限。
FTOL	実数型	入力	関数値に関する収束判定定数。 $f_j(x)$ の全部の値が FTOL 以下になるならば, 反復を終了する。 $0 \leq$ FTOL
XTOL	実数型	入力	未知パラメータ x_j に関する収束判定定数。 $0 \leq$ XTOL
LDERIV	整数型	入力	f_j の x_i についての1次微係数を与えるサブルーチン MODEL D を用意するか否かを指定する。 1: MODEL D を用いる。 0: MODEL D を用いない。
NPRINT	整数型	入力	各反復計算ごとに印刷する変数を指定する。 0: 何も印刷しない。

引 数	型 と 種 類*	属 性	内 容
			1 : 2乗和と x_i 。 2 : 2乗和, x_i, f_j 。 3 : 2乗和, $x_i, f_j, \partial f_j/x_i$ 。
FF2	実 数 型	出 力	2乗和の値。
F	実 数 型 1 次 元 配 列	出 力	残差 f_j の値。
DF	実 数 型 2 次 元 配 列	出 力	1次微係数 $\partial f_j/x_i$ の値。DF (MAXM, N)。
ITER	整 数 型	出 力	実際の反復回数。
NFE	整 数 型	出 力	実際の関数評価回数。
NDE	整 数 型	出 力	実際の1次微係数の評価回数 (MODEL Dの呼出し回数)。
INFORM	整 数 型	出 力	収束状況の情報を出力する。INFORM = 1 のとき, (3)計算 法での②の条件を満たした。それ以外は INFORM = 0。
XO, DX	実 数 型 1 次 元 配 列	作 業 領 域	XO (N), DX (N)。
FO	実 数 型 1 次 元 配 列	作 業 領 域	FO (M)。
DFO	実 数 型 2 次 元 配 列	作 業 領 域	DFO (MAXM, N)。
H, SL	実 数 型 2 次 元 配 列	作 業 領 域	H (MAXN, N), SL (MAXN, N)。
D, S, Y, R, W1, W2	実 数 型 1 次 元 配 列	作 業 領 域	D (N), S (N), Y (N), R (N), W1 (N), W2 (N)。
W3, W4	実 数 型 1 次 元 配 列	作 業 領 域	W3 (M), W4 (M)

(3) 計算法

Biggs による準ニュートンの反復法に基づいて、局所的な最小値を求める。収束判定の制御は、引数 XTOL と FTOL の値によって行われ、次の条件のどれかが満たされたとき終了する。

- ① $|f_j(x)| < \max(\text{FTOL}, \epsilon), j = 1, 2, \dots, m$
 ② $|(f(x^+), \partial_j f(x^+))| \leq \alpha_1 \|f(x^+)\|_2 \|\partial_j f(x^+)\|_2 (j = 1, 2, \dots, m)$
 かつ $\|x^+ - x\|_\infty \leq \alpha_2 \max(\|x^+\|_\infty, 1.0)$

ここで、

$$\alpha_1 = \begin{cases} 10^{-3} & (f \text{のみ与えられたとき}) \\ 10^{-4/4} & (f \text{と} \partial f \text{が与えられたとき}) \end{cases}$$

$$\alpha_2 = \max(\text{XTOL}, \beta)$$

$$\beta = \begin{cases} 16 \epsilon^{1/2} & (f \text{のみ与えられたとき}) \\ 32 \epsilon & (f \text{と} \partial f \text{のみ与えられたとき}) \end{cases}$$

$$\partial_i f_j(x) = (\partial f_j / \partial x_i)$$

ϵ は、マシンに依存する定数、 x^+ と x は反復計算における相続く二つの x の値である。

- ③ 反復回数が上限値を越えたとき。
- ④ 関数値の計算回数がその上限値を越えたとき。
- ⑤ x の値の変化が顕著でなくなったとき。

(4) 使用例

```

DIMENSION X(20),F(100),DF(100,20),X0(20),DX(20),F0(100)
DIMENSION DF0(100,20),H(20,20),SL(20,20),D(20),S(20),Y(20)
DIMENSION R(20),W1(20),W2(20),W3(100),W4(100)
MAXM=100
MAXN=20
M=2
N=2
X(1)=-1.2
X(2)=1.0
ITMAX=100
FTOL=1.0E-5
XTOL=1.0E-5
NFEMAX=5000
LDERIV=1
NPRINT=3
WRITE(6,6000) MAXM,MAXN,M,N,ITMAX,NFEMAX,FTOL,XTOL,LDERIV,NPRINT
WRITE(6,6100) (X(J),J=1,N)
CALL NOLLS1(
-MAXM,MAXN,M,N,X,ITMAX,NFEMAX,FTOL,XTOL,LDERIV,NPRINT
-,FF,F,DF,ITER,NFE,NDE,INFORM,X0,DX
-,FO,DF0,H,SL,D,S,Y,R,W1,W2,W3,W4)
WRITE(6,6200) ITER,NFE,NDE
6000 FORMAT(1H0,4X,'INITIAL VALUES',/1H,10X,'MAXM=',I4
-, 'MAXN=',I4,' M=',I2,' N=',I2,' ITMAX=',I4
-, 'NFEMAX=',I5/1H,10X,'FTOL=',1PE16.7,' XTOL='
-,E16.7/1H,10X,'LDERIV=',I2,' NPRINT=',I2)
6100 FORMAT(1H,10X,'X=',1P5E16.7/(1H,10X,5E16.7))
6200 FORMAT(1H0,10X,'ITERATION',I6/1H,10X,'MODELF-CALL',I4
-/1H,10X,'MODEL-D-CALL',I4)
STOP
END
SUBROUTINE MODELF(M,N,X,F)
DIMENSION X(N),F(M)
F(1)=10.0*(X(1)*X(1)-X(2))
F(2)=1.0-X(1)
RETURN
END
SUBROUTINE MODELD(MAXM,M,N,X,DF)
DIMENSION X(N),DF(MAXM,N)
DF(1,1)=20.0*X(1)
DF(1,2)=-10.0
DF(2,1)=-1.0
DF(2,2)=0.0
RETURN
END

```

出力結果

```

INITIAL VALUES
MAXM= 100  MAXN= 20  M= 2  N= 2  ITMAX= 100  NFEMAX= 5000
FTOL= 9.999997E-06  XTOL= 9.999997E-06
LDERIV= 1  NPRINT= 3
X= -1.1999998E+00  1.0000000E+00
0  THE SUM OF SQUARES= 2.4199875E+01
X= -1.1999998E+00  1.0000000E+00
F= 4.3999863E+00  2.1999998E+00
DF= -2.3999985E+01  -1.0000000E+01
-1.0000000E+00  0.0
1  THE SUM OF SQUARES= 2.1258163E+01
X= -1.0189848E+00  6.2381876E-01
F= 4.1451035E+00  2.0189848E+00

```

```

DF= -2.3999985E+01 -1.0000000E+01
-1.0000000E+00 0.0
2 THE SUM OF SQUARES= 3.9795551E+00
X= -9.9474800E-01 9.9184918E-01
F= -2.3256540E-02 1.9947472E+00
DF= -2.0379684E+01 -1.0000000E+00
-1.0000000E+00 0.0
3 THE SUM OF SQUARES=
X= -7.6816183E-01
F= 4.96682E-01
DF= -1.0000000E+00
1.0846248E-05
9.9394315E-01
2.9606819E-03
1.0000000E+01 -1.0000000E+01
1.0000000E+00 0.0
THE SUM OF SQUARES= 1.3669265E-08
X= 9.9991751E-01 9.9982673E-01
F= 8.2850456E-05 8.2492828E-05
DF= 1.9940781E+01 -1.0000000E+01
-1.0000000E+00 0.0
21 THE SUM OF SQUARES= 3.6948222E-13
X= 9.9999988E-01 9.9999970E-01
F= 5.9604645E-07 1.1920929E-07
DF= 1.9998337E+01 -1.0000000E+01
-1.0000000E+00 0.0
***** FINISHED *****
*****
ITERATION 21
MODEL-F-CALL 38
MODEL-D-CALL 21

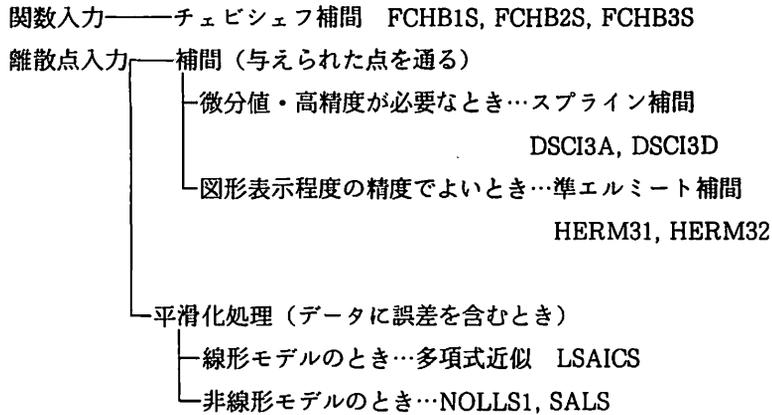
```

参 考 文 献

- 1) Bartholomew-Biggs, M.C.; "The estimation of Hessian matrix in nonlinear least squares problems with non-zero residuals", Mathematical Programming 12, pp.67-80, (1977).
- 2) 田辺國士; "非線型最小二乗法のアルゴリズム", 応用統計学, Vol. 9, No. 3, pp.119-140 (1981).
- 3) Kunio Tanabe and Sumie Ueda; "NOLLS1, A Fortran subroutine for nonlinear least squares by a quasi-Newton method", Computer Science Monographs, The Institute of Statistical Mathematics (1981).

補間、平滑化ルーチンの選び方

名大プログラム・ライブラリーには、データの与えられ方、及びそのデータに誤差が含まれるか否かにより、各々のルーチンが用意されている。先ず、データの与えられ方が、関数の形で与えられ、いかなる点における関数値でも計算できるときを関数入力と呼ぶ。この時はチェビシェフ補間のルーチンが良い。一方、データが離散点で与えられるときで、データに誤差はなく、又は小さいときは、スプライン補間が良い。誤差を含むときは、最小二乗近似ルーチンを用いるとよい。



DSCI1A DSFI1A
 DSCI2A DSFI2A
 DSCI3A DSFI3A
 DSCI4A DSFI4A

Spline Interpolation (One Dimensional)

スプライン補間 (1変数)

作成	桑野 和郎	1978年6月
形式	サブルーチン	言語; FORTRAN サイズ; 298, 141, 263, 141, 280, 150, 389, 176行

(1) 概要

機能は、離散点で関数値が与えられ、一部では両端で更に若干の端条件が与えられたとき、

- (1) 与えられた点を通り、端条件を満足する $2m-1$ ($m \geq 2$) 次の多項式スプラインを構成すること (サブルーチン名の3字目がCのもの)、
- (2) 構成された $2m-1$ 次の多項式スプラインの関数値 (補間値)、 l ($1 \leq l \leq 2m-1$) 次微係数及び左端から与えられた点までの積分を計算すること (サブルーチン名の3字目がFのもの)、

である。

端で与えられる条件に応じて、

- (1) Type-I スプライン補間
- (2) Type-II スプライン補間
- (3) Type-III スプライン補間
- (4) 周期スプライン補間

の4種類がある。

1. Type-I スプライン補間

$n+1$ 個の点

$$x_0 < x_1 < \dots < x_n \quad (n \geq 1)$$

において関数値 $f(x_i)$ 、両端 x_0, x_n において $m-1$ 次までの微係数 $f^{(l)}(x_0), f^{(l)}(x_n)$, ($1 \leq l \leq m-1$) が与えられたとき、 x_0, x_n を $2m$ 重、 x_i , ($1 \leq i \leq n-1$) を 1 重節点とする $2m-1$ 次 ($m \geq 2$) の多項式スプライン

$$S(x) = \sum_{j=-2m+1}^{n-1} c_j N_j(x)$$

により、 $f(x)$ を補間する。ここで、 $N_j(x)$, ($-2m+1 \leq j \leq n-1$) は正規化された B-spline で次のように定義される関数である。

$$g_{2m}(t; x) = (t-x)_+^{2m-1} = \begin{cases} (t-x)^{2m-1} & (t \geq x) \\ 0 & (t < x) \end{cases}$$

$$N_j(x) = (t_{j+2m} - t_j) g_{2m}[t_j, t_{j+1}, \dots, t_{j+2m}; x]$$

$$t_j = \begin{cases} x_0 & (-2m+1 \leq j \leq -1) \\ x_j & (0 \leq j \leq n) \\ x_n & (n+1 \leq j \leq n+2m-1) \end{cases}$$

サブルーチン DSCI1A により補間係数 c_j , $(-2m+1 \leq j \leq n-1)$ を算出し, DSFI1A により $x_0 \leq x \leq x_n$ に対する $S(x)$, $S^{(l)}(x)$, $\int_{x_0}^x S(x) dx$ を算出する。

両端で $m-1$ 次までの微係数を与えることができるならば, これらのルーチンの使用が好ましい。4 種類のうちでは最も高精度が期待できる手法である。

2. Type-II スプライン補間

$n+1$ 個の点

$$x_0 < x_1 < \dots < x_n \quad (n \geq m-1)$$

において関数値 $f(x_i)$, 両端 x_0, x_n において m から $2m-2$ 次までの微係数 $f^{(l)}(x_0), f^{(l)}(x_n)$, $(m \leq l \leq 2m-2)$ が与えられたとき, x_0, x_n を $2m$ 重, x_i , $(1 \leq i \leq n-1)$ を 1 重節点とする $2m-1$ 次 $(m \geq 2)$ の多項式スプライン

$$S(x) = \sum_{j=-2m+1}^{n-1} c_j N_j(x)$$

により, $f(x)$ を補間する。ここで, $N_j(x)$ は TYPE-1 のときと同じである。

サブルーチン DSCI2A により補間係数 c_j , $(-2m+1 \leq j \leq n-1)$ を算出し, DSFI2A により $x_0 \leq x \leq x_n$ に対する $S(x)$, $S^{(l)}(x)$, $\int_{x_0}^x S(x) dx$ を算出する。

4 種類のうちで最も実用性が低い。しかし, 両端で m から $2m-2$ 次までの微係数を 0 とおいて得られる $2m-1$ 次の補間スプラインは「自然スプライン」と呼ばれていて, スプラインの応用の中では最も有名なものである。

本ルーチンを使って自然スプラインを構成することができるが, たいていの場合, 次の TYPF-III スプライン補間に比較して誤差が大きい。

3. Type-III スプライン補間

$n+1$ 個の点

$$x_0 < x_1 < \dots < x_n \quad (n \geq 2m)$$

において関数値 $f(x_i)$ が与えられたとき, x_0, x_n を $2m$ 重, x_i , $(m \leq i \leq n-m)$ を 1 重節点とする $2m-1$ 次 $(m \geq 2)$ の多項式スプライン

$$S(x) = \sum_{j=-2m+1}^{n-2m+1} c_j N_j(x)$$

により, $f(x)$ を補間する。ここで, $N_j(x)$, $(-2m+1 \leq j \leq n-2m+1)$ は正規化された B-spline で次のように定義される関数である。

$$g_{2m}(t; x) = (t-x)_+^{2m-1} = \begin{cases} (t-x)^{2m-1} & (t \geq x) \\ 0 & (t < x) \end{cases}$$

$$N_j(x) = (t_{j+2m} - t_j) g_{2m}[t_j, t_{j+1}, \dots, t_{j+2m}; x]$$

$$t_j = \begin{cases} x_0 & (-2m+1 \leq j \leq 0) \\ x_{j+m-1} & (1 \leq j \leq n-2m+1) \\ x_n & (n-2m+2 \leq j \leq n+1) \end{cases}$$

サブルーチン DSCI3A により補間係数 c_j , $(-2m+1 \leq j \leq n-2m+1)$ を算出し, DSFI3A により $x_0 \leq x \leq x_n$ に対する $S(x)$, $S^{(l)}(x)$, $\int_{x_0}^x S(x) dx$ を算出する。

関数値のみを使って補間し得るので, $f(x)$ が非周期関数の場合は最も実用性が高い。

4. 周期スプライン補間

補間される関数 $f(x)$ が周期 $x_n - x_0$ の周期関数で, $n+1$ 個の点

$$x_0 < x_1 < \dots < x_n \quad (n \geq 2m)$$

において関数値 $f(x_i)$ が与えられているとする。このとき, 次のように定義される関数 $S(x)$ により, $f(x)$ を補間する。

$$g_{2m}(t; x) = (t-x)^{2m-1} = \begin{cases} (t-x)^{2m-1} & (t \geq x) \\ 0 & (t < x) \end{cases}$$

$$N_j(x) = (t_{j+2m} - t_j) g_{2m}[t_j, t_{j+1}, \dots, t_{j+2m}; x]$$

$$t_j = \begin{cases} x_{n+j} - (x_n - x_0) & (-2m+1 \leq j \leq -1) \\ x_j & (0 \leq j \leq n) \\ x_{j-n} + (x_n - x_0) & (n+1 \leq j \leq n+2m-1) \end{cases}$$

$$S(x) = \sum_{j=-2m+1}^{n-1} c_j N_j(x)$$

$$\begin{cases} c_j = c_{j+n} & (-2m+1 \leq j \leq -m) \\ c_j = c_{j-n} & (n-m+1 \leq j \leq n-1) \end{cases}$$

上式で定義される $2m-1$, $(m \geq 2)$ 次の多項式スプライン $S(x)$ は,

$$S^{(l)}(x_0) = S^{(l)}(x_n) \quad (0 \leq l \leq 2m-2)$$

を満足するという意味で周期関数とみなすことができる。

サブルーチン DSCI4A により補間係数 c_j , $(-2m+1 \leq j \leq n-1)$ を算出し, DSFI4A により $x_0 \leq x \leq x_n$ に対する $S(x)$, $S^{(l)}(x)$, $\int_{x_0}^x S(x) dx$ を算出する。

$f(x)$ が周期関数のときにはこれらのルーチンを使用することが好ましい。

(2) 使用法

CALL DSCI1A (XI, F, DER, CJ, N, M, WORKC)

CALL DSFI1A (XP, I, L, FP, N, M, XI, CJ, WORKF)

引数	型と種類	属性	内容
XI	倍精度実数型 1次元配列	入力	離散点 x_i . 大きさ $n+1$ の配列. x_i , $(0 \leq i \leq n)$ を XI(i+1) に入れる。
F	倍精度実数型 1次元配列	入力	離散点 x_i における関数値 $f(x_i)$, $(0 \leq i \leq n)$. 大きさ $n+1$ の配列. $f(x_i)$ を F(i+1) に入れる。
DER	倍精度実数型 2次元配列	入力	端点 x_0, x_n における l 次微係数 $(1 \leq l \leq m-1)$. 大きさ $(2m-1)$ の 2次元配列. $f^{(l)}(x_0)$ DER(1, l) に, $f^{(l)}(x_n)$ を DER(2, l) に入れる。
CJ	倍精度実数型 1次元配列	入出力	DSCI1Aでは出力. DSCF1Aでは入力. 補間係数 c_j , $(-2m+1 \leq j \leq n-1)$. 大きさ $n+2m-1$ の配列. c_j が CJ(j+2m) に入る。

引数	型と種類	属性	内容
N	整数型	入力	離散点の個数。 $n+1$ における n を入れる。
M	整数型	入力	スプラインの次数 $2m-1$ における m を入れる。
WORKC	倍精度実数型 1次元配列	入出力	作業領域。大きさ $(n-1)(2m-1)+2m^2+2m$ の配列。
XP	倍精度実数型	入力	補間値などを評価したい点 x 。 $XI(1) \leq XP \leq XI(N+1)$ を満足すること。この範囲外のXPを与えると、エラー・メッセージを印刷し、 $FP=0.0$ とする。
I	整数型	入力	$XI(I+1) \leq XP < XI(I+2)$ を満足する整数 I を入れる。 I が上の条件を満たしていなくても正常に計算するが、探索のために計算時間が若干増える。
L	整数型	入力	$-1 \leq L \leq 2*M-1$ を満たす整数。計算の種類を与える。 $L=-1$: $XI(1)$ から XP までの不定積分を計算し、 FP に出力する。 $L=0$: XP における補間値を計算し、 FP に出力する。 $1 \leq L \leq 2*M-1$: XP における L 次微係数を計算し、 FP に出力する。 $L < -1, L > 2*M-1$: エラー・メッセージを印刷し、 $FP=0.0$ とする。
FP	倍精度実数型	出力	補間値など計算結果が入る。
WORKF	倍精度実数型 1次元配列	入出力	作業領域。大きさ $2m$ の配列。

CALL DSCI2A (XI, F, DER, CJ, N, M, WORKC)

CALL DSFI2A (XP, I, L, FP, N, M, XI, CJ, WORKF)

引数	型と種類	属性	内容
DER	倍精度実数型 2次元配列	入力	端点 x_0, x_n における l 次微係数($m \leq l \leq 2m-2$)。大きさ $(2, m-1)$ の2次元配列。 $f^{(l)}(x_0)$ を $DER(1, l-m+1)$ に、 $f^{(l)}(x_n)$ を $DER(2, l-m+1)$ に入れる。
WORKC	倍精度実数型 1次元配列	入出力	作業領域。大きさ $(n+2m-3)(2m-1)+4m$ の配列。
他の引数は、TYPE-1 スプラインと同じである。(ただし、CJは、DSCI2Aでは出力、DSFI2Aでは入力である。)			

CALL DSCI3A (XI, F, CJ, X30, N, M, WORKC)

CALL DSFI3A (XP, I, L, FP, N, M, XI, CJ, X30, WORKF)

引数	型と種類	属性	内容
CJ	倍精度実数型 1次元配列	入出力	DSCI3Aでは出力。DSFI3Aでは入力。補間係数 c_j 、 $(-2m+1 \leq j \leq n-2m+1)$ 。大きさ $n+1$ の配列。 c_j がCJ($j+2m$)に入る。
X30	倍精度実数型 1次元配列	入出力	DSCI3Aでは出力。DSFI3Aでは入力。スプラインの節点 $x_0, x_m, x_{m+1}, \dots, x_{n-m}, x_n$ が入る。大きさ $n-2m+3$ の配列。

引 数	型 と 種 類	属 性	内 容
WORKC	倍精度実数型 1次元配列	入出力	作業領域。大きさ $(n-1)(2m-1)+4m$ の配列。
他の引数は、TYPE-I スプラインと同じである。			

CALL DSCI4A (XI, F, CJ, N, M, WORKC)

CALL DSFI4A (XP, I, L, FP, N, M, XI, CJ, WORKF)

引 数	型 と 種 類	属 性	内 容
WORKC	倍精度実数型 1次元配列	入出力	作業領域。大きさ $n(2m-1)+2n(m-1)+2m$ の配列。
他の引数は、TYPE-I スプラインと同じである。(ただし、CJ は、DSCI4A では出力、DSFI4Aでは入力である。)			

(3) 備 考

関数 $f(x)$ の性質によって、4種類のサブルーチンを次のように使い分けるのが望ましい。

1. $f(x)$ が周期関数のときは DSCI4A, DSCI4F を用いる。
2. 両端で $f(x)$ の微係数 $f^{(l)}(x_0), f^{(l)}(x_n), (1 \leq l \leq m-1)$ を与えることができれば、DSCI1A, DSFI1A を用いる。例えば、3次スプライン ($m=2$) 補間のとき、両端における1次微係数を与えればよい。
3. 関数値のみを与えられたときは、DSCI3A, DSFI3A を用いる。
4. いわゆる「自然スプライン」で補間したい場合、DSCI2A, DSFI2A を用いる。

DSCI1D	DSFI1D	DSCI5D	DSFI5D
DSCI2D	DSFI2D	DSCI6D	DSFI6D
DSCI3D	DSFI3D	DSCI7D	DSFI7D
DSCI4D	DSFI4D		

Spline Interpolation (Two Dimensional)

スプライン補間 (2変数)

作 成	秦野 和郎	1978年6月
形 式	サブルーチン 言語 ; FORTRAN サイズ ; 各々約300行	

(1) 概 要

プログラムの機能は、長方形領域の格子点で関数値が与えられ、更に、境界で必要な境界条件が与えられたとき、

- (1) 与えられた点を通り、境界条件を満足する双 $2\nu-1$ ($\nu \geq 2$) 次の多項式スプライン $S(x, y)$ を構成すること (サブルーチン名の3字目がCのもの)
- (2) 構成された双 $2\nu-1$ 次の多項式スプライン $S(x, y)$ の関数値 (補間値)、偏微係数 $\partial^{\lambda+\mu} S(x, y) / \partial^{\lambda} x \partial^{\mu} y$ ($0 \leq \lambda \leq 2\nu-1, 0 \leq \mu \leq 2\nu-1$) 及び不定積分 $\int_{y_0}^y \int_{x_0}^x S(x, y) dx dy$ などを計算すること (サブルーチン名の3字目がFのもの)

である。

境界で与えられる条件に応じて、

- (1) (Type-I) × (Type-I) スプライン補間
- (2) (Type-II) × (Type-II) スプライン補間
- (3) (Type-III) × (Type-III) スプライン補間
- (4) (周期) × (周期) スプライン補間
- (5) (Type-I) × (周期) スプライン補間
- (6) (Type-II) × (周期) スプライン補間
- (7) (Type-III) × (周期) スプライン補間

の7種類がある。

1. (Type-I) × (Type-I) スプライン補間

$$\begin{cases} a = x_0 < x_1 < \dots < x_m = b \\ c = y_0 < y_1 < \dots < y_n = d \end{cases} \quad (1)$$

とする。二変数関数 $f(x, y)$ に関して、

$$\begin{aligned} \textcircled{1} \quad & f_{i,j} = f(x_i, y_j) \quad (0 \leq i \leq m), (0 \leq j \leq n) \\ \textcircled{2} \quad & f_{i,j}^{(\lambda,0)} = f^{(\lambda,0)}(x_i, y_j) \quad (i = 0, m), (0 \leq j \leq n) \\ \textcircled{3} \quad & f_{i,j}^{(0,\mu)} = f^{(0,\mu)}(x_i, y_j) \quad (j = 0, n), (0 \leq i \leq m) \\ \textcircled{4} \quad & f_{i,j}^{(\lambda,\mu)} = f^{(\lambda,\mu)}(x_i, y_j) \quad (i = 0, m), (j = 0, n) \end{aligned} \quad (2)$$

$$(1 \leq \lambda \leq \nu-1), (1 \leq \mu \leq \nu-1)$$

が与えられたとき、すなわち、

- ① すべての格子点上で関数値が与えられ,
- ② $x = x_0 = a$, $x = x_m = b$ 上の格子点上で x 方向の $\nu-1$ 次までの法線微分 $\partial^\lambda f / \partial x^\lambda$ ($1 \leq \lambda \leq \nu-1$) が与えられ,
- ③ $y = y_0 = c$, $y = y_n = d$ 上の格子点上で y 方向の $\nu-1$ 次までの法線微分 $\partial^\mu f / \partial y^\mu$ ($1 \leq \mu \leq \nu-1$) が与えられ,
- ④ 四隅 (x_0, y_0) , (x_m, y_0) , (x_0, y_n) , (x_m, y_n) で偏微係数 $\partial^{\lambda+\mu} / \partial x^\lambda \partial y^\mu$ ($1 \leq \lambda, \mu \leq \nu-1$) が与えられたとき,

双 $2\nu-1$ 次の多項式スプライン

$$S(x, y) = \sum_{\beta=-2\nu+1}^{n-1} \sum_{\alpha=-2\nu+1}^{m-1} c_{\alpha, \beta} N_\alpha(x; \Delta_x) N_\beta(y; \Delta_y) \quad (3)$$

により $f(x, y)$ を補間する。ここで, $c_{\alpha, \beta}$ ($-2\nu+1 \leq \alpha \leq m-1$), ($-2\nu+1 \leq \beta \leq n-1$) は補間係数である。また,

$$N_\alpha(x; \Delta_x) = (s_{\alpha+2\nu} - s_\alpha) g_{2\nu}[s_\alpha, s_{\alpha+1}, \dots, s_{\alpha+2\nu}; x]$$

$$g_{2\nu}(s; x) = (s-x)_{+}^{2\nu-1} = \begin{cases} (s-x)^{2\nu-1} & (s \geq x) \\ 0 & (s < x) \end{cases} \quad (4)$$

$$s_\alpha = \begin{cases} x_0 & (-2\nu+1 \leq \alpha \leq -1) \\ x_\alpha & (0 \leq \alpha \leq m) \\ x_m & (m+1 \leq \alpha \leq m+2\nu-1) \end{cases}$$

であり,

$$N_\beta(y; \Delta_y) = (t_{\beta+2\nu} - t_\beta) g_{2\nu}[t_\beta, t_{\beta+1}, \dots, t_{\beta+2\nu}; y]$$

$$g_{2\nu}(t; y) = (t-y)_{+}^{2\nu-1}$$

$$t_\beta = \begin{cases} y_0 & (-2\nu+1 \leq \beta \leq -1) \\ y_\beta & (0 \leq \beta \leq n) \\ y_n & (n+1 \leq \beta \leq n+2\nu-1) \end{cases} \quad (5)$$

である。式 (3) に補間条件 (2) を適用すると, 補間係数 $c_{\alpha, \beta}$ を未知数とする $(m+2\nu-1) \cdot (n+2\nu-1)$ 元の連立一次方程式が得られる。これを解いて得られた補間係数を式 (3) に代入すれば, 任意の $a \leq x \leq b$, $c \leq y \leq d$ に対する補間値などを計算することができる。

サブルーチン DSCI1D により補間係数 $c_{\alpha, \beta}$ ($-2\nu+1 \leq \alpha \leq m-1$), ($-2\nu+1 \leq \beta \leq n-1$) を計算し, サブルーチン DSFI1D により補間値など $S^{(\lambda, \mu)}(x, y)$ ($-1 \leq \lambda, \mu \leq 2\nu-1$) を計算する。ここで,

$$S^{(-1, -1)}(x, y) = \int_c^y \int_a^x S(x, y) dx dy$$

$$S^{(-1, \mu)}(x, y) = \int_a^x \frac{\partial^\mu S(x, y)}{\partial y^\mu} dx \quad (0 \leq \mu \leq 2\nu-1)$$

$$S^{(\lambda, -1)}(x, y) = \int_c^y \frac{\partial^\lambda S(x, y)}{\partial x^\lambda} dy \quad (0 \leq \lambda \leq 2\nu-1)$$

$$S^{(\lambda, \mu)}(x, y) = \frac{\partial^{\lambda+\mu} S(x, y)}{\partial x^\lambda \partial y^\mu} \quad (0 \leq \lambda, \mu \leq 2\nu-1)$$

である。

2. (Type-II) × (Type-II) スプライン補間

$$\begin{cases} a = x_0 < x_1 < \dots < x_m = b \\ c = y_0 < y_1 < \dots < y_n = d \end{cases} \quad (7)$$

とする。二変数関数 $f(x, y)$ に関して

$$\begin{aligned} \textcircled{1} \quad & f_{i,j} = f(x_i, y_j) \quad (0 \leq i \leq m), (0 \leq j \leq n) \\ \textcircled{2} \quad & f_{i,j}^{(\lambda,0)} = f^{(\lambda,0)}(x_i, y_j) \quad (i=0, m), (0 \leq j \leq n) \\ \textcircled{3} \quad & f_{i,j}^{(0,\mu)} = f^{(0,\mu)}(x_i, y_j) \quad (j=0, n), (0 \leq i \leq m) \\ \textcircled{4} \quad & f_{i,j}^{(\lambda,\mu)} = f^{(\lambda,\mu)}(x_i, y_j) \quad (i=0, m), (j=0, n) \\ & (\nu \leq \lambda \leq 2\nu-2), (\nu \leq \mu \leq 2\nu-2) \end{aligned} \quad (8)$$

が与えられたとき、すなわち、

- ① すべての格子点上で関数値が与えられ、
- ② $x = x_0 = a, x = x_m = b$ 上の格子点上で x 方向の ν 次から $2\nu-2$ 次までの法線微分 $\partial^\lambda f / \partial x^\lambda$ ($\nu \leq \lambda \leq 2\nu-2$) が与えられ、
- ③ $y = y_0 = c, y = y_n = d$ 上の格子点上で y 方向の ν 次から $2\nu-2$ 次までの法線微分 $\partial^\mu f / \partial y^\mu$ ($\nu \leq \mu \leq 2\nu-2$) が与えられ、
- ④ 四隅 $(x_0, y_0), (x_m, y_0), (x_0, y_n), (x_m, y_n)$ で偏微係数 $\partial^{\lambda+\mu} / \partial x^\lambda \partial y^\mu$ ($\nu \leq \lambda, \mu \leq 2\nu-2$) が与えられたとき、

双 $2\nu-1$ 次の多項式スプライン

$$S(x, y) = \sum_{\beta=-2\nu+1}^{n-1} \sum_{\alpha=-2\nu+1}^{m-1} c_{\alpha,\beta} N_\alpha(x; \Delta_x) N_\beta(y; \Delta_y) \quad (9)$$

により $f(x, y)$ を補間する。ここで、 $c_{\alpha,\beta} (-2\nu+1 \leq \alpha \leq m-1), (-2\nu+1 \leq \beta \leq n-1)$ は補間係数である。また、 $N_\alpha(x; \Delta_x), N_\beta(y; \Delta_y)$ は式 (4), (5) で与えられる関数である。式 (9) に補間条件 (8) を適用すると、補間係数 $c_{\alpha,\beta}$ を未知数とする $(m+2\nu-1) \cdot (n+2\nu-1)$ 元の連立一次方程式が得られる。これを解いて得られた補間係数を式 (9) に代入すれば、任意の $a \leq x \leq b, c \leq y \leq d$ に対する補間値などを計算することができる。

サブルーチン DSCI2D により補間係数 $c_{\alpha,\beta} (-2\nu+1 \leq \alpha \leq m-1), (-2\nu+1 \leq \beta \leq n-1)$ を計算し、サブルーチン DSFI2D により補間値など $S^{(\lambda,\mu)}(x, y) (-1 \leq \lambda, \mu \leq 2\nu-1)$ を計算する。ここで、 $S^{(\lambda,\mu)}(x, y)$ は式 (6) で与えられる。

3. (Type-III) × (Type-III) スプライン補間

$$\begin{cases} a = x_0 < x_1 < \dots < x_m = b \\ c = y_0 < y_1 < \dots < y_n = d \end{cases} \quad (11)$$

とする。二変数関数 $f(x, y)$ の格子点上の値 $f_{i,j} = f(x_i, y_j) (0 \leq i \leq m), (0 \leq j \leq n)$ が与えられたとき、双 $2\nu-1$ 次の多項式スプライン

$$S(x, y) = \sum_{\beta=-2\nu+1}^{n-2\nu+1} \sum_{\alpha=-2\nu+1}^{m-2\nu+1} c_{\alpha,\beta} N_\alpha(x; \Delta'_x) N_\beta(y; \Delta'_y) \quad (12)$$

により $f(x, y)$ を補間する。ここで、 $c_{\alpha,\beta} (-2\nu+1 \leq \alpha \leq m-2\nu+1), (-2\nu+1 \leq \beta \leq n-2\nu+1)$ は補間係数である。また、

$$\begin{aligned} N_\alpha(x; \Delta'_x) &= (s_{\alpha+2\nu} - s_\alpha) g_{2\nu} [s_\alpha, s_{\alpha+1}, \dots, s_{\alpha+2\nu}; x] \\ g_{2\nu}(s; x) &= (s-x)_{+}^{2\nu-1} \end{aligned} \quad (13)$$

$$s_\alpha = \begin{cases} x_0 & (-2\nu+1 \leq \alpha \leq 0) \\ x_{\alpha+\nu-1} & (1 \leq \alpha \leq m-2\nu+1) \\ x_m & (m-2\nu+2 \leq \alpha \leq m+1) \end{cases}$$

であり、

$$\begin{aligned} N_\beta(y; \Delta'y) &= (t_{\beta+2\nu} - t_\beta) g_{2\nu}[t_\beta, t_{\beta+1}, \dots, t_{\beta+2\nu}; x] \\ g_{2\nu}(t; y) &= (t-y)_{+}^{2\nu-1} \\ t_\beta &= \begin{cases} y_0 & (-2\nu+1 \leq \beta \leq 0) \\ y_{\beta+\nu-1} & (1 \leq \beta \leq n-2\nu+1) \\ y_n & (n-2\nu+2 \leq \beta \leq n+1) \end{cases} \end{aligned} \quad (14)$$

である。式(12)に補間条件を適用すると、補間係数 $c_{\alpha, \beta}$ を未知数とする $(m+1) \cdot (n+1)$ 元の連立一次方程式

$$\sum_{\beta=-2\nu+1}^{n-2\nu+1} \sum_{\alpha=-2\nu+1}^{m-2\nu+1} c_{\alpha, \beta} N_\alpha(x_i; \Delta'x) N_\beta(y_j; \Delta'y) = f_{i,j} \quad (i=0, 1, \dots, m), (j=0, 1, \dots, n) \quad (15)$$

が得られる。これを解いて得られた補間係数を式(12)に代入すれば、任意の $a \leq x \leq b, c \leq y \leq d$ に対する補間値などを計算することができる。

サブルーチン DSCI3D により補間係数 $c_{\alpha, \beta} (-2\nu+1 \leq \alpha \leq m-2\nu+1), (-2\nu+1 \leq \beta \leq n-2\nu+1)$ を計算し、サブルーチン DSFI3D により補間値など $S^{(\lambda, \mu)}(x, y) (-1 \leq \lambda, \mu \leq 2\nu-1)$ を計算する。ここで、 $S^{(\lambda, \mu)}(x, y)$ は式(6)で与えられる。

4. (周期) × (周期) スプライン補間

$$\begin{cases} a = x_0 < x_1 < \dots < x_m = b \\ c = y_0 < y_1 < \dots < y_n = d \end{cases} \quad (16)$$

とする。二変数関数 $f(x, y)$ が、変数 x に関して周期 $b-a$ の周期関数、変数 y に関して周期 $d-c$ の周期関数であるとする。 $f(x, y)$ の格子点上の値 $f_{i,j} = f(x_i, y_j) (0 \leq i \leq m), (0 \leq j \leq n)$ が与えられたとき、双 $2\nu-1$ 次の多項式スプライン

$$S(x, y) = \sum_{\beta=-2\nu+1}^{n-1} \sum_{\alpha=-2\nu+1}^{m-1} c_{\alpha, \beta} N_\alpha(x; \Delta_x) N_\beta(y; \Delta_y) \quad (17)$$

$$\begin{cases} c_{\alpha, \beta} = c_{\alpha+m, \beta} & (-2\nu+1 \leq \alpha \leq -\nu) \\ c_{\alpha, \beta} = c_{\alpha-m, \beta} & (m-\nu+1 \leq \alpha \leq m-1) \end{cases} \quad (18)$$

$$(-2\nu+1 \leq \beta \leq n-1)$$

$$\begin{cases} c_{\alpha, \beta} = c_{\alpha, \beta+n} & (-2\nu+1 \leq \beta \leq -\nu) \\ c_{\alpha, \beta} = c_{\alpha, \beta-n} & (n-\nu+1 \leq \beta \leq n-1) \end{cases} \quad (19)$$

$$(-2\nu+1 \leq \alpha \leq m-1)$$

$$\begin{aligned} N_\alpha(x; \Delta_x) &= (s_{\alpha+2\nu} - s_\alpha) g_{2\nu}[s_\alpha, s_{\alpha+1}, \dots, s_{\alpha+2\nu}; x] \\ g_{2\nu}(s; x) &= (s-x)_{+}^{2\nu-1} \\ s_\alpha &= \begin{cases} x_{m+\alpha} - (x_m - x_0) & (-2\nu+1 \leq \alpha \leq -1) \\ x_\alpha & (0 \leq \alpha \leq m) \\ x_{\alpha-m} + (x_m - x_0) & (m+1 \leq \alpha \leq m+2\nu-1) \end{cases} \end{aligned} \quad (20)$$

$$\begin{aligned}
N_\beta(y; \Delta y) &= (t_{\beta+2\nu} - t_\beta) g_{2\nu}[t_\beta, t_{\beta+1}, \dots, t_{\beta+2\nu}; y] \\
g_{2\nu}(t; y) &= (t-y)^{2\nu-1} \\
t_\beta &= \begin{cases} y_{n+\beta} - (y_n - y_0) & (-2\nu+1 \leq \beta \leq -1) \\ y_\beta & (0 \leq \beta \leq n) \\ y_{\beta-n} + (y_n - y_0) & (n+1 \leq \beta \leq n+2\nu-1) \end{cases} \quad (21)
\end{aligned}$$

により $f(x, y)$ を補間する。ここで, $c_{\alpha, \beta} (-2\nu+1 \leq \alpha \leq m-1)$, $(-2\nu+1 \leq \beta \leq n-1)$ は補間係数である。式 (17) ~ (21) で与えられる $S(x, y)$ は,

$$\begin{cases} S^{(\lambda, \mu)}(x_0, y) = S^{(\lambda, \mu)}(x_m, y) \\ S^{(\lambda, \mu)}(x; y_0) = S^{(\lambda, \mu)}(x, y_n) \end{cases} \quad (22)$$

$(0 \leq \lambda, \mu \leq 2\nu-1), (x_0 \leq x \leq x_m), (y_0 \leq y \leq y_n)$

を満足するという意味で周期関数とみなすことができる。式 (17) に補間条件を適用すると, 補間係数 $c_{\alpha, \beta} (-\nu+1 \leq \alpha \leq m-\nu)$, $(-\nu+1 \leq \beta \leq n-\nu)$ を未知数とする $m \cdot n$ 元の連立一次方程式が得られる。これを解いて得られた補間係数を式 (17) に代入すれば, 任意の $a \leq x \leq b$, $c \leq y \leq d$ に対する補間値などを計算することができる。

サブルーチン DSCI4D により, 補間係数 $c_{\alpha, \beta} (-2\nu+1 \leq \alpha \leq m-1)$, $(-2\nu+1 \leq \beta \leq n-1)$ を計算し, サブルーチン DSFI4D により, 補間値など $S^{(\lambda, \mu)}(x, y) (-1 \leq \lambda, \mu \leq 2\nu-1)$ を計算する。ここで, $S^{(\lambda, \mu)}(x, y)$ は式 (6) で与えられる。

5. (Type-I) × (周期) スプライン補間

$$\begin{cases} a = x_0 < x_1 < \dots < x_m = b \\ c = y_0 < y_1 < \dots < y_n = d \end{cases} \quad (23)$$

とする。二変数関数 $f(x, y)$ が, 変数 y に関して周期 $d-c$ の周期関数であるとする。 $f(x, y)$ に関して,

$$\begin{aligned}
\textcircled{1} \quad f_{i,j} &= f(x_i, y_j) \quad (0 \leq i \leq m), (0 \leq j \leq n) \\
\textcircled{2} \quad f_{i,j}^{(\lambda, 0)} &= f^{(\lambda, 0)}(x_i, y_j) \quad (1 \leq \lambda \leq \nu-1), (i=0, m), (0 \leq j \leq n) \end{aligned} \quad (24)$$

が与えられたとき, すなわち,

① すべての格子点上で関数値が与えられ,

② $x = x_0 = a$, $x = x_m = b$ 上の格子点上で x 方向の $\nu-1$ 次までの法線微分 $\partial^\lambda f / \partial x^\lambda$ ($1 \leq \lambda \leq \nu-1$) が与えられたとき,

双 $2\nu-1$ 次の多項式スプライン

$$S(x, y) = \sum_{\beta=-2\nu+1}^{n-1} \sum_{\alpha=-2\nu+1}^{m-1} c_{\alpha, \beta} N_\alpha(x; \Delta x) N_\beta(y; \Delta y) \quad (25)$$

$$\begin{aligned}
c_{\alpha, \beta} &= c_{\alpha, \beta+n} \quad (-2\nu+1 \leq \beta \leq -\nu) \\
c_{\alpha, \beta} &= c_{\alpha, \beta-n} \quad (n-\nu+1 \leq \beta \leq n-1) \\
& \quad (-2\nu+1 \leq \alpha \leq m-1) \end{aligned} \quad (26)$$

により $f(x, y)$ を補間する。ここで, $c_{\alpha, \beta} (-2\nu+1 \leq \alpha \leq m-1)$, $(-2\nu+1 \leq \beta \leq n-1)$ は補間係数である。また, $N_\alpha(x; \Delta x)$ は式 (4) で, $N_\beta(y; \Delta y)$ は式 (21) で与えられる。式 (25) に補間条件 (24) を適用すると, 補間係数 $c_{\alpha, \beta} (-2\nu+1 \leq \alpha \leq m-1)$, $(-\nu+1 \leq \beta \leq n-\nu)$ を未知数とする $(m+2\nu-1) \cdot n$ 元の連立一次方程式が得られる。これを解いて得られた補間係数を式 (25) に

代入すれば、任意の $a \leq x \leq b$, $c \leq y \leq d$ に対する補間値などを計算することができる。

サブルーチン DSCI5D により補間係数 $c_{\alpha,\beta}$ ($-2\nu+1 \leq \alpha \leq m-1$), ($-2\nu+1 \leq \beta \leq n-1$) を計算し、サブルーチン DSFI5D により補間値など $S^{(\lambda,\mu)}(x,y)$ ($-1 \leq \lambda, \mu \leq 2\nu-1$) を計算する。ここで、 $S^{(\lambda,\mu)}(x,y)$ は式 (6) で与えられる。

6. (Type-II) × (周期) スプライン補間

$$\begin{cases} a = x_0 < x_1 < \dots < x_m = b \\ c = y_0 < y_1 < \dots < y_n = d \end{cases} \quad (27)$$

とする。二変数関数 $f(x,y)$ が、変数 y に関して周期 $d-c$ の周期関数であるとする。 $f(x,y)$ に関して、

$$\begin{aligned} \textcircled{1} \quad f_{i,j} &= f(x_i, y_j) && (0 \leq i \leq m), (0 \leq j \leq n) \\ \textcircled{2} \quad f_{i,j}^{(\lambda,0)} &= f^{(\lambda,0)}(x_i, y_j) && (\nu \leq \lambda \leq 2\nu-2), (i=0, m), (0 \leq j \leq n) \end{aligned} \quad (28)$$

が与えられたとき、すなわち、

① すべての格子点上で関数値が与えられ、

② $x = x_0 = a, x = x_m = b$ 上の格子点上で x 方向の ν 次から $2\nu-2$ 次までの法線微分 $\partial^\lambda f / \partial x^\lambda$ ($\nu \leq \lambda \leq 2\nu-2$) が与えられたとき、

双 $2\nu-1$ 次の多項式スプライン

$$S(x,y) = \sum_{\beta=-2\nu+1}^{n-1} \sum_{\alpha=-2\nu+1}^{m-1} c_{\alpha,\beta} N_\alpha(x; \Delta_x) N_\beta(y; \Delta_y) \quad (29)$$

$$\begin{cases} c_{\alpha,\beta} = c_{\alpha,\beta+n} & (-2\nu+1 \leq \beta \leq -\nu) \\ c_{\alpha,\beta} = c_{\alpha,\beta-n} & (n-\nu+1 \leq \beta \leq n-1) \end{cases} \quad (30)$$

$$(-2\nu+1 \leq \alpha \leq m-1)$$

により $f(x,y)$ を補間する。ここで、 $c_{\alpha,\beta}$ ($-2\nu+1 \leq \alpha \leq m-1$), ($-2\nu+1 \leq \beta \leq n-1$) は補間係数である。また、 $N_\alpha(x; \Delta_x)$ は式 (4) で、 $N_\beta(y; \Delta_y)$ は式 (21) で与えられる。式 (29) に補間条件 (28) を適用すると、補間係数 $c_{\alpha,\beta}$ ($-2\nu+1 \leq \alpha \leq m-1$), ($-\nu+1 \leq \beta \leq n-\nu$) を未知数とする $(m+2\nu-1) \cdot n$ 元の連立一次方程式が得られる。これを解いて得られた補間係数を式 (29) に代入すれば、任意の $a \leq x \leq b$, $c \leq y \leq d$ に対する補間値などを計算することができる。

サブルーチン DSCI6D により補間係数 $c_{\alpha,\beta}$ ($-2\nu+1 \leq \alpha \leq m-1$), ($-2\nu+1 \leq \beta \leq n-1$) を計算し、サブルーチン DSFI6D により補間値など $S^{(\lambda,\mu)}(x,y)$ ($-1 \leq \lambda, \mu \leq 2\nu-1$) を計算する。ここで、 $S^{(\lambda,\mu)}(x,y)$ は式 (6) で与えられる。

7. (Type-III) × (周期) スプライン補間

$$\begin{cases} a = x_0 < x_1 < \dots < x_m = b \\ c = y_0 < y_1 < \dots < y_n = d \end{cases} \quad (31)$$

とする。二変数関数 $f(x,y)$ が、変数 y に関して周期 $d-c$ の周期関数であるとする。 $f(x,y)$ の格子点上の値 $f_{i,j} = f(x_i, y_j)$ ($0 \leq i \leq m$), ($0 \leq j \leq n$) が与えられたとき、

双 $2\nu-1$ 次の多項式スプライン

$$S(x,y) = \sum_{\beta=-2\nu+1}^{n-1} \sum_{\alpha=-2\nu+1}^{m-2\nu+1} c_{\alpha,\beta} N_\alpha(x; \Delta'_x) N_\beta(y; \Delta_y) \quad (32)$$

$$\begin{cases} c_{\alpha,\beta} = c_{\alpha,\beta+n} & (-2\nu+1 \leq \beta \leq -\nu) \\ c_{\alpha,\beta} = c_{\alpha,\beta-n} & (n-\nu+1 \leq \beta \leq n-1) \end{cases} \quad (33)$$

$$(-2\nu+1 \leq \alpha \leq m-2\nu+1)$$

により $f(x, y)$ を補間する。ここで, $c_{\alpha,\beta} (-2\nu+1 \leq \alpha \leq m-2\nu+1), (-2\nu+1 \leq \beta \leq n-1)$ は補間係数である。また, $N_\alpha(x; \Delta'_x)$ は式(13)で, $N_\beta(y; \Delta_y)$ は式(21)で与えられる。式(32)に補間条件を適用すると, 補間係数 $c_{\alpha,\beta} (-2\nu+1 \leq \alpha \leq m-2\nu+1), (-\nu+1 \leq \beta \leq n-\nu)$ を未知数とする $(m+1) \cdot n$ 元の連立一次方程式が得られる。これを解いて得られた補間係数を式(32)に代入すれば, 任意の $a \leq x \leq b, c \leq y \leq d$ に対する補間値などを計算することができる。

サブルーチン DSCI7D により補間係数 $c_{\alpha,\beta} (-2\nu+1 \leq \alpha \leq m-2\nu+1), (-2\nu+1 \leq \beta \leq n-1)$ を計算し, サブルーチン DSFI7D により補間値など $S^{(\lambda,\mu)}(x, y) (-1 \leq \lambda, \mu \leq 2\nu-1)$ を計算する。ここで, $S^{(\lambda,\mu)}(x, y)$ は式(6)で与えられる。

(2) 使用法

CALL DSCI1D (XI, YJ, F, CAB, NX, NY, M, WORKC, NXM2D)

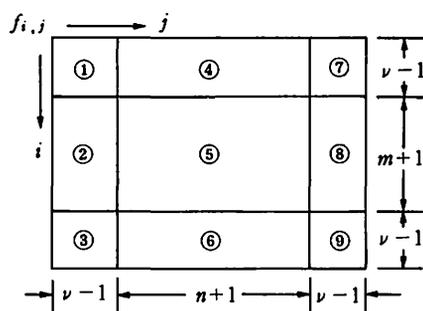
CALL DSFI1D (XP, YP, IX, IY, LX, LY, FP, NX, NY, M, XI, YJ, CAB, WORKF, NXM2D)

引 数	型 と 種 類	属 性	内 容
XI	倍精度実数型 1次元配列	入 力	x 方向の格子点 x_i 。大きさ $m+1$ の配列。 $x_i (0 \leq i \leq m)$ を $XI(i+1)$ に入れる。
YJ	倍精度実数型 1次元配列	入 力	y 方向の格子点 y_j 。大きさ $n+1$ の配列。 $y_j (0 \leq j \leq n)$ を $YJ(j+1)$ に入れる。
F	倍精度実数型 2次元配列	入 力	格子点における関数値など $\bar{f}_{i,j}$ 。大きさ $(m+2\nu-1) \times (n+2\nu-1)$ の2次元配列。 $\bar{f}_{i,j} (1 \leq i \leq m+2\nu-1), (1 \leq j \leq n+2\nu-1)$ を $F(i, j)$ に入れる。
CAB	倍精度実数型 2次元配列	入 出 力	DSCI1Dでは出力。DSFI1Dでは入力。補間係数 $c_{\alpha,\beta} (-2\nu+1 \leq \alpha \leq m-1), (-2\nu+1 \leq \beta \leq n-1)$ 。大きさ $(m+2\nu-1) \times (n+2\nu-1)$ の2次元配列。 $c_{\alpha,\beta}$ が $CAB(\alpha+2\nu, \beta+2\nu)$ に入る。
NX	整 数 型	入 力	x 方向の格子の個数 m を入れる。
NY	整 数 型	入 力	y 方向の格子の個数 n を入れる。
M	整 数 型	入 力	スプラインの次数 $2\nu-1$ における ν を入れる。
WORKC	倍精度実数型 1次元配列	入 出 力	作業領域。大きさは, $k = \max(m, n)$ として $(k-1)(2\nu-1) + 2\nu^2 + 6\nu + 2k - 2$ 。
NXM2D	整 数 型	入 力	整合配列の寸法。F, CABの第1添字の大きさ。NXM2D $\geq m+2\nu-1$ を満たすこと。
XP, YP	倍精度実数型	入 力	補間値などを評価したい点 (x, y) 。 x を XP, y を YP に入れる。 $XI(1) \leq XP \leq XI(NX+1), YJ(1) \leq YP \leq YJ(NY+1)$ を満足すること。この範囲外の XP, YP を与えると, エラー・メッセージを印刷して $FP = 0.0$ とする。
IX, IY	整 数 型	入 力	$XI(IX+1) \leq XP \leq XI(IX+2), YJ(IY+1) \leq YP \leq YJ(IY+2)$ を満足する整数 IX, IY を入れる。IX, IY が上の条件

引数	型と種類	属性	内容
			を満たしていても正常に計算するが、探索のために計算時間が若干増える。
LX, LY	整数型	入力	$-1 \leq LX, LY \leq 2\nu - 1$ を満たす整数。計算の種類を与える。すなわち、評価したい量 $S^{(\lambda, \mu)}(x, y)$ における λ を LX に、 μ を LY に入れる。
FP	倍精度実数型	出力	補間値など $S^{(\lambda, \mu)}(x, y)$ の計算結果が出力される。
WORKF	倍精度実数型 1次元配列	入出力	作業領域。大きさは $m+6\nu-1$ 。

ただし、上の使用法の説明を簡単にするために、被補間関数 $f(x, y)$ に関して、次のように $\bar{f}_{i,j}$ ($1 \leq i \leq m+2\nu-1$), ($1 \leq j \leq n+2\nu-1$) を定義する。

- ① $\bar{f}_{i,j} = f^{(\nu-i, \nu-j)}(x_0, y_0)$ ($1 \leq i, j \leq \nu-1$)
- ② $\bar{f}_{i,j} = f^{(0, \nu-j)}(x_{i-\nu}, y_0)$ ($\nu \leq i \leq m+\nu$), ($1 \leq j \leq \nu-1$)
- ③ $\bar{f}_{i,j} = f^{(i-m-\nu, \nu-j)}(x_m, y_0)$ ($m+\nu+1 \leq i \leq m+2\nu-1$), ($1 \leq j \leq \nu-1$)
- ④ $\bar{f}_{i,j} = f^{(\nu-i, 0)}(x_0, y_{j-\nu})$ ($1 \leq i \leq \nu-1$), ($\nu \leq j \leq n+\nu$)
- ⑤ $\bar{f}_{i,j} = f(x_{i-\nu}, y_{j-\nu})$ ($\nu \leq i \leq m+\nu$), ($\nu \leq j \leq n+\nu$)
- ⑥ $\bar{f}_{i,j} = f^{(i-m-\nu, 0)}(x_m, y_{j-\nu})$ ($m+\nu+1 \leq i \leq m+2\nu-1$), ($\nu \leq j \leq n+\nu$)
- ⑦ $\bar{f}_{i,j} = f^{(\nu-i, j-n-\nu)}(x_0, y_n)$ ($1 \leq i \leq \nu-1$), ($n+\nu+1 \leq j \leq n+2\nu-1$)
- ⑧ $\bar{f}_{i,j} = f^{(0, j-n-\nu)}(x_{i-\nu}, y_n)$ ($\nu \leq i \leq m+\nu$), ($n+\nu+1 \leq j \leq n+2\nu-1$)
- ⑨ $\bar{f}_{i,j} = f^{(i-m-\nu, j-n-\nu)}(x_m, y_n)$ ($m+\nu+1 \leq i \leq m+2\nu-1$), ($n+\nu+1 \leq j \leq n+2\nu-1$)



例えば、 $\nu = 2$, $m = 2$, $n = 3$ のときは、 $\bar{f}_{i,j}$ の並びは次のようになる。

		$j = 1 \sim 6$					
$i = 1 \sim 5$	{	$f_{00}^{(1,1)}$	$f_{00}^{(1,0)}$	$f_{01}^{(1,0)}$	$f_{02}^{(1,0)}$	$f_{03}^{(1,0)}$	$f_{03}^{(1,1)}$
	$f_{00}^{(0,1)}$	f_{00}	f_{01}	f_{02}	f_{03}	$f_{03}^{(0,1)}$	
	$f_{10}^{(0,1)}$	f_{10}	f_{11}	f_{12}	f_{13}	$f_{13}^{(0,1)}$	
	$f_{20}^{(0,1)}$	f_{20}	f_{21}	f_{22}	f_{23}	$f_{23}^{(0,1)}$	
	$f_{20}^{(1,1)}$	$f_{20}^{(1,0)}$	$f_{21}^{(1,0)}$	$f_{22}^{(1,0)}$	$f_{23}^{(1,0)}$	$f_{23}^{(1,1)}$	

CALL DSCI2D (XI, YJ, F, CAB, NX, NY, M, WORKC, NXM2D)

CALL DSFI2D (XP, YP, IX, IY, LX, LY, FP, NX, NY, M, XI, YJ, CAB, WORKF, NXM2D)

引 数	型 と 種 類	属 性	内 容
P	倍精度実数型 2次元配列	入 力	格子点における関数値など $f_{i,j}$ (式(10)で与えられる量)。大きさ $(m+2\nu-1) \times (n+2\nu-1)$ の2次元配列。 $\bar{f}_{i,j} (1 \leq i \leq m+2\nu-1, (1 \leq j \leq n+2\nu-1))$ を $F(i, j)$ に入れる。
WORKC	倍精度実数型 1次元配列	入 出 力	作業領域。大きさは、 $k = \max(m, n)$ として $(k+2\nu-3)(2\nu-1) + 2\nu^2 + 6\nu + 2k - 2$ 。
他の引数は、(Type-I) \times (Type-I) スプラインと同じである。(ただし、CABはDSCI2Dでは出力、DSFI2Dでは入力である。)			

ただし、上の使用法の説明を簡単にするために、被補間関数 $f(x, y)$ に関して、次のように $\bar{f}_{i,j} (1 \leq i \leq m+2\nu-1, (1 \leq j \leq n+2\nu-1))$ を定義する。

- ① $\bar{f}_{i,j} = f^{(2\nu-1-i, 2\nu-1-j)}(x_0, y_0) \quad (1 \leq i, j \leq \nu-1)$
- ② $\bar{f}_{i,j} = f^{(0, 2\nu-1-j)}(x_{i-\nu}, y_0) \quad (\nu \leq i \leq m+\nu, (1 \leq j \leq \nu-1))$
- ③ $\bar{f}_{i,j} = f^{(i-m-1, 2\nu-1-j)}(x_m, y_0) \quad (m+\nu+1 \leq i \leq m+2\nu-1, (1 \leq j \leq \nu-1))$
- ④ $\bar{f}_{i,j} = f^{(2\nu-1-i, 0)}(x_0, y_{j-\nu}) \quad (1 \leq i \leq \nu-1, (\nu \leq j \leq n+\nu))$
- ⑤ $\bar{f}_{i,j} = f(x_{i-\nu}, y_{j-\nu}) \quad (\nu \leq i \leq m+\nu, (\nu \leq j \leq n+\nu))$
- ⑥ $\bar{f}_{i,j} = f^{(i-m-1, 0)}(x_m, y_{j-\nu}) \quad (m+\nu \leq i \leq m+2\nu-1, (\nu \leq j \leq n+\nu))$
- ⑦ $\bar{f}_{i,j} = f^{(2\nu-1-i, j-n-1)}(x_0, y_n) \quad (1 \leq i \leq \nu-1, (n+\nu \leq j \leq n+2\nu-1))$
- ⑧ $\bar{f}_{i,j} = f^{(0, j-n-1)}(x_{i-\nu}, y_n) \quad (\nu \leq i \leq m+\nu, (n+\nu \leq j \leq n+2\nu-1))$
- ⑨ $\bar{f}_{i,j} = f^{(i-m-1, j-n-1)}(x_m, y_n) \quad (m+\nu \leq i \leq m+2\nu-1, (n+\nu \leq j \leq n+2\nu-1))$

例えば、 $\nu = 2, m = 2, n = 3$ のときは、 $\bar{f}_{i,j}$ の並びは次のようになる。

$j = 1 \sim 6$

i 1 5	$f_{00}^{(2,2)}$	$f_{00}^{(2,0)}$	$f_{01}^{(2,0)}$	$f_{02}^{(2,0)}$	$f_{03}^{(2,0)}$	$f_{03}^{(2,2)}$
	$f_{00}^{(0,2)}$	f_{00}	f_{01}	f_{02}	f_{03}	$f_{03}^{(0,2)}$
	$f_{10}^{(0,2)}$	f_{10}	f_{11}	f_{12}	f_{13}	$f_{13}^{(0,2)}$
	$f_{20}^{(0,2)}$	f_{20}	f_{21}	f_{22}	f_{23}	$f_{23}^{(0,2)}$
	$f_{20}^{(2,2)}$	$f_{20}^{(2,0)}$	$f_{21}^{(2,0)}$	$f_{22}^{(2,0)}$	$f_{23}^{(2,0)}$	$f_{23}^{(2,2)}$

CALL DSCI3D (XI, YJ, F, CAB, XY30, NX, NY, M, WORKC, NXP1D)

CALL DSFI3D (XP, YP, IX, IY, LX, LY, FP, NX, NY, M, XI, YJ, CAB, XY30, WORKF, NXP1D)

引 数	型 と 種 類	属 性	内 容
F	倍精度実数型 2次元配列	入 力	格子点における関数値 $f_{i,j}$ 。大きさ $(m+1) \times (n+1)$ の2次元配列。 $f_{i,j} (0 \leq i \leq m, (0 \leq j \leq n))$ を $F(i+1, j+1)$ に入れる。

引 数	型 と 種 類	属 性	内 容
CAB	倍精度実数型 2次元配列	入出力	DSCI3Dでは出力。DSFI3Dでは入力。補間係数 $c_{\alpha,\beta}$ ($-2v+1 \leq \alpha \leq m-2v+1$), ($-2v+1 \leq \beta \leq n-2v+1$)。大きさ $(m+1) \times (n+1)$ の2次元配列。 $c_{\alpha,\beta}$ が CAB ($\alpha+2v, \beta+2v$) に入る。
XY3O	倍精度実数型 1次元配列	入出力	DSCI3Dでは出力。DSFI3Dでは入力。スプラインの節点列 $x_0, x_v, x_{v+1}, \dots, x_{m-v}, x_m; y_0, y_v, y_{v+1}, \dots, y_n$ が入る。大きさ $m+n-4v+6$ の配列。
WORKC	倍精度実数型 1次元配列	入出力	作業領域。大きさは、 $k = \max(m, n)$ として $(k-1)(2v-1) + 4v+2(k+1)$ 。
NXP1D	整数型	入力	整合配列の寸法。F, CABの第1添字の大きさ。NXP1D $\geq m+1$ を満たすこと。
他の引数は (Type-I) \times (Type-I) スプラインと同じである。			

CALL DSCI4D (XI, YJ, F, CAB, NX, NY, M, WORKC, NXP1D, NXM2D)

CALL DSFI4D (XP, YP, IX, IY, LX, LY, FP, NX, NY, M, XI, YJ, CAB, WORKF, NXM2D)

引 数	型 と 種 類	属 性	内 容
F	倍精度実数型 2次元配列	入力	格子点における関数値 $f_{i,j}$ 。大きさ $(m+1) \times (n+1)$ の2次元配列。 $f_{i,j}$ ($0 \leq i \leq m$), ($0 \leq j \leq n$) を F ($i+1, j+1$) に入れる。
CAB	倍精度実数型 2次元配列	入出力	DSCI4Dでは出力。DSFI4Dでは入力。補間係数 $c_{\alpha,\beta}$ ($-2v+1 \leq \alpha \leq m-1$), ($-2v+1 \leq \beta \leq n-1$)。大きさ $(m+2v-1) \times (n+2v-1)$ の2次元配列。 $c_{\alpha,\beta}$ が CAB ($\alpha+2v, \beta+2v$) に入る。
WORKC	倍精度実数型 1次元配列	入出力	作業領域。大きさは、 $k = \max(m, n)$ として $k(4v-1) + 4v$ 。
NXP1D	整数型	入力	整合配列の寸法。配列Fの第1添字の大きさ。NXP1D $\geq m+1$ を満たすこと。
NXM2D	整数型	入力	整合配列の寸法。配列CABの第1添字の大きさ。NXM2D $\geq m+2v-1$ を満たすこと。
他の引数は (Type-I) \times (Type-I) スプラインと同じである。			

CALL DSCI5D (XI, YJ, F, CAB, NX, NY, M, WORKC, NXM2D)

CALL DSFI5D (XP, YP, IX, IY, LX, LY, FP, NX, NY, M, XI, YJ, CAB, WORKF, NXM2D)

引 数	型 と 種 類	属 性	内 容
F	倍精度実数型 2次元配列	入力	格子点における関数値など $f_{i,j}$ ($1 \leq i \leq m+2v-1$), ($1 \leq j \leq n+1$)。大きさ $(m+2v-1) \times (n+1)$ の2次元配列。 $\bar{f}_{i,j}$ を F (i, j) に入れる。

引 数	型 と 種 類	属 性	内 容
CAB	倍精度実数型 2次元配列	入出力	DSCI5Dでは出力。DSFI5Dでは入力。補間係数 $c_{\alpha,\beta}$ ($-2v+1 \leq \alpha \leq m-1$), ($-2v+1 \leq \beta \leq n-1$)。大きさ $(m+2v-1) \times (n+2v-1)$ の2次元配列。 $c_{\alpha,\beta}$ がCAB($\alpha+2v, \beta+2v$)に入る。
WORKC	倍精度実数型 1次元配列	入出力	作業領域。大きさは、次式で与えられる k_1, k_2 のうちの大きい方である。 $k_1 = (m-1)(2v-1) + 2v^2 + 6v + 2m - 2$ $k_2 = n(4v-1) + 4v$
他の引数は (Type-I) × (Type-I) スプラインと同じである。			

ただし、上の使用法の説明を簡単にするために、被補間関数 $f(x, y)$ に関して、次のように $\bar{f}_{i,j}$ ($1 \leq i \leq m+2v-1$), ($1 \leq j \leq n+1$)を定義する。

- ① $\bar{f}_{i,j} = f^{(\nu-i,0)}(x_0, y_{j-1})$ ($1 \leq i \leq \nu-1$)
- ② $\bar{f}_{i,j} = f(x_{i-\nu}, y_{j-1})$ ($\nu \leq i \leq m+\nu$)
- ③ $\bar{f}_{i,j} = f^{(i-m-\nu,0)}(x_m, y_{j-1})$ ($m+\nu+1 \leq i \leq m+2v-1$)
($1 \leq j \leq n+1$)

例えば、 $\nu = 2, m = 2, n = 3$ のときは、 $f_{i,j}$ の並びは次のようになる。

$$j = 1 \sim 4$$

$i = 1 \sim 5$	$f_{00}^{(1,0)}$	$f_{01}^{(1,0)}$	$f_{02}^{(1,0)}$	$f_{03}^{(1,0)}$
	f_{00}	f_{01}	f_{02}	f_{03}
	f_{10}	f_{11}	f_{12}	f_{13}
	f_{20}	f_{21}	f_{22}	f_{23}
	$f_{20}^{(1,0)}$	$f_{21}^{(1,0)}$	$f_{22}^{(1,0)}$	$f_{23}^{(1,0)}$

CALL DSCI6D (XI, YJ, F, CAB, NX, NY, M, WORKC, NXM2D)

CALL DSFI6D (XP, YP, IX, IY, LX, LY, FP, NX, NY, M, XI, YJ, CAB, WORKF,
NXM2D)

引 数	型 と 種 類	属 性	内 容
F	倍精度実数型 2次元配列	入 力	格子点における関数値など $\bar{f}_{i,j}$ ($1 \leq i \leq m+\nu-1$), ($1 \leq j \leq n+1$)。大きさ $(m+2v-1) \times (n+1)$ の2次元配列。 $\bar{f}_{i,j}$ をF(i, j)に入れる。
CAB	倍精度実数型 2次元配列	入出力	DSCI6Dでは出力。DSFI6Dでは入力。補間係数 $c_{\alpha,\beta}$ ($-2v+1 \leq \alpha \leq m-1$) ($-2v+1 \leq \beta \leq n-1$)。大きさ $(m+2v-1) \times (n+2v-1)$ の2次元配列。 $c_{\alpha,\beta}$ がCAB($\alpha+2v, \beta+2v$)に入る。
WORKC	倍精度実数型 1次元配列	入出力	作業領域。大きさは、次式で与えられる k_1, k_2 のうちの大きい方である。 $k_1 = (m+2v-3)(2v-1) + 2v^2 + 6v + 2m - 2$ $k_2 = n(4v-1) + 4v$
他の引数は (Type-I) × (Type-I) スプラインと同じである。			

ただし、上の使用法の説明を簡単にするために、被補間関数 $f(x, y)$ に関して、次のように $\bar{f}_{i,j}$ ($1 \leq i \leq m+2\nu-1$), ($1 \leq j \leq n+1$) を定義する。

- ① $\bar{f}_{i,j} = f^{(2\nu-1-i,0)}(x_0, y_{j-1})$ ($1 \leq i \leq \nu-1$)
 - ② $\bar{f}_{i,j} = f(x_{i-\nu}, y_{j-1})$ ($\nu \leq i \leq m+\nu$)
 - ③ $\bar{f}_{i,j} = f^{(i-m-1,0)}(x_m, y_{j-1})$ ($m+\nu+1 \leq i \leq m+2\nu-1$)
- ($1 \leq j \leq n+1$)

例えば、 $\nu = 2, m = 2, n = 3$ のときは、 $\bar{f}_{i,j}$ の並びは次のようになる。

		$j = 1 \sim 4$			
5 1 1 1 1	$f_{00}^{(2,0)}$	$f_{01}^{(2,0)}$	$f_{02}^{(2,0)}$	$f_{03}^{(2,0)}$	
	f_{10}	f_{11}	f_{12}	f_{13}	
	f_{20}	f_{21}	f_{22}	f_{23}	
	$f_{20}^{(2,0)}$	$f_{21}^{(2,0)}$	$f_{22}^{(2,0)}$	$f_{23}^{(2,0)}$	

CALL DSCI7D (XI, YJ, F, CAB, X30, NX, NY, M, WORKC, NXP1D)
 CALL DSFI7D (XP, YP, IX, IY, LX, LY, FP, NX, NY, M, XI, YJ, CAB, X30,
 WORKF, NXP1D)

引 数	型 と 種 類	属 性	内 容
F	倍精度実数型 2次元配列	入 力	格子点における関数値 $f_{i,j}$ ($0 \leq i \leq m$), ($0 \leq j \leq n$)。大きさ $(m+1) \times (n+1)$ の2次元配列。 $f_{i,j}$ を F(i+1, j+1) に入れる。
CAB	倍精度実数型 2次元配列	入 出 力	DSCI7Dでは出力。DSFI7Dでは入力。補間係数 $c_{\alpha,\beta}$ ($-2\nu+1 \leq \alpha \leq m-2\nu+1$), ($-2\nu+1 \leq \beta \leq n-1$)。大きさ $(m+1) \times (n+2\nu-1)$ の2次元配列。 $c_{\alpha,\beta}$ が CAB($\alpha+2\nu, \beta+2\nu$) に入る。
X30	倍精度実数型 1次元配列	入 出 力	DSCI7Dでは出力。DSFI7Dでは入力。スプラインの節点列 $x_0, x_\nu, x_{\nu+1}, \dots, x_{m-\nu}, x_m$ が入る。大きさ $m-2\nu+3$ の配列。
WORKC	倍精度実数型 1次元配列	入 出 力	作業領域。大きさは、次式で与えられる k_1, k_2 のうち大きい方である。 $k_1 = (m-1)(2\nu-1) + 4\nu + 2(m+1)$ $k_2 = n(4\nu-1) + 4\nu$
NXP1D	整 数 型	入 力	整合配列の寸法。F, CAB の第1添字の大きさ。NXP1D $\geq m+1$ を満たすこと。
他の引数は (Type-I) \times (Type-I) スプラインと同じである。			

(3) 備 考

1. 境界で $\nu-1$ 次までの偏微係数を与えることができれば、DSCI1D, DSFI1D を使用するとよい。7種類のうちで最も高精度を期待できる。
2. 格子点上における関数値のみを使って補間し得る点では、DSCI3D, DSFI3D が最も有効である。

3. x 方向, y 方向ともに周期性を持つ関数の補間には, DSCI4D, DSFI4D が有効である。。
4. DSCI2D, DSFI2D は境界で与えるべき $f(x, y)$ の偏微係数を全部 0 とおくと, 一変数の自然スプラインを二変数へ拡張した補間式が得られる。
5. DSCI5D, DSFI5D は円柱座標系で定義された二変数関数 $z = f(r, \theta)$ ($0 \leq a \leq r \leq b$), ($0 \leq \theta \leq 2\pi$) の関数値が格子点上で与えられ, $r = a, r = b$ で r 方向の $\nu-1$ 次までの偏微係数が与えられたとき, 適用し得る。
6. DSCI6D, DSFI6D は円柱座標系で定義された二変数関数 $z = f(r, \theta)$ ($0 \leq a \leq r \leq b$), ($0 \leq \theta \leq 2\pi$) の関数値が格子点上で与えられ, $r = a, r = b$ で r 方向の ν 次から $2\nu-2$ 次までの偏微係数が与えられたとき, 適用し得る。
7. DSCI7D, DSFI7D は円柱座標系で定義された二変数関数 $z = f(r, \theta)$ ($0 \leq a \leq r \leq b$), ($0 \leq \theta \leq 2\pi$) の関数値が格子点上で与えられたときに適用し得る。

SPLINE
SPLINT
SPLN22
INTGRL

Calculation Based on the Spline Fit Curve (Derivative, Interpolation and Integration)
 スプライン・フィット曲線による近似計算 (導関数, 内挿, 積分の値)

作成	Theodore Katsanis 1964年 改訂; 峯村 吉泰 1975年12月
形式	サブルーチン 言語; FORTRAN サイズ; 38, 66, 35行

(1) 概要

離散点 x_i 上での関数値 y_i を用いて, スプライン・フィット法に基づき, 微係数, 導関数の値, 補間値及び積分値の計算を行う。

SPLINE } 離散点 x_i 上での1次, 2次の微係数を求める。
 SPLN 22 }

SPLINT 任意の点 $x(k)$ における補間値及び1次導関数の値を求める。

INTGRL 定積分 $\int_{x_1}^{x_2} y dx$ を求める。

(2) 使用法

CALL SPLINE (X, Y, N, SLOPE, EM, L)
 CALL SPLINT (X, Y, N, Z, MAX, YINT, DYDX, L)
 CALL SPLN 22 (X, Y, Y1P, YNP, N, SLOPE, EM, L)
 CALL INTGRL (X, Y, N, SUM)

引数	型と種類	属性	内容
X	実数型 1次元配列	入力	N個の要素を持つ配列名。離散点入力データ x_i の値 ($i = 1, N$)。
Y	実数型 1次元配列	入力	N個の要素を持つ配列名。離散点入力データ x_i 上での関数値 y_i ($i = 1, N$)。
N	整数型	入力	入力データ x_i の総数。保存される。 $2 \leq N \leq 100$
SLOPE	実数型 1次元配列	出力	N個の要素を持つ配列名。離散点入力データ X(I) 上での1次微係数 ($I = 1, N$)。
EM	実数型 1次元配列	出力	N個の要素を持つ配列名。離散点入力データ X(I) 上での2次微係数 ($I = 1, N$)。
L	整数型	入力	ライン・プリンター出力の制御をする。 $L = 1$ のとき, 各入出力変数の値を印刷する。 $L \neq 1$ のとき, 何も印刷しない。
Z	実数型 1次元配列	入力	MAX個の要素を持つ配列名。補間値を求める点の x 座標。保存される。
MAX	整数型	入力	補間値を求める点の数。

引 数	型 と 種 類	属 性	内 容
YINT	実 数 型 1 次 元 配 列	出 力	MAX 個の要素を持つ配列名。Z (K) 上での補間値が YINT (K) に出力される (K = 1, MAX)。
DYDX	実 数 型 1 次 元 配 列	出 力	MAX 個の要素を持つ配列名。Z (K) 上での 1 次導関数の値が DYDX (K) に出力される (K = 1, MAX)。
Y1P	実 数 型	入 力	X (1) 上での微係数。
YNP	実 数 型	入 力	X (N) 上での微係数。
SUM	実 数 型 1 次 元 配 列	出 力	N 個の要素を持つ配列名。始点から、各入力データ X (I) までの面積分の値。 SUM (1) = 0 SUM (2) = $\int_{X(1)}^{X(2)} y dx, \dots, \int_{X(1)}^{X(N)} y dx$

(3) 備 考

本ルーチンは参考文献(1)及び(2)にあるサブルーチンを転用したものである。

参 考 文 献

- 1) Theodore Katsanis; "Use of Arbitrary Quasi-Orthogonals for Calculating Flow Distribution in the Meridional Plane of a Turbomachine", NASA TN-2549 (1964).
- 2) Theodore Katsanis; "Use of Arbitrary Quasi-Orthogonals for Calculating Flow Distribution on a Blade-to-Blade Surface in a Turbomachine", NASA TN-2809 (1965).
- 3) J.L. Walsh, J.H. Ahlberg and E.N. Nilson; "Best Approximation Properties of the Spline Fit", *J. Math. and Mech.*, Vol.11, No.2, p.225 (1962).

HERM31

HERM51

Curve Fitting by the Piecewise Hermite Interpolation Formula (3, 5-Degrees)

区分的エルミート補間による曲線のあてはめ

作 成	秦野 甯世 1976年6月
形 式	サブルーチン 言語 ; FORTRAN サイズ ; 151, 175行

(1) 概 要

離散点 x_i 上で与えられた関数値 $f_i = f(x_i) (i = 1, n)$ を用いて、任意の点 x における関数値 $y = \bar{f}(x)$ 及び微係数を求める。補間法は3次 (HERM31) 又は5次 (HERM51) の区分的エルミート補間による。

(2) 使用法

CALL HERM31 (I, X, Y, M, N, XI, YI, YD, ND, ILL)

CALL HERM51 (I, X, Y, M, N, XI, YI, YD, ND, ILL)

引 数	型 と 種 類	属 性	内 容
I	整 数 型	入 力	$x_i \leq x \leq x_{i+1}$ となる格子点の番号 i の値。 $1 \leq i < N$
X	実 数 型	入 力	補間値を求める点の x 座標。 $XI(1) \leq XI(I) \leq X \leq XI(I+1) \leq XI(N)$
Y	実 数 型 1 次 元 配 列	出 力	(M+1)個の要素を持つ1次元配列名。 x における関数の補間値 y とその点での微係数。 $M=0$ のときは実数型変数名でも可。
M	整 数 型	入 力	求めたい微係数の最大次数 (0のときは関数値のみ)。 HERM31の場合 $0 \leq M \leq 2$, HERM51の場合 $0 \leq M \leq 3$ 。
N	整 数 型	入 力	入力データ x_i の総数。 $2 \leq N$ 。
XI	実 数 型 1 次 元 配 列	入 力	N個の要素を持つ1次元配列名。離散点 x_i の値。 $XI(1) < XI(2) < \dots < XI(N)$
YI	実 数 型 1 次 元 配 列	入 力	N個の要素を持つ1次元配列名。 x_i 上の関数値 $f_i (i = 1, \dots, N)$ 。
YD	実 数 型 1 次 元 配 列 (HERM31の場合) 又は 実 数 型 2 次 元 配 列 (HERM51の場合)	入 出 力	(1) HERM31の場合: N個の要素を持つ1次元配列名。入力としての意味は、離散点 $XI(1), XI(I+1)$ における1次微係数を $YD(1), YD(I+1)$ にそれぞれ入力する。このとき $ILL=0$ を指定しなければならない。離散点上での微係数が未知の場合には、出力としての意味を持ち、 $ILL \neq 0$ を指定することによってすべての点 (N個) における1次微係数の近似値が出力される。 (2) HERM51の場合: $ND \times 2$ 個の要素を持つ2次元配列名。入力としての意味は、離散点上での1次と2次の微係

引 数	型 と 種 類	属 性	内 容
			数を指定する。XI(I)における1次微係数を YD(I, 1)に、 2次微係数を YD(I, 2)に、XI(I+1)におけるそれらを YD(I+1, 1), YD(I+1, 2)に、それぞれ入力する。この とき ILL = 0を指定しなければならない。 微係数が未知の場合には ILL ≠ 0を指定することによって 出力としての意味を持ち、すべての点 (N個) における 1 次、2次の微係数の近似値が出力される。
ND	整 数 型	入 力	YDの配列宣言における第1添字の値。N ≤ ND
ILL	整 数 型	入 出 力	入力としての意味は、各点における微係数 (HERM31の場合 は1次, HERM51の場合は1次, 2次) が既知のとき, ILL = 0として YDにその値を入力する。未知のときは、本ルーチ ンの中で区分的ラグランジュ補間を用いて近似値を求めて Y Dに出力するので、用いるべき補間式の次数を指定する。ILL = 1のとき1次式を、ILL = 2のとき2次式を、ILL ≥ 3の とき4次式を用いる。出力としての意味は、ILL = 0のとき正 常終了、ILL = 30000のときは引数に関する制限が破られた ために全く計算が行われなかったことを示す。

(3) 備 考

離散点上での微係数が未知の場合でも、2点以上の補間値を繰返し求める場合は、最初の1回のみ ILL ≠ 0を指定すると以後は自動的に ILL = 0が設定されているので、計算時間は ILL = 0の場合と同じですむ。

参 考 文 献

- 1) A. Ralston; "A First Course in Numerical Analysis", McGraw-Hill, p.42, 60 (1965).

HERM32

HERM52

Surface Fitting by the Piecewise Hermite Interpolation Formula (3, 5-Degrees)

区分的エルミート補間による曲面のあてはめ

作成	秦野 甯世	1977年4月
形式	サブルーチン 言語 ; FORTRAN サイズ ; 190, 242行	

(1) 概要

長方形格子点 (x_i, y_i) 上で与えられた関数値 $f_{ij} = f(x_i, y_i)$, $(i = 1, \dots, n_x, j = 1, \dots, n_y)$ を用いて, 任意点 (x, y) における関数値 $z = \bar{f}(x, y)$ 及び微係数を求める。補間法は3次(HERM32)及び5次(HERM52)の区分的エルミート補間による。

(2) 使用法

CALL HERM32 (IX, X, JY, Y, Z, M, XI, NX, YI, NY, F, N1, N2, ILL)

CALL HERM52 (IX, X, JY, Y, Z, M, XI, NX, YI, NY, F, N1, N2, ILL)

引数	型と種類	属性	内容
IX	整数型	入力	補間値を求める点の x 座標について $x_i \leq x \leq x_{i+1}$ なる格子点の番号 i の値。 $1 \leq IX < NX$
X	実数型	入力	補間値を求める点の x 座標。 $XI(1) \leq XI(1) \leq X \leq XI(1+1) \leq XI(N)$
JY	整数型	入力	補間値を求める点の y 座標について $y_j \leq y \leq y_{j+1}$ なる格子点の番号 j の値。 $1 \leq JY < NY$
Y	実数型	入力	補間値を求める点の y 座標。 $YI(1) \leq YI(JY) \leq Y \leq YI(JY+1) \leq YI(NY)$
Z	実数型 1次元配列	出力	$(M+1)^2$ 個の要素を持つ1次元配列名。点 (X, Y) における補間値及びその点での微係数。例えば, $M=1$ のときは $Z(1) = \bar{f}, \quad Z(2) = \left[\frac{\partial \bar{f}}{\partial x} \right]_{x=X, y=Y},$ $Z(3) = \left[\frac{\partial \bar{f}}{\partial y} \right]_{x=X, y=Y}, \quad Z(4) = \left[\frac{\partial^2 \bar{f}}{\partial x \partial y} \right]_{x=X, y=Y}$ の順に出力される。 $M=0$ のときは実数型変数名でも可。
M	整数型	入力	求めたい微係数の最大次数 (0 のときは関数値のみ)。 HERM32のとき $0 \leq M \leq 2$ 。 HERM52のとき $0 \leq M \leq 3$ 。
XI	実数型 1次元配列	入力	NX 個の要素を持つ1次元配列名。格子点の x 座標。 $XI(1) < XI(2) < \dots < XI(NX)$
NX	整数型	入力	x 方向の格子点の総数。 $2 \leq NX$

引 数	型 と 種 類	属 性	内 容
YI	実 数 型 1 次 元 配 列	入 力	NY 個の要素を持つ 1 次元配列名。格子点の y 座標。 $YI(1) < YI(2) < \dots < YI(NY)$
NY	整 数 型	入 力	y 方向の格子点の総数。 $2 \leq NY$
F	実 数 型 3 次 元 配 列	入 出 力	<p>(1) HERM32 の場合： $N1 \times N2 \times 4$ 個の要素を持つ 3 次元配列名。入力としての意味は、格子点上の関数値 f_{ij} を指定する。 $F(I, J, 1)$ には格子点 $(X(I), Y(J))$ における関数値 $f_{i,j}$ を指定する ($I = 1, \dots, NX, J = 1, \dots, NY$)。 $ILL = 0$ を指定した時には、更に、 $F(I, J, K)$ ($K = 2, 3, 4$) に、それぞれ、</p> $\left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial^2 f}{\partial x \partial y} \right]_{x=XI(I), y=YI(J)}$ <p>を順に入力する。保存される。これらの微係数が未知の場合は、 $ILL \neq 0$ を指定することによって、 $F(I, J, K)$ ($I = 1, \dots, NX, J = 1, \dots, NY, K = 2, 3, 4$) は出力変数としての意味を持ち、微係数の近似値が出力される。</p> <p>(2) HERM52 の場合： $N1 \times N2 \times 9$ 個の要素を持つ 3 次元配列名。入力としての意味は、格子点上での関数値 f_{ij} を指定する。 $ILL = 0$ を指定したときには、更に、 $F(I, J, K)$ ($K = 2, \dots, 9$) に、それぞれ、</p> $\left[\frac{\partial f}{\partial x}, \frac{\partial^2 f}{\partial x^2}, \frac{\partial f}{\partial y}, \frac{\partial^2 f}{\partial x \partial y}, \frac{\partial^3 f}{\partial x^2 \partial y}, \frac{\partial^2 f}{\partial y^2}, \frac{\partial^3 f}{\partial x \partial y^2}, \frac{\partial^4 f}{\partial x^2 \partial y^2} \right]_{x=XI(I), y=YI(J)}$ <p>の順に入力する。保存される。これらの微係数が未知の場合は、 $ILL \neq 0$ を指定することによって、 $F(I, J, K)$ ($I = 1, \dots, NX, J = 1, \dots, NY, K = 2, \dots, 9$) は出力変数としての意味を持ち、近似値が出力される。</p>
N1, N2	整 数 型	入 力	F の配列宣言における第 1, 第 2 添字。 $NX \leq N1, NY \leq N2$
ILL	整 数 型	入 出 力	<p>入力としての意味は、 $ILL = 0$ のとき、対応する点における微係数を F に入力しなければならない。これらが未知のときは、本ルーチンの中で区分的ラグランジュ補間を用いて近似値を求めて F に出力するので、用いるべき補間式の次数を指定する。 $ILL = 1$ のとき 1 次式を、 $ILL = 2$ のとき 2 次式を、 $ILL \geq 3$ のとき 4 次式を用いる。</p> <p>$ILL \leq NX, ILL \leq NY$</p> <p>出力としての意味は、 $ILL = 0$ のとき正常終了、 $ILL = 30000$ のときは引数に関する制限が破られたために全く計算が行われなかったことを示す。</p>

(3) 備 考

格子点上での微係数が未知の場合でも、2 点以上の補間値を繰返し求める場合は、最初の 1 回のみ $ILL = 0$ を指定すると以後は自動的に $ILL = 0$ が設定されているので、計算時間は $ILL = 0$ の場合と同じです。二変数関数の等高線・立体図表示などの前処理として本ルーチンを使用することにより、なめらかな美しい図を得ることができる。

CFS1A

SFC1A

Curve Fitting by Splines

スプラインによる曲線のあてはめ

作 成	秦野 和郎 1982年1月
形 式	サブルーチン 言語 ; FORTRAN サイズ ; 593行

(1) 概 要

N 個の離散点 $\bar{x}_r : 1 \leq r \leq N$ において観測値 \bar{f}_r , 観測誤差 $\bar{\sigma}_r$ が与えられたとき

$$a = \bar{x}_1 = x_0 < x_1 < \dots < x_n = \bar{x}_N = b \quad (1)$$

を節点とする order k ($k-1$ 次) の多項式スプラインにより $\bar{f}_r : 1 \leq r \leq N$ を最小二乗近似する。

すなわち,

$$N_j(x) = (t_{j+k} - t_j) g_k(t_j, t_{j+1}, \dots, t_{j+k}; x)$$

$$g_k(t; x) = (t-x)_+^{k-1} = \begin{cases} (t-x)^{k-1} : t \geq x \\ 0 : t < x \end{cases} \quad (2)$$

$$t_j = \begin{cases} x_0 : -k+1 \leq j \leq -1 \\ x_j : 0 \leq j \leq n \\ x_n : n+1 \leq j \leq n+k-1 \end{cases}$$

により定義される。正規化された B-spline $N_j(x)$ の線型結合

$$S(x) = \sum_{j=-k+1}^{n-1} c_j N_j(x) \quad (3)$$

の係数 $c_j : -k+1 \leq j \leq n-1$ を, 残差二乗和

$$J = \sum_{r=1}^N \frac{1}{\bar{\sigma}_r^2} \left\{ \bar{f}_r - \sum_{j=-k+1}^{n-1} c_j N_j(\bar{x}_r) \right\}^2 \quad (4)$$

が最小になるようにきめる。(CFS1A)

また, 与えられた x に対して式(3)を計算する。(SFC1A)

(2) 使用法

CALL CFS1A(XR, FR, SIGMAR, XI, CJ, DRESP, STATI, IHIST, PERCT, WORKC,
IWORKC, N, K, KOSU, IWR, ICON)

引 数	型 と 種 類	属 性	内 容
XR	実 数 型 1 次 元 配 列	入 力	離散点 $\bar{x}_r : 1 \leq r \leq N$. 大きさ N .
FR	実 数 型 1 次 元 配 列	入 力	観測値 $\bar{f}_r : 1 \leq r \leq N$. 大きさ N .

引 数	型 と 種 類	属 性	内 容
SIGMAR	実 数 型 1 次 元 配 列	入 力	測定誤差 $\bar{\sigma}_r : 1 \leq r \leq N$ 。大きさ N 又は 1 。 r によって $\bar{\sigma}_r$ が異なるなら $IWR = 1$ とする。 $\bar{\sigma}_r$ が r によらず一定なら、それを SIGMAR(1) に入れて、 $IWR = 0$ とする。このとき配列 SIGMAR の大きさは 1 であり (配列宣言がなくてもよい)。
XI	実 数 型 1 次 元 配 列	入 力	節点 $x_i : 0 \leq i \leq n$ 。大きさ $n+1$ 。 x_i を XI($i+1$) に入れる。
CJ	実 数 型 1 次 元 配 列	出 力	B-spline に掛かる係数 $c_j : -k+1 \leq j \leq n-1$ 。大きさ $n+k-1$ 。 c_j が CJ($j+k$) に入る。
DRESP	実 数 型 1 次 元 配 列	出 力	係数 c_j による残差二乗和の減少量。 $d_j = c_j \sum_{r=1}^N \frac{1}{\bar{\sigma}_r^2} \bar{f}_r N_j(\bar{x}_r)$ 。大きさ $n+k-1$ 。 d_j が DRESP($j+k$) に入る。
STATI	実 数 型 1 次 元 配 列	出 力	大きさ 3 の配列。 STATI(1) : 残差二乗和 J (式(4)) が入る。一般に $J = \sum_{j=-k+1}^{n-1} d_j$ なる関係がある。 STATI(2) : $\sigma = J / (N - (n+k-1))$ が入る。この値がほぼ 1 なら結果は妥当と考えられる。 STATI(3) : $AIC = Ne_w J + 2(n+k-1)$ なる量が入る。
IHIST	整 数 型 2 次 元 配 列	出 力	残差のヒストグラムが入る。大きさ IHIST(2, 25)。 $-0.2i \geq \left\{ \bar{f}_r - \sum_{j=-k+1}^{n-1} c_j N_j(\bar{x}_r) \right\} / \bar{\sigma}_r > -0.2(i+1)$ をみたす r の個数が IHIST(1, $i+1$) に、 $0.2i < \left\{ \bar{f}_r - \sum_{j=-k+1}^{n-1} c_j N_j(\bar{x}_r) \right\} / \bar{\sigma}_r \leq 0.2(i+1)$ をみたす r の個数が IHIST(2, $i+1$) に入る。
PERCT	実 数 型 1 次 元 配 列	出 力	残差の累積度数分布が入る。大きさ PERCT(10)。 $i-1 \leq \left \bar{f}_r - \sum_{j=-k+1}^{n-1} c_j N_j(\bar{x}_r) \right / \bar{\sigma}_r < i$ をみたす r の個数を $K(i)$ とすると、 $PERCT(i) = \sum_{j=1}^i K(j) / N$ $i = 1, 2, \dots, 10$ である。
WORKC	実 数 型 1 次 元 配 列	作 業 領 域	大きさ, WORKC $((n+k)(k+1)-1)$ 。
IWORKC	整 数 型 1 次 元 配 列	作 業 領 域	大きさ, IWORKC $(n+k)$ 。
N	整 数 型	入 力	n 。節点数 -1 。
K	整 数 型	入 力	k 。スプラインの order。
KOSU	整 数 型	入 力	N 。データの個数 N 。

引 数	型 と 種 類	属 性	内 容
IWR	整 数 型	入 力	0 又は 1。測定誤差 $\bar{\sigma}_r$ がデータによらず一定かどうかを指示する。 r によって $\bar{\sigma}_r$ が異なったら IWR = 1。 r によらず $\bar{\sigma}_r$ が一定なら IWR = 0 とする。
ICON	整 数 型	出 力	ICON = 0: 正常終了。 ICON < 0: 異常終了。

CALL SFC1A (XP, I, L, FP, N, K, XI, CJ, WORKF, ICON)

引 数	型 と 種 類	属 性	内 容
XP	実 数 型	入 力	$S(x)$ を計算したい点 x 。 $x_0 \leq x \leq x_n$ でなければならない。
I	整 数 型	入 力	$x_i \leq n < x_{i+1}$ をみたく i 。 $XI(I+1) \leq XP < XI(I+2)$ 。
L	整 数 型	入 力	このサブルーチンは、 $S(x)$ の l 次導関数も計算することができる。評価すべき $S^{(l)}(x)$ における l 。 $0 \leq l \leq k-1$ であること。
FP	実 数 型	出 力	$S^{(l)}(x)$ の計算値。
N	整 数 型	入 力	CFS1A に同じ。
K	整 数 型	入 力	CFS1A に同じ。
XI	実 数 型 1 次 元 配 列	入 力	CFS1A に同じ。
CJ	実 数 型 1 次 元 配 列	入 力	B-spline に掛かる係数。 $c_j: -k+1 \leq j \leq n-1$ 。大きさ $n+k-1$ 。 CFS1A の出力。
WORKF	実 数 型 1 次 元 配 列	作 業 領 域	大きさ $-k$ 。
ICON	整 数 型	出 力	ICON = 0: 正常終了。 ICON < 0: 異常終了。

CFS2A

SFS1A

Surface Fitting by Splines

スプラインによる曲面のあてはめ

作 成	桑野 和郎 1982年1月
形 式	サブルーチン 言語 ; FORTRAN

(1) 概 要

M, N 個の格子点 $(\bar{x}_r, \bar{y}_s) : 1 \leq r \leq M, 1 \leq s \leq N$ において観測値 $\bar{f}_{r,s}$, 観測誤差 $\bar{\lambda}_r \cdot \bar{\mu}_s$ が与えられたとき

$$a = \bar{x}_1 = x_0 < x_1 < \dots < x_M = \bar{x}_M = b \quad (1)$$

を x 方向の節点

$$c = \bar{y}_1 = y_0 < y_1 < \dots < y_N = \bar{y}_N = d \quad (2)$$

を y 方向の節点とする order k (双 $k-1$ 次) の多項式スプラインにより $\bar{f}_{r,s} : 1 \leq r \leq M, 1 \leq s \leq N$ を最小二乗近似する。

すなわち, 正規化された B-spline の双線型結合

$$S(x, y) = \sum_{\alpha=-k+1}^{m-1} \sum_{\beta=-k+1}^{n-1} c_{\alpha, \beta} N_{\alpha}(x) N_{\beta}(y) \quad (3)$$

の係数 $c_{\alpha, \beta} : -k+1 \leq \alpha \leq m-1, -k+1 \leq \beta \leq n-1$ を残差二乗和

$$J = \sum_{r=1}^M \sum_{s=1}^N \frac{1}{\bar{\lambda}_r^2 \cdot \bar{\mu}_s^2} \left\{ \bar{f}_{r,s} - \sum_{\alpha=-k+1}^{m-1} \sum_{\beta=-k+1}^{n-1} c_{\alpha, \beta} N_{\alpha}(\bar{x}_r) N_{\beta}(\bar{y}_s) \right\}^2 \quad (4)$$

が最小になるようにきめる。(CFS2A)

また, 与えられた x, y に対して式(3)を計算する。(SFS1A)

(2) 使用法

CALL CFS2A (XR, YS, FRS, SIGMXR, SIGMYS, XI, YJ, CAB, DRESP, STATI, IHIST,
PERCT, WORKC, IWORKC, KOSUX, KOSUY, NX, NY, K, IWR, KOSXD,
NXK1D, ICON)

引 数	型 と 種 類	属 性	内 容
XR	実 数 型 1 次 元 配 列	入 力	格子点の x 方向の座標 $\bar{x}_r : 1 \leq r \leq M$ 。大きさ M の 1 次元配列。
YS	実 数 型 1 次 元 配 列	入 力	格子点の y 方向の座標 $\bar{y}_s : 1 \leq s \leq N$ 。大きさ N の 1 次元配列。
FRS	実 数 型 2 次 元 配 列	入 力	観測値 $\bar{f}_{r,s} : 1 \leq r \leq M, 1 \leq s \leq N$ 。大きさ, $NX * NY$ の 2 次元配列。
SIGMXR	実 数 型 1 次 元 配 列	入 力	$\bar{f}_{r,s}$ の測定誤差は二つの数の積 $\bar{\lambda}_r, \bar{\mu}_s$ であたえる。 SIGMXR に $\bar{\lambda}_r : 1 \leq r \leq M$ を入れる。大きさ M 又は 1。 r, s によって $\bar{\lambda}_r, \bar{\mu}_s$ が異なるなら IWR = 1 とする。

引 数	型 と 種 類	属 性	内 容
			$\bar{\lambda}_r, \bar{\mu}_s$ が r, s によらず一定なら、それらをそれぞれ SIGM-XR(1), SIGMYS(1)に入れて IWR=0 とする。このとき配列 SIGMXR, SIGMYS の大きさは 1 でよい (配列宣言がなくてもよい)。
SIGMYS	実 数 型 1 次 元 配 列	入 力	$\bar{f}_{r,s}$ の測定誤差, $\bar{\lambda}_r, \bar{\mu}_s$ における $\bar{\mu}_s : 1 \leq s \leq N$ を入れる。大きさ N 又は 1。
XI	実 数 型 1 次 元 配 列	入 力	x 方向の節点 $x_i : 0 \leq i \leq m$ 。大きさ $m+1$ 。 x_i を XI($i+r$) に入れる。
YJ	実 数 型 1 次 元 配 列	入 力	y 方向の節点 $y_j : 0 \leq j \leq n$ 。大きさ $n+1$ 。 y_j を YJ($j+1$) に入れる。
CAB	実 数 型 2 次 元 配 列	出 力	B-spline に掛かる係数 $c_{\alpha,\beta} : -k+1 \leq \alpha \leq m-1, -k+1 \leq \beta \leq n-1$ 。大きさ $(m+k-1)(n+k-1)$ 。 $c_{\alpha,\beta}$ が CAB($\alpha+k, \beta+k$) に入る。
DRESP	実 数 型 2 次 元 配 列	出 力	係数 $c_{\alpha,\beta}$ による残差二乗和の減少量 $d_{\alpha,\beta} = c_{\alpha,\beta} \sum_{r=1}^M \sum_{s=1}^N \frac{1}{\bar{\lambda}_r^2 \bar{\mu}_s^2} \bar{f}_{r,s} \times N_{\alpha}(\bar{x}_r) N_{\beta}(\bar{y}_s)$ 大きさ $(m+k-1) \cdot (n+k-1)$ $d_{\alpha,\beta}$ が DRESP($\alpha+k, \beta+k$) に入る。
STATI	実 数 型 1 次 元 配 列	出 力	大きさ 3 の配列 STATI(1): 残差二乗和, J (式(4)) が入る。一般に $J = \sum_{\alpha=-k+1}^{m-1} \sum_{\beta=-k+1}^{n-1} d_{\alpha,\beta}$ なる関係がある。 STATI(2): $\sigma = J / (MN - (m+k-1)(n+k-1))$ なる量が入る。 この値がほぼ 1 なら結果は妥当と考えられる。 STATI(3): $AIC = M \cdot N \ln J + 2(m+k-1) \times (n+k-1)$ なる量が入る。
IHIST	整 数 型 2 次 元 配 列	出 力	残差のヒストグラムが入る。大きさ IHIST(2, 25)。 $-0.2 \leq \left\{ \bar{f}_{r,s} - \sum_{\alpha=-k+1}^{m-1} \sum_{\beta=-k+1}^{n-1} c_{\alpha,\beta} N_{\alpha}(\bar{x}_r) N_{\beta}(\bar{y}_s) \right\} / \bar{\lambda}_r \bar{\mu}_s > -0.2 (s+1)$ をみたす (r, s) の個数が IHIST($l, i+1$) に, $0.2 \leq \left\{ \bar{f}_{r,s} - \sum_{\alpha=-k+1}^{m-1} \sum_{\beta=-k+1}^{n-1} c_{\alpha,\beta} N_{\alpha}(\bar{x}_r) N_{\beta}(\bar{y}_s) \right\} / \bar{\lambda}_r \bar{\mu}_s \leq 0.2 (i+1)$ をみたす (r, s) の個数が IHIST(2, $i+1$) に入る。
PERCT	実 数 型 1 次 元 配 列	出 力	残差の累積度数分布が入る。大きさ PERCT(10)。 $i-1 \leq \left \bar{f}_{r,s} - \sum_{\alpha=-k+1}^{m-1} \sum_{\beta=-k+1}^{n-1} c_{\alpha,\beta} N_{\alpha}(\bar{x}_r) N_{\beta}(\bar{y}_s) \right / \bar{\lambda}_r \bar{\mu}_s < i$ をみたす (r, s) の個数を $K(i)$ とすると, $\text{PERCT}(i) = \sum_{j=1}^i K(j) / (MN) : i = 1, 2, \dots, 10$ である。

引 数	型 と 種 類	属 性	内 容
WORKC	実 数 型 1 次 元 配 列	作 業 領 域	大きさ, $k \times (m+n+\min(M, N))$ 。
IWORKC	整 数 型 1 次 元 配 列	作 業 領 域	大きさ $M+N+m+n+2$ 。
KOSUX	整 数 型	入 力	データ個数 $M \cdot N$ における M (x 方向の分割数)。
KOSUY	整 数 型	入 力	データ個数 $M \cdot N$ における N (y 方向の分割数)。
NX	整 数 型	入 力	m 。 x 方向の節点数 -1 。
NY	整 数 型	入 力	n 。 y 方向の節点数 -1 。
K	整 数 型	入 力	k 。 スプラインの order。
IWR	整 数 型	入 力	0 又は 1。 測定誤差 $\bar{\lambda}_r \cdot \bar{\mu}_s$ がデータによらず一定かどうかを指示する。 r, s によって $\bar{\lambda}_r \cdot \bar{\mu}_s$ が異なるなら $IWR = 1$ 。 r, s によらず $\bar{\lambda}_r \cdot \bar{\mu}_s$ が一定なら $IWR = 0$ とする。
KOSXD	整 数 型	入 力	整合配列 FRS の第 1 添字。 $KOSXD \geq KOSUX$ であること。
NXK1D	整 数 型	入 力	整合配列 CAB, DRESP の第 1 添字。 $NXK1D \geq NX+K-1$ であること。
ICON	整 数 型	出 力	$ICON = 0$: 正常終了。 $ICON < 0$: 異常終了。

CALL SFS1A(XP, YP, IX, IY, LX, LY, FP, NX, NY, K, XI, YJ, CAB, WORKF, NXK1D,
ICON)

引 数	型 と 種 類	属 性	内 容
XP, YP	実 数 型	入 力	$S(x, y)$ を計算したい点 (x, y) 。 $x_0 \leq x \leq x_m, y_0 \leq y \leq y_n$ であること。 x を XP, y を YP に入れる。
IX, IY	整 数 型	入 力	$x_0 \leq x < x_{i+1}$ をみたく i 及び $y_j \leq y < y_{j+1}$ をみたく j 。 i を IX に, j を IY に入れる。 $XI(IX+1) \leq XP < XI(IX+2)$ $YJ(IY+1) \leq YP < YJ(IY+2)$
LX, LY	整 数 型	入 力	このサブルーチンは, $S(x, y)$ の偏微分, $\partial^{\lambda+\mu} S(x, y) / \partial x^\lambda \partial y^\mu$ を計算することができる。 評価すべき, $S^{(\lambda, \mu)}(x, y)$ における $\lambda, \mu, 0 \leq \lambda, \mu \leq k-1$ であること。
FP	実 数 型	出 力	$S^{(\lambda, \mu)}(x, y)$ の計算値。
NX, NY	整 数 型	入 力	CFS2A と同じ。
K	整 数 型	入 力	CFS2A と同じ。
XI, YJ	実 数 型 1 次 元 配 列	入 力	CFS2A と同じ。

引 数	型 と 種 類	属 性	内 容
CAB	実 数 型 2 次 元 配 列	入 力	B-spline に掛かる係数 $c_{\alpha, \beta} : -k+1 \leq \alpha \leq m-1, -k+1 \leq \beta \leq n-1$ 。 大きさ $(m+k-1)(n+k-1)$ 。 CFS2A の出力。
WORKF	実 数 型 1 次 元 配 列	作 業 領 域	大きさ, $m+5k-1$ 。
NXK1D	整 数 型	入 力	整合配列 CAB の第 1 添字。 $NXK1D \geq NX+K-1$ であること。
ICON	整 数 型	出 力	ICON=0: 正常終了。 ICON<0: 異常終了。

DCOMD1

DCPFR1

Curve Fitting by Composite Polynomials

複合多項式による曲線のあてはめ

作 成	秦野 和郎 1982年1月
形 式	サブルーチン 言語 ; FORTRAN サイズ ; 1491行

(1) 概 要

$N+1$ 個の等間隔離散点

$$\bar{x}_r = \frac{2\pi r}{N} : r = 0, 1, \dots, N \quad (1)$$

において関数値 $f(\bar{x}_r)$ が与えられたとき、複合多項式

$$h(x) = \frac{1}{2} c_0 + \sum_{j=1}^{n-1} (c_j \cos jx + b_j \sin jx) + \sum_{i=1}^m c_i q_i(x; n) \quad (2)$$

により $f(x) : 0 \leq x \leq 2\pi$ を最小二乗近似する。ここで、

$$\begin{cases} q_{2i}(x; n) = \sum_{j=n}^{\infty} \frac{(+)^{i-1}}{j^{2i}} \cos jx \\ q_{2i+1}(x; n) = \sum_{j=n}^{\infty} \frac{(+)^{i-1}}{j^{2i+1}} \sin jx \end{cases} \quad (3)$$

である。残差二乗和の定数倍

$$J = \frac{2}{N} \sum_{r=0}^N \{f(x_r) - h(x_r)\}^2 \quad (4)$$

を最小にするように $h(x)$ の係数 $a_0, a_j, b_j : j = 1, 2, \dots, n-1, c_i : i = 1, 2, \dots, m$ をきめる。(DCOMD1)。

また、与えられた等間隔離散点 $\bar{x}_s = 2\pi s/k : s_{20}, 1, \dots, k$ における式(2), $h(\bar{x}_s)$ は N の倍数でなければならない。

(2) 使用法

CALL DCOMD1 (FR, NL, ABJ, CJ, NS, MDEG, WORK, ICON)

引 数	型 と 種 類	属 性	内 容
FR	倍精度実数型 1次元配列	入 力	等間隔離散点における関数値 $f(\bar{x}_r)$ 。 $0 \leq r \leq N$, 大きさ $N+1$ 。
NL	整 数 型	入 力	N 。データの個数-1。 N は偶数でなければならない。
ABJ	倍精度実数型 1次元配列	出 力	$a_0, a_j, b_j : 1 \leq j \leq n-1$ が入る。 大きさ N (一部を作業領域として使う)。

引 数	型 と 種 類	属 性	内 容
CJ	倍精度実数型 1次元配列	出 力	$c_i : 1 \leq i \leq m$ が入る。大きさ m 。
NS	整 数 型	入 力	n を与える。
MDEG	整 数 型	入 力	m を与える。 m は12以下の偶数でなければならない。
WORK	倍精度実数型 1次元配列	作 業 領 域	大きさは N により変わる。 v を整数として $N = 2^v$ のとき大きさ1, そうでない偶数のとき大きさ N 。
ICON	整 数 型	出 力	ICON = 0 : 正常終了。 ICON < 0 : 異常終了。

CALL DCPFR1 (ABJ, CJ, NS, NCUT, MDEG, FR, NL, WORK, ICON)

引 数	型 と 種 類	属 性	内 容
ABJ	倍精度実数型 1次元配列	入 力	$a_0, a_j, b_j : 1 \leq j \leq n-1$ 。 DCOMD1の出力。大きさ N 。
CJ	倍精度実数型 1次元配列	入 力	$c_i : 1 \leq i \leq m$ 。DCOMD1の出力, 大きさ m 。
NS	整 数 型	入 力	n を与える。
NCUT	整 数 型	入 力	n を与える。(NSと同じ値にしておく)。
MDEG	整 数 型	入 力	m 。12以下の偶数であること。
FR	倍精度実数型 1次元配列	出 力	等間隔離散点における近似値 $k(\bar{x}_s) : 0 \leq s \leq K$ 。大きさ $K+1$ 。
NL	整 数 型	入 力	K を与える。計算したい近似値の個数-1。
WORK	倍精度実数型	作 業 領 域	大きさ K により変わる。 v を整数として $K = 2^v$ のとき大きさ1。そうでない偶数のとき大きさ K 。
ICON	整 数 型	出 力	ICON = 0 : 正常終了。 ICON < 0 : 異常終了。

LSAICS

LSAICD

Least Square Approximation by Orthogonal Polynomials

直交多項式による最小二乗近似

作 成	桑野 甯世 1976年2月
形 式	サブルーチン 言語; FORTRAN サイズ; 51, 53行

(1) 概 要

離散点 x_1, x_2, \dots, x_N 上での観測値 $\bar{f}_1, \bar{f}_2, \dots, \bar{f}_N$ から、未知関数 $f(x)$ の m 次直交多項式による最小二乗解¹⁾を求める。最適次数 m の決定を Minimum AIC Estimation 法²⁾により自動的に行う。次の四つのエントリー名から成る。

- LSAICS/D 近似多項式の最適次数 m を求める。
- LSFUNS/D 近似多項式 $y_m(x)$ の値を求める。
- LSCOFS/D $y_m(x) = \sum_{j=0}^m c_j^{(m)} x^j$ と書いたときの $c_j^{(m)}$ を求める。
- LSDEGS/D 近似多項式の次数を変更する。

(2) 使用法

- CALL LSAICS/D (X, F, W, N, MIN, MAX, MOPT, P, P1, P2, AIC, ILL)
- CALL LSFUNS/D (D, Y, K)
- CALL LSCOFS/D (COF)
- CALL LSDEGS/D (MD)

引 数	型 と 種 類*	属 性	内 容
X	実 数 型 1 次 元 配 列	入 力	N個の要素を持つ配列名。離散点 x_1, x_2, \dots, x_N 。必ずしも大小順でなくてもよい。
F	実 数 型 1 次 元 配 列	入 力	N個の要素を持つ配列名。X (1) における観測値をF (1) に入れる。
W	実 数 型 1 次 元 配 列	入 出 力	N個の要素を持つ配列名。入力としての意味は重みを入れる。ILL = 0 を指定すると出力の意味を持ち、すべてのWに1がセットされる。
N	整 数 型	入 力	離散点 x_i の総数。
MIN	整 数 型	入 力	近似多項式の次数の下限。 $0 \leq \text{MIN} \leq 20$
MAX	整 数 型	入 力	近似多項式の次数の上限。 $\text{MIN} \leq \text{MAX} \leq \min(N-1, 20)$
MOPT	整 数 型	出 力	近似多項式の最適次数。
P, P1, P2	実 数 型 1 次 元 配 列	入 出 力	作業領域。それぞれ、N個の要素を持つ1次元配列名。

引 数	型 と 種 類*	属 性	内 容
AIC	実 数 型 1 次 元 配 列	出 力	(MAX+1) 個の要素を持つ配列名。J 次式における AIC の値が AIC (J+1) に入る。
ILL	整 数 型	入 出 力	入力としての意味は W の項を参照。出力としての意味は、ILL = 0 のときに正常終了、ILL = 30000 のときは引数に対する制限が破られたため全く計算が行われなかったことを示す。
D	実 数 型	入 力	求めたい x 座標の値。
Y	実 数 型 1 次 元 配 列	出 力	K+1 個の要素を持つ配列名。D における近似多項式の値。Y (J) には J-1 次微係数が入る。K = 0 のときは実数型変数名でも可。
K	整 数 型	入 力	求めたい微係数の最大次数。
COF	実 数 型 1 次 元 配 列	出 力	(MOPT+1) 個の要素を持つ配列名。近似多項式を $y_m(x) = \sum_{j=0}^m c_j^{(m)} x^j$ と書いたときの係数 $c_j^{(m)}$ の値が低次より順に入る。 例えば、MOPT = 2 のとき $y_2(x) = \text{COF}(1) + \text{COF}(2) \times x + \text{COF}(3) \times x^2$ となる。ただし、 $y_m(x)$ の値を求める目的には LSFUNS/D を用いる方がよい。
MD	整 数 型	入 力	変更したい次数を入れる。LSDEGS/D をコールした後の LSFUNS/D 及び LSCOFS/D の Y, COF は MD 多項式による値になる。MD ≤ MAX

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 備 考

数値的に安定な直交多項式を基底にとる方法を採用し、観測点 x_i と観測値 \bar{f}_i のスケールリングを行い、桁落ちの起こらないよう配慮している。あてはめるべき多項式の次数の分からない場合には、AIC による次数自動決定の機能は有用である。

(4) 計算法

名古屋大学大型計算機センター「研究開発部研究報告 No 2」(昭和 51 年) p. 5 を参照。

参 考 文 献

- 1) A. Ralston; "A First Course in Numerical Analysis", McGraw-Hill, p.232 (1965).
- 2) 田辺国土; "数値計算における誤差", bit 臨時増刊号, pp. 113-125 (1975)。

TRIPCK

Two Dimensional C^k Interpolation Scheme for Irregularly Spaced Data

不規則分布 2 変数関数データに対する C^k 級補間法 ($0 \leq k \leq 3$)

作 成	佐藤 義雄 1979 年 1 月
形 式	サブルーチン 言語 ; FORTRAN サイズ ; 549 行

(1) 概 要

不規則に分布した 2 変数関数データ $x_i, y_i, f_i = f(x_i, y_i)$, ($i = 1, 2, \dots, N$) に対して, 各データ点 (x_i, y_i) を頂点とする三角メッシュの作成と各データ点における k 階までの偏微分値の算出とを經由して, 定義域 (凸多角形領域) 全体にわたって C^k 級となる補間関数を各三角形要素ごとに設定し, 定義域内の矩形格子状補間点における補間値を求める。

(2) 使用法

CALL TRIPCK (X, Y, F, N, P, M1, MX, MY, XL, YL, XU, YU, K, ICON)

引 数	型 と 種 類	属 性	内 容
X, Y	実 数 型 1 次 元 配 列	入 力	N個の要素をもつ1次元配列名。各データ点の x, y 座標。ただし, 同一座標の点が複数個あってはいけない。
F	実 数 型 1 次 元 配 列	入 力	N個の要素をもつ1次元配列名。各データ点における関数値。
N	整 数 型	入 力	データ点の個数。ただし, Nの大きさは, $K = 0, 1, 2, 3$ について各々 3, 6, 10, 15 以上 5000 以下でなければいけない。
P	実 数 型 2 次 元 配 列	出 力	$M1 \times MY$ 個の要素をもつ2次元配列名。矩形格子状補間点における補間値が入る。
M1	整 数 型	入 力	Pの配列宣言における第1添字の値。 $M1 \geq MX$
MX, MY	整 数 型	入 力	矩形格子状補間点の x 方向及び y 方向の分点数。
XL, YL	実 数 型	入 力	矩形格子状補間点の x 方向及び y 方向の下端の位置。
XU, YU	実 数 型	入 力	矩形格子状補間点の x 方向及び y 方向の上端の位置。
K	整 数 型	入 力	補間法が C^k 級であることを示す。ただし, $K = 0, 1, 2, 3$ 。 $K \neq 0, 1, 2, 3$ のとき, 補間部分をスキップする。
ICON	整 数 型	入 出 力	入力引数としては次の意味をもつ。 $ICON > 0$: 三角メッシュを作成し ($ICON = 1$ のときのみ), 各データ点における k 階 ($K = 1, 2, 3$ のとき) までの偏微分値を計算する。 $ICON \leq 0$: 上記の部分をスキップする。 出力引数としては次の意味をもつ。 $ICON = 0$: 正常。 $ICON < 0$: $ ICON $ は定義域外にある補間点の個数。定義域外の場合には, $(k+1)$ 次多項式を用いた最小二乗法によって補外値を求め, これを P に代入する。 $ICON = 30000$: 入力引数の制限が守られていない。

(3) 使用例

TRIPCK を使用するためのメイン・プログラムの主要部を次に示します。

```
DIMENSION X(500),Y(500),F(500),P(20,20)
      .....X, Y, F等の計算
      :
      ICON=1
      CALL TRIPCK(X,Y,F,400,P,20,10,10,1.0,1.0,10.0,10.0,1,ICON)
      :
      END
```

次のプログラムも全く同じ結果を与える。(備考(3)を参照)

```
DIMENSION X(500),Y(500),F(500),P(20,20)
      .....X, Y, F等の計算
      :
      ICON=1
      XP=1.0
      DO 1 I=1,10
      YP=1.0
      DO 2 J=1,10
      CALL TRIPCK(X,Y,F,400,P(I,J),1,1,1,XP,YP,XP,YP,1,ICON)
      2 YP=YP+1.0
      1 XP=XP+1.0
      :
      END
```

(4) 備考

1. このルーチンは、同一の X, Y, F, N, K の値に対しては、最初だけ ICON = 1 としてコールする。最初のコールで三角メッシュの作成と偏微分値の計算は完了し、ICON ≤ 0 となる。以後のステップでは同一データに対しては上記の部分スキップさせた方がよいので ICON ≤ 0 のままコールする。
2. 作成された三角メッシュの各三角形要素の3頂点の番号は、次のような名前付き COMMON 文
COMMON /CL9995/L1(9995), L2(9995), L3(9995), L
によって引用できる。ここに、L1(i), L2(i), L3(i), (i = 1, 2, ..., L) には、L 個の三角形要素について、その3頂点の番号がそれぞれ左まわりに格納されている。
3. ある一点(x_p, y_p)における補間値を求めたい場合は、
 $M1 = MX = MY = 1, XL = x_p, YL = y_p$
とすればよい。この場合、出力引数 P は、単なる実数型の変数でかまわない。
4. 不規則分布2変数関数データ(5000点以下)に対する三角メッシュの作成と等高線表示のためのプログラム TRIMAP も用意されている。
「図形出力の手引」p. 110 を参照されたい。

参考文献

- 1) 佐藤義雄：“不規則分布データに対する等高線表示と C^k 級補間法”，名古屋大学大型計算機センターニュース，Vol. 10, No 2, p. 161 (1979)。
- 2) 佐藤義雄，二宮市三：“不規則分布二変数関数データに対する C^k 級補間法”，情報処理学会論文誌 Vol. 22, p. 581, (1981)。

TETPCK

Three Dimensional C^k Interpolation Scheme for Irregularly Spaced Data 不規則分布 3 変数関数データに対する C^k 級補間法 ($0 \leq k \leq 1$)

作 成	佐藤 義雄 1979 年 1 月
形 式	サブルーチン 言語 ; FORTRAN サイズ ; 970 行

(1) 概 要

不規則に分布した 3 変数関数データ $x_i, y_i, z_i, f_i = f(x_i, y_i, z_i)$, ($i = 1, 2, \dots, N$) に対して, 各データ点 (x_i, y_i, z_i) を頂点とする四面体メッシュの作成と各データ点における k 階までの偏微分値の算出とを經由して, 定義域 (凸多面体領域) 全体にわたって C^k 級となる補間関数を各四面体要素ごとに設定し, 定義域内の直方体格子状補間点における補間値を求める。

(2) 使用法

CALL TETPCK (X, Y, Z, F, N, P, M1, M2, MX, MY, MZ, XL, YL, ZL, XU, YU, ZU,
K, ICON)

引 数	型 と 種 類	属 性	内 容
X, Y, Z	実 数 型 1 次 元 配 列	入 力	N個の要素をもつ配列名。各データ点の x, y, z 座標。ただし同一座標の点が複数個あってはいけない。
F	実 数 型 1 次 元 配 列	入 力	N個の要素をもつ配列名。各データ点における関数値。
N	整 数 型	入 力	データ点の個数。ただし, Nの大きさは, $K = 0, 1$ について各々 4, 10 以上, 1000 以下でなければならない。
P	実 数 型 3 次 元 配 列	出 力	$M1 \times M2 \times MZ$ 個の要素をもつ配列名。直方体格子状補間点における補間値が入る。
M1, M2	整 数 型	入 力	Pの配列宣言における第1添字及び第2添字の値。 $M1 \geq MX, M2 \geq MY$
MX, MY, MZ	整 数 型	入 力	直方体格子状補間点の x 方向, y 方向, z 方向の分点数。
XL, YL, ZL	実 数 型	入 力	直方体格子状補間点の x 方向, y 方向, z 方向の下端の位置。
XU, YU, ZU	実 数 型	入 力	直方体格子状補間点の x 方向, y 方向, z 方向の上端の位置。
K	整 数 型	入 力	補間法が C^k 級であることを示す。ただし, $K = 0, 1$ 。 $K \neq 0, 1$ のとき, 補間部分をスキップする。
ICON	整 数 型	入 出 力	入力引数としては次の意味をもつ。 $ICON > 0$: 四面体メッシュを作成し ($ICON = 1$ のときのみ), 各データ点における 1 階 ($K = 1$ のとき) の偏微分値を計算する。 $ICON \leq 0$: 上記の部分をスキップする。 出力引数としては次の意味をもつ。

引 数	型 と 種 類	属 性	内 容
			ICON = 0: 正常。 ICON < 0: ICON は定義域外にある補間点の個数。定義域外の場合には、 $(k+1)$ 次多項式を用いた最小二乗法によって補外値を求め、これを P に代入する。 ICON = 30000: 入力引数の制限が守られていない。 ICON = 10000: 作業領域不足等による Break Down。 (稀少。この場合には、エラー・メッセージを印字する。)

(3) 使用例

TETPCK を使用するためのメイン・プログラムの主要部を次に示します。

```

DIMENSION X(350),Y(350),Z(350),F(350),P(20,20,20)
      .....X, Y, Z, F等の計算
      ICON=1
      CALL TETPCK(X,Y,Z,F,125,P,20,20,10,10,2,1.0,1.0,4.0,10.0,10.0,7.0,1,ICON)
      END
  
```

次のプログラムも全く同じ結果を与える。(備考3を参照)

```

DIMENSION X(350),Y(350),Z(350),F(350),P(20,20,20)
      .....X, Y, Z, F等の計算
      ICON=1
      XP=1.0
      DO 1 I=1,10
        YP=1.0
      DO 2 J=1,10
        ZP=4.0
      DO 3 K=1,2
        CALL TETPCK(X,Y,Z,F,125,P(I,J,K),1,1,1,1,XP,YP,ZP,XP,YP,ZP,1,ICON)
3     ZP=ZP+3.0
2     YP=YP+1.0
1     XP=XP+1.0
      END
  
```

(4) 備 考

- このルーチンは、同一の X, Y, Z, F, N, K の値に対しては、最初だけ ICON = 1 としてコールする。最初のコールで四面体メッシュの作成と偏微分値の計算は完了し、ICON ≤ 0 となる。以後のステップでは、同一データに対しては上記の部分スキップさせた方がよいので ICON ≤ 0 のままコールする。
- 作成された四面体メッシュの各四面体要素の4頂点の番号は、次のような名前付き COMMON 文
 COMMON /CL8000/L0(8000),L1(8000),L2(8000),L3(8000),L
 によって引用できる。ここに、L0(i), L1(i), L2(i), L3(i), (i = 1, 2, ..., L)には、L 個の四面体要素について、その4頂点の番号が、それぞれ、頂点 L0(i)から見て3頂点 L1(i), L2(i), L3(i)が右まわりとなるように格納されている。
- ある一点 (x_p, y_p, z_p) における補間値を求めたい場合は、
 M1 = M2 = MX = MY = MZ = 1, XL = x_p, YL = y_p, ZL = z_p
 とすればよい。この場合、出力引数 P は、単なる実数型の変数でかまわない。

FFTC / B

Complex Fast Fourier Analysis

複素高速フーリエ解析

作 成	二宮 市三 1981年4月
形 式	サブルーチン 言語 ; アセンブラ サイズ ; 267, 267行

(1) 概 要

周期関数の1周期の N 等分点でのサンプル値 $X_j, j = 0, 1, \dots, N-1$ (X_0 は原点での値である) が与えられたとき, その周期成分 $C_j, j = 0, 1, \dots, N-1$ は, フーリエ変換

$$C_j = \sum_{k=0}^{N-1} X_k W^{jk}, \quad j = 0, 1, \dots, N-1$$

で与えられる。ただし, $W = \exp\left(\frac{-2\pi i}{N}\right)$ である。また, 逆に周期成分 C_j が与えられたとき, X_j は逆変換

$$X_j = \sum_{k=0}^{N-1} C_k W^{-jk}, \quad j = 0, 1, \dots, N-1$$

で与えられる。本ルーチンは, $N = 2^M$ の形するとき, 上の計算を複素高速フーリエ変換の手法で行うためのものである。

(2) 使用法

CALL FFTC/B(A, M, INV, ILL)

引 数	型 と 種 類*	属 性	内 容
A	複 素 数 型 1 次 元 配 列	入 出 力	正変換のとき X_k を入力して C_j を出力し, 逆変換のとき C_k を入力して X_j を出力する。A(j) には $C_{j-1}(X_{j-1})$ が入る。
M	整 数 型	入 力	配列Aの大きさが 2^M であることを表す。 $M \geq 2$
INV	整 数 型	入 力	INV = 0 のとき正変換を, INV = 1 のとき逆変換を行うことを意味する。
ILL	整 数 型	出 力	ILL = 0 : 正常終了。 ILL = 30000 : $M \leq 1$ のとき。

* FFTB の場合は, 複素数型を倍精度複素数型とする。

(3) 性 能

本ルーチンは, 4 を基底とする高速複素フーリエ変換の手法が用いられている上に, アセンブリ言語で書かれているので, 高速であり, 精度もよい。

(4) 備 考

本ルーチンと同じ目的のために FFTS/D があるが, 引数の意味や, フーリエ変換の定義が多少異なるので注意されたい。FFTS/D では入力ベクトル A と同じ大きさの作業領域 B を必要としているが, 本ルーチンではそのようなことはなく, スピードも速いので本ルーチンを用いる方が有利である。

FFTS/D

Complex Fast Fourier Transform

複素高速フーリエ変換

作成	二宮 市三 1977年4月
形式	サブルーチン 言語; FORTRAN サイズ; 124, 129行

(1) 概要

周期関数の1周期の N 等分点でのサンプル値 X_j , $j = 0, 1, \dots, N-1$ (X_0 は原点での値である)が与えられたとき, その各周期成分 C_j , $j = 0, 1, \dots, N-1$ は, フーリエ変換

$$C_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X_k W^{jk}, \quad j = 0, 1, \dots, N-1$$

で与えられる。ただし, $W = \exp\left(\frac{-2\pi i}{N}\right)$ である。また, 逆に周期成分 C_j が与えられたとき, X_j は, 逆変換

$$X_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} C_k W^{-jk}, \quad j = 0, 1, \dots, N-1$$

で与えられる。本ルーチンは, N が 2^M の形のとき, 上の計算を高速フーリエ変換の手法で行うためのものである。

(2) 使用法

CALL FFTS/D (A, B, N, INV, ILL)

引数	型と種類*	属性	内容
A	複素数型 1次元配列	入出力	正変換のとき X_k を入力して C_j を出力し, 逆変換のとき C_k を入力して X_j を出力する。A(j)には C_{j-1} (X_{j-1})が入る。
B	複素数型 1次元配列	作業領域	サブルーチン内で使われる作業用領域。
N	整数型	入力	配列A, Bの大きさを表す。 2^M の形でなければならない。 $N \geq 2$
INV	整数型	入力	INV = 0のとき正変換, INV = 1のとき逆変換を行うことを意味する。
ILL	整数型	出力	ILL = 0 : 正常終了。 ILL = 30000 : N が 2^M ($M > 0$)の形でないとき。

(3) 性能

本ルーチンは高速フーリエ変換の手法が用いられている上に, 次のような特徴を持っているので非常に速く, 精度もよい。

1. 正弦, 余弦の値は引数の絶対値が $\pi/8$ 以内のときだけ計算し, 一度計算した値は小さな修正を加えつつ, 8回にわたって利用する。

2. 正弦、余弦の基本外部関数をコールせず、ルーチン内に用意された低次の近似多項式を用いる。

(4) 備考

1. 普通、フーリエ変換は、

$$C_j = \frac{1}{N} \sum_{k=0}^{N-1} X_k W^{jk}, \quad j = 0, 1, \dots, N-1$$

また、逆変換は、

$$X_j = \sum_{k=0}^{N-1} C_k W^{-jk}, \quad j = 0, 1, \dots, N-1$$

と定義されるが、本ルーチンはこの定義と異なる定義を用いているので注意されたい。

2. 実数入力データの場合には、そのための専門ルーチン FFTR、FFTRI が用意されているので、それらを利用されたい。

3. 本ルーチンと同様の機能をもつものに FFTC/B がある。適宜選択して利用されたい。

FFTR / FFTRD

Real Fast Fourier Analysis

実高速フーリエ解析

作成	二宮 市三 1981年4月
形式	サブルーチン 言語; アセンブラ サイズ; 214, 214行

(1) 概要

原点を左端とする、実周期関数の1周期を $N = 2^M$ 等分した点での値、 $X_j, j = 0, 1, \dots, N-1$ を入力すると、その余弦成分 $C_j, j = 0, 1, \dots, N/2$ と正弦成分 $S_j, j = 1, 2, \dots, N/2-1$ を実高速フーリエ解析の手法を用いて計算する。ただし、

$$C_j = \frac{\epsilon_j}{N} \sum_{k=0}^{N-1} X_k \cos \frac{2jk\pi}{N}, \quad j = 0, 1, \dots, N/2$$

$$\epsilon_0 = \epsilon_{N/2} = 1; \quad \epsilon_j = 2, \quad j = 2, \dots, N/2-1$$

$$S_j = \frac{2}{N} \sum_{k=0}^{N-1} X_k \sin \frac{2jk\pi}{N}, \quad j = 1, 2, \dots, N/2-1$$

(2) 使用法

CALL FFTR (A, M, ILL)

CALL FFTRD (A, M, ILL)

引数	型と種類*	属性	内容
A	実数型 1次元配列	入出力	2^M 個の成分を持つ1次元配列名。周期関数の1周期の 2^M 等分点での値を原点での値から順に入れると、余弦成分、正弦成分がこの順に、そして、各成分内では次数の順に入る。すなわち、 K 次の余弦成分はA(K+1)に、 J 次の正弦成分はA(N/2+J+1)に入る。
M	整数型	入力	1周期を 2^M 等分することを示す。 $M \geq 0$
ILL	整数型	出力	$M < 0$ ならば ILL = 30000として計算をしない。それ以外は計算が行われ、ILL = 0となる。

* FFTRDの場合は、実数型を倍精度実数型とする。

(3) 性能

アセンブリ言語で書かれていて無駄がない上に、三角関数の計算回数を減らすように工夫がなされているので、非常に速く、また、精度もよい。

(4) 計算法

ベルグランド¹⁾の算法とは異なり、初めにビット逆転による並べ換え(サブルーチン BITREV をコールする)を行う方法を独自に考案した。

(5) 備考

フーリエ解析の方法は外にもいろいろあるが、特別のことがない限り、分割数は 2^M の形にして、実高速フーリエ解析の方法を用いるべきである。

参考文献

- 1) G.D. Bergland; "A Fast Fourier Transform Algorithm for Real-Valued Series", *Comm. ACM*, Vol.11, pp.703-710 (1968).

FFTRI / FFTRID

Real Fast Fourier Synthesis

実高速フーリエ合成

作成	二宮 市三 1981年4月
形式	サブルーチン 言語 ; アセンブラ サイズ ; 196, 196行

(1) 概要

実周期関数の余弦成分 $C_j, j = 0, 1, \dots, N/2$ と正弦成分 $S_j, j = 1, 2, \dots, N/2-1$ を入力すると、その関数の原点を左端とする 1 周期の N 等分点での値 $X_k, k = 0, 1, \dots, N-1$ を、実高速フーリエ解析の手法で計算する。ただし、

$$X_k = \sum_{j=0}^{N/2} C_j \cos \frac{2\pi jk}{N} + \sum_{j=1}^{N/2-1} S_j \sin \frac{2\pi jk}{N}, \quad k = 0, 1, \dots, N-1$$

で N は、 $N = 2^M$ の形の整数である。

(2) 使用法

CALL FFTRI (A, M, ILL)

CALL FFTRID (A, M, ILL)

引数	型と種類*	属性	内容
A	実数型 1次元配列	入出力	2^M 個の成分を持つ1次元配列名。 K 次の余弦成分を $A(K+1)$ に、 J 次の正弦成分を $A(N/2+J+1)$ に入れると、1周期の 2^M 等分点での値が原点での値から順に入る。
M	整数型	入力	1周期を 2^M 等分することを示す。 $M \geq 0$
ILL	整数型	出力	$M < 0$ ならば $ILL = 30000$ として計算をしない。それ以外は計算が行われ、 $ILL = 0$ となる。

* FFTRID の場合は、実数型を倍精度実数型とする。

(3) 性能

アセンブリ言語で書かれていて無駄がない上に、三角関数の計算回数を減らす工夫がなされているので、非常に速く、また、精度もよい。

(4) 計算法

実高速フーリエ解析 (FFTR, FFTRD) の算法を逆順をたどって得られる。FFTR の参考文献¹⁾を参照されたい。

FT235C

FT235R

Complex and Real Fast Fourier Transform for The Case of Sample Number of The Form of $2^K 3^L 5^M$

サンプル数が $2^K 3^L 5^M$ の形の場合の複素及び実高速フーリエ変換

作成	二宮 市三	1977年4月
形式	サブルーチン 言語 ; FORTRAN サイズ ; 178, 42行	

(1) 概要

1周期の分割数が $N = 2^K 3^L 5^M$ の形で表される場合に、複素高速フーリエ解析 (FT235C) と、実高速フーリエ変換 (FT235R) を行うためのサブルーチンである。

FT235C での諸定義は FFTC でのものと、また、FT235R での諸定義は FFTR でのものと同一であるので、それぞれの解説を参照されたい。

(2) 使用法

CALL FT235C (A, B, N, INV, ILL)

CALL FT235R (A, B, N, ILL)

引数	型と種類		属性	内容
	FT235C	FT235R		
A	複素数型 1次元配列		入出力	N 個の成分を持つ1次元配列名。正変換のとき X_k を入力して C_j を出力し、逆変換のとき C_k を入力して X_j を出力する。 $A(j)$ には $C_{j-1} (X_{j-1})$ が入る。
		実数型 1次元配列	入出力	N 個の成分を持つ1次元配列名。周期関数の1周期の N 等分点での値を順に入れると、余弦成分、正弦成分がこの順に、そして、各成分内では次数の順に出力される。 K 次の余弦成分は $A(K+1)$ に、 J 次の正弦成分は $A(N/2+J+1)$ に出力される。
B	複素数型 1次元配列	実数型 1次元配列	作業領域	サブルーチン内で使用される。作業領域Aと同じ型、同じ大きさが必要。
N	整数型	整数型	入力	1周期の分割数で、 $N = 2^K 3^L 5^M$ の形でなければならない。 $N > 2$ FT235R の場合は $K \geq 1$ でなければならない。
INV	整数型		入力	INV = 0 のとき正変換、INV = 1 のとき逆変換を行うことを意味する。
ILL	整数型	整数型	出力	ILL = 30000 : 入力に関する制限が満たされないときそれ以外は0となる。

* FFTDの場合は、複素数をすべて倍精度複素数とする。

(3) 性能

本ルーチンは、 $N = 2^M$ 型でないので、他のルーチンに比べてスピードがおそい。したがって、 2^M 型の場合には、そのための専門ルーチンを用いるのが合理的である。

(4) 使用例

「CALL FT235R(A,B,N,ILL)」とすると、FT235Rの中で「CALL FT235C(A,B,N/2,0,ILL)」というCALLが行われる。したがって、Nは偶数($N = 2^K 3^L 5^M$, $K \geq 1$)でなければならない。また、A,Bは、このCALLで複素数1次元配列の扱いを受ける($A(1)+iA(2)$, $A(3)+iA(4)$ …という $N/2$ の大きさの配列)ので、そのような扱いを受けてもよいような準備が必要である。そのための一つの方法は、次の用例にあるようなEQUIVALENCE文の使用である。(A,Bの先頭が偶数番地に割り当てられることが必要十分である。)

```
C      MAIN PROGRAM
      DIMENSION A(720),B(720)
      COMPLEX CA(360),CB(360)
      EQUIVALENCE (A,CA),(B,CB)
      READ(5,500) (A(I),I=1,720)
500   FORMAT(6F12.0)
      CALL FT235R(A,B,720,ILL)
      :
      STOP
      END
```

(5) 備考

FT235Rを用いる場合には、分割数Nは $N = 2^K 3^L 5^M$ の形であって、しかも偶数でなければならない。また、配列A,Bは、大きさNの実数の1次元配列であるとともに、大きさ $N/2$ の複素数1次元配列の取扱いを受けるので、そのための処置が必要である。用例を参照されたい。

FFT2DC, FFT3DC

2- and 3-Dimensional Complex Fast Fourier Transform

2次元及び3次元複素高速フーリエ変換

作成	二宮 市三	1982年5月
形式	サブルーチン 言語; FORTRAN 77 サイズ; 21, 30行	

(1) 概要

FFT2DC } は { 2次元 } 複素高速フーリエ変換のためのサブルーチン。
 FFT3DC } { 3次元 }

2次元の場合についてのみ説明する。2次元の周期複素数値関数の基本周期長方形の $N_1 (= 2^{M_1}) \times N_2 (= 2^{M_2})$ 等分メッシュ点での関数値 $X_{rs}; r = 0, 1, \dots, N_1 - 1; s = 0, 1, \dots, N_2 - 1$ (ただし, X_{00} は原点での値) が与えられたとき, そのフーリエ変換は

$$C_{kl} = \frac{1}{N_1 N_2} \sum_{r=0}^{N_1-1} \sum_{s=0}^{N_2-1} X_{rs} e^{-\frac{2\pi i k r}{N_1}} e^{-\frac{2\pi i l s}{N_2}}, \quad k = 0, 1, \dots, N_1 - 1; l = 0, 1, \dots, N_2 - 1$$

で与えられる。これを正変換という。逆に C_{kl} を周期成分とする周期関数の基本周期長方形の2のべき乗等分メッシュ点での関数値

$$X_{rs} = \sum_{k=0}^{N_1-1} \sum_{l=0}^{N_2-1} C_{kl} e^{\frac{2\pi i k r}{N_1}} e^{\frac{2\pi i l s}{N_2}}, \quad r = 0, 1, \dots, N_1 - 1; s = 0, 1, \dots, N_2 - 1$$

を求めることを逆変換という。

(2) 使用法

CALL FFT2DC(A, KA, M, INV, W, ILL)

CALL FFT3DC(A, KA, LA, M, INV, W, ILL)

引数	型と種類	属性	内容
A	複素数型 2次元配列 (3次元配列)	入出力	正変換: X を入力すると $N_1 N_2 C(N_1 N_2 N_3 C)$ が出力される。 逆変換: C を入力すると, X が出力される。 大きさ $2^{M(1)} \times 2^{M(2)} (2^{M(1)} \times 2^{M(2)} \times 2^{M(3)})$ 。
KA	整数型	入力	A の配列宣言での第1添字の値。 $KA \geq 2^{M(1)}$
LA	整数型	入力	A の配列宣言での第2添字の値。 $LA \geq 2^{M(2)}$
M	整数型 1次元配列	入力	$2^{M(1)}, 2^{M(2)}, 2^{M(3)}$ が各軸方向の等分割数を表わす。 $M(1) > 1, M(2) > 1, M(3) > 1$
INV	整数型	入力	INV = 0 のとき正変換を行う。 INV = 1 のとき逆変換を行う。
W	複素数型 1次元配列	作業領域	2次元の場合大きさ $2^{M(2)}$ 。 3次元の場合大きさ $\max(2^{M(2)}, 2^{M(3)})$ 。
ILL	整数型	出力	ILL = 0: 正常終了。 ILL = 30000: 引数エラー。

(3) 使用例

複素周期関数

$$f(x, y) = (1 + 2ie^{2\pi ix} + 3e^{4\pi ix})(-1 - 2ie^{2\pi iy})$$

の基本周期正方形 $[0, 1]^2$ の 128×128 等分メッシュ点での関数値を、逆変換で求め、これに正変換を施す。

```
COMPLEX*8 A,B,C,S
DIMENSION A(128,128),B(128,128),C(128),S(2),M(2)
N=128
C(1)=1.0
C(2)=(0.,2.)
C(3)=3.
S(1)=-1.
S(2)=(0.,-2.)
DO 10 J=1,N
DO 10 I=1,N
A(I,J)=0.
10 B(I,J)=0.
DO 20 J=1,2
DO 20 I=1,3
A(I,J)=C(I)*S(J)
20 B(I,J)=A(I,J)
KA=N
M(1)=7
M(2)=7
INV=1
CALL FFT2DC(A,KA,M,INV,C,ILL)
INV=0
CALL CLOCKM(IO)
CALL FFT2DC(A,KA,M,INV,C,ILL)
CALL CLOCKM(I1)
IT=I1-IO
D=1./FLOAT(N)**2
DO 30 J=1,N
DO 30 I=1,N
30 EM=AMAX1(CABS(A(I,J)*D-B(I,J)),EM)
WRITE(6,600) IT,ILL,EM
600 FORMAT(10X,'TIME =',I7,'MS',2X,'ILL =',I6,2X,'EM =',E11.3)
STOP
END
```

(4) 摘要

正変換のときにえられる出力は、フーリエ変換そのものではなく、その $N_1 N_2$ ($N_1 N_2 N_3$) 倍である。引数 A の説明、使用例ならびにサブルーチン FFTC の説明を参照されたい。

FFT2DR, FFT3DR

2- and 3-Dimensional Real Fast Fourier Analysis and Synthesis

2次元及び3次元実高速フーリエ解析ならびに合成

作成	二宮 市三	1982年5月
形式	サブルーチン 言語; FORTRAN 77 サイズ; 28, 40行	

(1) 概要

FFT2DR } は { 2次元 } 実高速フーリエ解析ならびに合成のためのサブルーチン。
 FFT3DR } は { 3次元 }

2次元の場合についてのみ説明する。2次元の実周期関数の基本周期長方形の $N_1(=2^M) \times N_2(=2^M)$ 等分メッシュ点での関数値 F_{rs} ; $r=0, 1, \dots, N_1-1$; $s=0, 1, \dots, N_2-1$ (ただし F_{00} は原点での値) が与えられたとき, その余弦(C) 正弦(S)成分は

$$\begin{Bmatrix} C \\ S \end{Bmatrix}_{kl} = \frac{\epsilon_1 \epsilon_2}{N_1 N_2} \sum_{r=0}^{N_1-1} \sum_{s=0}^{N_2-1} F_{rs} \begin{Bmatrix} \cos \frac{2\pi kr}{N_1} \\ \sin \frac{2\pi kr}{N_1} \end{Bmatrix} \begin{Bmatrix} \cos \frac{2\pi ls}{N_2} \\ \sin \frac{2\pi ls}{N_2} \end{Bmatrix}, \quad k=0, 1, \dots, N_1/2; l=0, 1, \dots, N_2/2$$

で与えられる。ただし,

$$\epsilon_1 = \begin{cases} 2, & 0 < k < N_1/2 \\ 1, & k=0, N_1/2 \end{cases}$$

$$\epsilon_2 = \begin{cases} 2, & 0 < l < N_2/2 \\ 1, & l=0, N_2/2 \end{cases}$$

であり,

$$\begin{cases} CS_{kl} = 0, & l=0, N_2/2 \\ SC_{kl} = 0, & k=0, N_1/2 \\ SS_{kl} = 0, & k=0, N_1/2 \text{ 又は } l=0, N_2/2 \end{cases}$$

である。上記の計算をフーリエ解析という。逆に余弦, 正弦成分より基本周期長方形の等分メッシュ点での関数値

$$F_{rs} = \sum_{k=0}^{N_1/2} \left\{ \cos \frac{2\pi kr}{N_1} \sum_{l=0}^{N_2/2} \left(CC_{kl} \cos \frac{2\pi ls}{N_2} + CS_{kl} \sin \frac{2\pi ls}{N_2} \right) \right. \\ \left. + \sin \frac{2\pi kr}{N_1} \sum_{l=0}^{N_2/2} \left(SC_{kl} \cos \frac{2\pi ls}{N_2} + SS_{kl} \sin \frac{2\pi ls}{N_2} \right) \right\}, \quad r=0, 1, \dots, N_1-1; s=0, 1, \dots, N_2-1$$

を計算することをフーリエ合成という。

(2) 使用法

CALL FFT2DR (A, KA, M, INV, W, ILL)

CALL FFT3DR (A, KA, LA, M, INV, W, ILL)

引 数	型 と 種 類	属 性	内 容
A	実 数 型 2 次 元 配 列 (3 次 元 配 列)	入 出 力	フーリエ解析： F を入力すると余弦正弦成分が出力される。 出力の格納順序は1次元の場合の直積である。例えば2次元の場合 CS_{IJ} は $A(I+1, 2*(M(2)-1)+1+J)$ に格納される。 フーリエ合成： 余弦正弦成分を上掲の順序に格納して入力すると、関数値 F が自然の順序に出力される。 大きさ $2^{M(1)} \times 2^{M(2)} (2^{M(1)} \times 2^{M(2)} \times 2^{M(3)})$ 。
KA	整 数 型	入 力	A の配列宣言での第1添字の値。 $KA \geq 2^{M(1)}$
LA	整 数 型	入 力	A の配列宣言での第2添字の値。 $LA \geq 2^{M(2)}$
M	整 数 型 1 次 元 配 列	入 力	$2^{M(1)}, 2^{M(2)}, 2^{M(3)}$ が各軸方向の等分数を表わす。 $M(1) > 1, M(2) > 1, M(3) > 1$ 。
INV	整 数 型	入 力	INV = 0 のときフーリエ解析を行う。 INV = 1 のときフーリエ合成を行う。
W	実 数 型 1 次 元 配 列	作 業 領 域	2次元の場合大きさ $2^{M(2)}$ 。 3次元の場合大きさ $\max(2^{M(2)}, 2^{M(3)})$ 。
ILL	整 数 型	出 力	ILL = 0 : 正常終了。 ILL = 30000 : 引数エラー。

(3) 使用例

周期関数

$$f(x, y) = (1 + \cos 2\pi x + 2\cos 4\pi x)(-\sin 2\pi x - 2\sin 4\pi x)$$

の基本周期正方形 $[0, 1]^2$ の 128×128 等分メッシュ点での関数値を、フーリエ合成で求め、これをフーリエ解析する。

```

DIMENSION A(128,128),B(128,128),C(128),S(2),M(2)
N=128
C(1)=1.0
C(2)=2.
C(3)=3.
S(1)=-1.
S(2)=-2.
DO 10 J=1,N
DO 10 I=1,N
A(I,J)=0.
10 B(I,J)=0.
NH1=N/2+1
DO 20 J=1,2
DO 20 I=1,3
A(I,J+NH1)=C(I)*S(J)
20 B(I,J+NH1)=A(I,J+NH1)
KA=N
M(1)=7
M(2)=7
INV=1
CALL FFT2DR(A,KA,M,INV,C,ILL)
INV=0
CALL CLOCKM(IO)
CALL FFT2DR(A,KA,M,INV,C,ILL)
CALL CLOCKM(I1)
IT=I1-IO
DO 30 J=1,N
DO 30 I=1,N

```

```

30 EM=AMAX1(ABS(A(I,J))-B(I,J)),EM)
WRITE(6,600) IT,ILL,EM
600 FORMAT(10X,'TIME =',I7,'MS',2X,'ILL =',I6,2X,'EM =',E11.3)
STOP
END

```

(4) 摘要

余弦正弦成分の格納順序は、1次元の実フーリエ解析の場合の直積である。引数Aの説明、使用例及びサブルーチンFFTRの説明を参照されたい。

BITREV / BITRVD / BITRVB

Rearrangement of Data by Bit Reversal

ビット逆転によるデータの並べ換え

作成	二宮 市三 1981年4月
形式	サブルーチン 言語; アセンブラ サイズ; 108, 110, 114行

(1) 概要

高速フーリエ変換に必要なビット逆転による並べ換えを行うサブルーチンである。ビット逆転というのは、2進数の各ビットを逆順に並べ換えることである。 M 桁の2進数 K をビット逆転したものを \bar{K} で表すと、本ルーチンで行うことは、1から 2^M までの整数 K に対して、 $A(K)$ を $A(\bar{K}-1+1)$ に納めかえることである。

(2) 使用法

CALL BITREV (A, M, ILL)

CALL BITRVD (A, M, ILL)

CALL BITRVB (A, M, ILL)

引数	型と種類*	属性	内容
A	実数型 1次元配列	入出力	2^M 個の要素を持つ1次元配列名。本ルーチンによりビット逆転による並べ換えが行われる。
M	整数型	入力	配列Aの大きさが 2^M であることを表す。 $M \geq 0$
ILL	整数型	出力	$M < 0$ のとき $ILL = 30000$ となり、計算は行われない。その他はすべて計算が行われ、 $ILL = 0$ となる。

* BITRVD (BITRVB) の場合は、実数型を倍精度実数型 (倍精度複素数型) とする。

(3) 性能

アセンブラで書かれていて無駄がない上に、アルゴリズムとコーディングに工夫がこらしてあって非常に速い。

(4) 計算法

参考文献¹⁾を参照されたい。

参考文献

- 1) 二宮市三; "ビット逆転スクランプリングの一方", 情報処理学会第23回全国大会予稿集, pp. 899-900 (1981)。

FCOSMS / D

FSINMS / D

Fast Fourier Cosine Transform Based on The Midpoint Rule (FCOSM/D)

中点公式に基づく高速 cosine 変換 (FCOSMS/D)

Fast Fourier Sine Transform Based on The Midpoint Rule (FSINM/D)

中点公式に基づく高速 sine 変換 (FSINMS/D)

作 成	鳥居 達生 1978年12月
形 式	サブルーチン 言語 ; FORTRAN サイズ ; 165, 166, 165, 166行

(1) 概 要

周期 2π の関数 $X(t)$ の半周期を N 等分して N 個の標本

$$X_{j+\frac{1}{2}} = X\left[\frac{\pi}{N}\left(j+\frac{1}{2}\right)\right], \quad 0 \leq j < N, N = 2^n$$

がつくられているとする。 $X(t)$ が偶関数のとき、

$$B_k = \sum_{0 \leq j < N} X_{j+\frac{1}{2}} \cos \frac{\pi}{N} k \left(j+\frac{1}{2}\right), \quad 0 \leq k < N$$

奇関数のとき、

$$B_k = \sum_{0 \leq j < N} X_{j+\frac{1}{2}} \sin \frac{\pi}{N} k \left(j+\frac{1}{2}\right), \quad 0 < k \leq N$$

を求める。

(2) 使用法

三角関数表の計算 (TRIGQP, TRIGQD) と入力データを2進逆順に並べ換えること (BITREV, BITRVD) が、あらかじめ必要である。すなわち、

CALL TRIGQP (W, MW, ICON)

CALL BITREV (X, MX, ICON)

倍精度の場合は

CALL TRIGQD (W, MW, ICON)

CALL BITRVD (X, MX, ICON)

と準備した後、目的のサブルーチンを呼ぶ。

CALL FCOSMS/D (X, MX, LX, W, MW, ICON)

CALL FSINMS/D (X, MX, LX, W, MW, ICON)

引 数	型 と 種 類	属 性	内 容
X	実 数 型 1 次 元 配 列	入 出 力	配列 X の大きさ $\geq 2^{MX}$ 。入力の標本数 = 2^{MX} 。LX は配列 X 上において、入力データの先頭番地を指定する。すなわち、 $X(LX+1), X(LX+2), \dots, X(LX+2^{MX})$ が入力され、出力
MX	整 数 型		

引 数	型 と 種 類	属 性	内 容
LX	整 数 型		は、この上に重ね書きされる。 FCOSM/Dの場合 $B_{k-1} = X(LX+k)$ 。 FSINM/Dの場合 $B_2^{MX-k+1} = X(LX+k)$ 。 入力MX, LXは保存される。MX \geq 1, LX \geq 0
W MW	実 数 型 1 次 元 配 列 整 数 型	入 力	配列Wの大きさ $\geq 2^{MW}-1$ 。保存される。MW \geq MX
ICON	整 数 型	出 力	ICON = 0 : 正常。 ICON = 30000 : パラメータ・エラー。

(3) 性 能

N 項 cosine (sine) 変換に必要な実数乗算回数は $N \log_2 N$ ($N = 2^n$) である。出力データは入力データの上に重ね書きされる。算法は安定である。

(4) 計算法

中点公式に基づく高速 cosine (sine) 変換の算法³⁾を入力データの上に出力データが重ね書きとなるよう工夫した。入力データの実数性のみならず対称性も利用しているので、実数データの FFT に比べ、演算回数及び作業領域が半減されている。

(5) 使用例

入力データの個数が 2^M のとき、少なくとも 2^M-1 の大きさの三角関数表 (TRIGQP) を計算しておく。入力データが番号順に並んでいるならば2進逆順に並べ換えた後、本サブルーチンを呼ぶ。

入力データの先頭番地を LX+1 と浮動的にしたのは、LX を適当にとって台形公式に基づく高速 cosine 変換、チェビシェフ級数展開などに本サブルーチンを補助的に使うためである。単に $X(1), X(2), \dots, X(2^M)$ を cosine 変換する場合であれば LX = 0 とすればよい。

cosine 変換のテストのため解析解が明確な次の二つの問題を取りあげる。

$$\text{問題(1)} \quad \sum_{0 \leq k \leq N} \cos k\theta = \sin\left(N + \frac{1}{2}\right)\theta / \left(2 \sin \frac{\theta}{2}\right)$$

$$\text{問題(2)} \quad \sum_{0 \leq k \leq N} (k+1) \cos k\theta = \left\{ 2(N+1) \sin \frac{\theta}{2} \sin\left(N + \frac{1}{2}\right)\theta + \cos N\theta - 1 \right\} / \left(4 \sin^2 \frac{\theta}{2}\right)$$

右辺の関数を標本点 $\theta_j = \frac{\pi}{N} \left(j + \frac{1}{2}\right)$ で標準化し入力すれば、それぞれのフーリエ係数 ($N/2$ 倍) が出力される。すなわち、

$$X_{j+\frac{1}{2}} = \frac{1}{2} (-1)^j \cot \frac{\pi}{2N} \left(j + \frac{1}{2}\right), \quad 0 \leq j < N$$

を入力すれば、フーリエ係数

$$\frac{2}{N} B_k = 1, \quad 0 \leq k < N$$

また、

$$X_{j+\frac{1}{2}} = \{(-1)^j(N+1)\sin\theta_{j-1}\} / \left(4 \sin^2 \frac{\theta_j}{2}\right)$$

を入力すれば,

$$\frac{2}{N}B_k = k+1$$

を得る。これらを確認するプログラムを以下に示す。

```

C      TEST PROBLEMS OF SUBROUTINE FCOSMS
      DIMENSION X(128),C(127)
      EXTERNAL F
      COMMON L,N,J
      M=7
      N=2**M
      CALL TRIGQS(C,M,ICON)
      DO 30 L=1,2
      DO 10 J=1,N
      P=(FLOAT(J)-0.5)/FLOAT(N)
      X(J)=F(P)
10    CONTINUE
      CALL BITREV(X,M,ICON)
      LX=0
      CALL FCOSMS(X,M,LX,C,M,ICON)
      CT=2.0/FLOAT(N)
      DO 20 I=1,N
      X(I)=X(I)*CT
20    CONTINUE
      WRITE(6,600) L,N,(X(I),I=1,N)
30    CONTINUE
600   FORMAT(///8X,9HPROBLEM (,I1,1H),4X,2HN=,I3/1X/(1H ,4F15.06))
      STOP
      END
      FUNCTION F(P)
      COMMON L,N,J
      SGN=1.0
      IF(MOD(J,2).EQ.0) SGN=-SGN
      GO TO (10,20),L
C      PROBLEM (1)
10    F=0.5*SGN*COTHP(P)
      RETURN
C      PROBLEM (2)
20    F=0.25*(SGN*FLOAT(N+1)*SINH(P+P)-1.0)/SINH(P)**2
      RETURN
      END

```

上で述べた二つの問題に対する中点公式に基づく cosine 変換の計算結果を示す。

k	問題(1)	問題(2)
0	1.000000	0.999998
1	1.000000	1.999999
2	1.000000	2.999998
3	1.000000	3.999999
⋮	⋮	⋮
125	1.000000	125.999998
126	1.000000	126.999999
127	1.000000	127.999999

注：番号 k は出力データの順番である。

つぎに, sine 変換のテスト用問題として,

$$\text{問題(1)} \quad \sum_{1 \leq k \leq N} \sin k\theta = \left\{ \cos \frac{\theta}{2} - \cos \left(N + \frac{1}{2} \right) \theta \right\} / \left(2 \sin \frac{\theta}{2} \right)$$

$$\text{問題(2)} \quad \sum_{1 \leq k \leq N} k \sin k\theta = \left\{ \sin N\theta - 2N \sin \frac{\theta}{2} \cos \left(N + \frac{1}{2} \right) \theta \right\} / \left(4 \sin^2 \frac{\theta}{2} \right)$$

を用いる。右辺の関数を $\theta_j = \frac{\pi}{N} \left(j + \frac{1}{2} \right)$ で標準化して

$$X_{j+\frac{1}{2}} = \frac{1}{2} \left\{ \cot \frac{\theta_j}{2} + (-1)^j \right\}, \quad 0 \leq j < N$$

を入力すれば、フーリエ係数

$$\frac{2}{N} B_k = 1, \quad 1 \leq k < N$$

$$\frac{2}{N} B_N = 2$$

が求められ、

$$X_{j+\frac{1}{2}} = (-1)^j \left\{ 1 + 2N \sin^2 \frac{\theta_j}{2} \right\} / \left(4 \sin^2 \frac{\theta_j}{2} \right)$$

を入力すれば、

$$\frac{2}{N} B_k = k, \quad 1 \leq k < N$$

$$\frac{2}{N} B_N = 2N$$

を得る。

```

C      TEST PROBLEMS OF SUBROUTINE FSINMS
      DIMENSION X(128),C(127)
      EXTERNAL F
      COMMON L,N,J
      M=7
      N=2**M
      CALL TRIGQS(C,M,ICON)
      DO 30 L=1,2
      DO 10 J=1,N
      P=(FLOAT(J)-0.5)/FLOAT(N)
      X(J)=F(P)
10     CONTINUE
      CALL BITREV(X,M,ICON)
      LX=0
      CALL FSINMS(X,M,LX,C,M,ICON)
      CT=2.0/FLOAT(N)
      DO 20 I=1,N
      X(I)=X(I)*CT
20     CONTINUE
      WRITE(6,600)'L,N,(X(I),I=1,N)
30     CONTINUE
600    FORMAT(////8X,9HPROBLEM (,I1,1H),4X,2HN=,I3/1X/(1H ,4F15.06))
      STOP
      END
      FUNCTION F(P)
      COMMON L,N,J
      SGN=1.0
      IF(MOD(J,2).EQ.0) SGN=-SGN
      GO TO (10,20),L
C      PROBLEM (1)

```

```

10 F=0.5*(COTHP(P)+SGN)
   RETURN
C   PROBLEM (2)
20 F=0.25*(1.0/SINHP(P)**2+FLOAT(N+N))*SGN
   RETURN
   END

```

中点公式による高速 sine 変換の計算結果。

k	問題(1)	問題(2)
1	2.000000	255.999999
2	1.000000	126.999999
3	1.000000	125.999999
4	1.000000	124.999999
⋮	⋮	⋮
126	1.000000	3.000000
127	1.000000	2.000000
128	1.000000	1.000000

注：番号 k は出力データの順番である。

参 考 文 献

- 1) J.W. Cooley & J.W. Tukey; "An Algorithm for Machine Calculation of Complex Fourier Series", *Mathematics of Computation*, Vol.19, pp.297-301 (1965).
- 2) 二宮市三; "実高速フーリエ解析 (合成), ビット逆転, その他" ライブラリー・プログラム利用の手引, pp. 95-106, 名古屋大学大型計算機センター (昭53)。
- 3) 鳥居達生; "高速 sine 変換, cosine 変換とその数値積分への応用", *情報処理*, Vol. 15, pp. 670-679 (1974)。
- 4) 牧之内三郎, 鳥居達生; "数値解析", pp. 293-303, オーム社 (昭50)。
- 5) C.W. Clenshaw & A.R. Curtis; "A Method for Numerical Integration on an Automatic Computer", *Numerische Mathematik*, Vol.2, pp.197-205 (1960).
- 6) 二宮市三; "引数 $\frac{\pi}{2}x$ に対する三角関数", ライブラリー・プログラム利用の手引, p. 12. 名古屋大学大型計算機センター (昭53)。
- 7) FACOM FORTRAN SSL II 使用手引書, p. 247, 富士通 (昭53)。
- 8) 鳥居達生; "フーリエ変換サブルーチンパッケージの作成", *名古屋大学大型計算機センターニュース*, Vol. 10, No. 1, p. 24 (1979), Vol.10, No. 2, p. 138 (1979)。

TRIGQP / TRIGQD

Table of Trigonometric Function Arranged in Bit Reverse Order

2進逆順に並べられた三角関数表

作成	鳥居 達生 1978年12月
形式	サブルーチン 言語; FORTRAN サイズ; 51, 52行

(1) 概要

高速 sine 変換, cosine 変換及び関数のチェビシェフ級数展開に必要な三角関数表を作成する。

n ビットの整数

$$j = j_1 2^0 + j_2 2^1 + \dots + j_n 2^{n-1}, \quad j_i \in \{0, 1\}$$

に対し, 1 より小さい n ビットの小数

$$j^* = j_1 2^{-1} + j_2 2^{-2} + \dots + j_n 2^{-n}$$

を定義し, 複素三角関数

$$e^{i\frac{\pi}{4} j^*}, \quad j = 0, 1, 2, \dots$$

を計算する。

(2) 使用法

CALL TRIGQP (C, M, ICON)

CALL TRIGQD (C, M, ICON)

引数	型と種類	属性	内容
C	実数型 1次元配列	出力	$C(1) = \cos \frac{\pi}{4}$ $C(2j) = \cos \frac{\pi}{4} j^*$ $C(2j+1) = \sin \frac{\pi}{4} j^*$ $1 \leq j < 2^{M-1}$
M	整数型	入力	配列Cの大きさ $\geq 2^M - 1$ 。 $M \geq 1$
ICON	整数型	出力	ICON = 0; 正常。 ICON = 30000; パラメータ・エラー。

(3) 性能

三角関数表のデータ数が $2^M - 1$ のとき, 必要な演算回数は, 平方根が M 回, 乗算が 2^M 回である。

(4) 計算法

簡単のため $W_j = e^{i\frac{\pi}{4} j^*}$ とおけば, これは次の漸化式に従う。

$$W_0 = e^{i\frac{\pi}{4}}, W_1 = e^{i\frac{\pi}{8}} \text{ 初期値}$$

$$W_{2^l} = (W_{2^{l-1}})^{1/2}, \text{ 平方根の虚数部は正。}$$

$$W_{2^l + j} = W_{2^{l-1} + j} \cdot \bar{W}_{2^l}, \quad 1 \leq j < 2^{l-1}$$

$$W_{2^l + 2^{l-1} + j} = W_{2^{l-1} + j} \cdot W_{2^l}, \quad 0 \leq j < 2^{l-1}$$

$$l = 1, 2, \dots$$

FCOSTS / D

FSINTS / D

Fast Fourier Cosine Transform Based on The Trapezoidal Rule (FCOSTS/D)

台形公式に基づく高速 cosine 変換 (FCOSTS/D)

Fast Fourier Sine Transform Based on The Trapezoidal Rule (FSINTS/D)

台形公式に基づく高速 sine 変換 (FSINTS/D)

作 成	鳥居 達生 1978年7月
形 式	サブルーチン 言語 ; FORTRAN サイズ ; 64, 65, 33, 34行

(1) 概 要

周期 2π の関数 $X(t)$ の半周期を N 等分して得られる $N+1$ 個の標本を

$$X_j = X\left(\frac{\pi}{N}j\right), \quad 0 \leq j \leq N, \quad N = 2^n$$

とする。 $X(t)$ が偶関数のとき、

$$C_k = \sum_{0 \leq j \leq N} X_j \cos \frac{\pi}{N}kj, \quad 0 \leq k \leq N$$

ここで、 Σ は初項と末項を $1/2$ 倍して和をとることを意味する。

$X(t)$ が奇関数のとき、実質上の標本数は $N-1$ であって、

$$C_k = \sum_{0 < j < N} X_j \sin \frac{\pi}{N}kj, \quad 0 < k < N$$

を求める。逆変換も同一のプログラムで行うことができる。

(2) 使用法

CALL FCOSTS/D (X, MX, W, MW, ICON)

CALL FSINTS/D (X, MX, W, MW, ICON)

引 数	型 と 種 類	属 性	内 容
X	実 数 型 1 次 元 配 列	入 出 力	FCOST/D の場合 $X(j+1) = X_j, 0 \leq j \leq 2^{MX}$ を入力すれば、出力として $X(j+1) = C_j$ を得る。 FSINT/D の場合 $X(1) = X_0 = 0, X(j+1) = X_j, 1 \leq j < 2^{MX}$ を入力すれば、 $X(1) = C_0 = 0, X(j+1) = C_j$ が出力される。
MX	整 数 型	入 力	保存される。 $MX \geq 1$
W MW	実 数 型 1 次 元 配 列 整 数 型	入 力	TRIGQP/TRIGQD (W, MW, ICON) を用いて、W 上に三角関数表をあらかじめ求めておく。MW は三角関数表のデータの個数 $2^{MW}-1$ を指定する。配列 W の大きさ $\geq 2^{MW}-1$ 。保存される。 $MW \geq MX-1$
ICON	整 数 型	出 力	ICON = 0 : 正常。 ICON = 30000 : パラメータ・エラー。

(3) 性能

中点公式に基づく高速 cosine (sine)変換と性能は、さして変わらない。

(4) 使用例

偶関数 $\sin(N+\frac{1}{2})\theta/2\sin\frac{\theta}{2}$ を点 $\theta_j = \frac{\pi}{N}j$ で標本化し、 $N+1$ 個の標本

$$X_j = \begin{cases} N-\frac{1}{2}, & j = 0 \\ (-1)^j/2, & 1 \leq j \leq N \end{cases}$$

を入力すれば、正解は

$$\frac{2}{N} C_k = \begin{cases} 1, & 0 \leq k < N \\ 2, & k = N \end{cases}$$

である。また、 $\{2(N+1)\sin\frac{\theta}{2}\sin(N+\frac{1}{2})\theta+\cos N\theta-1\}/4\sin^2\frac{\theta}{2}$ を標本化して、

$$X_j = \begin{cases} (N^2+3N+1)/2, & j = 0 \\ (N+1)/2, & j \text{ は偶数}, 0 < j \leq N \\ -(N+1+\operatorname{cosec}^2\frac{\theta_j}{2})/2, & j \text{ は奇数}, 1 \leq j < N \end{cases}$$

を入力すれば、

$$\frac{2}{N} C_k = \begin{cases} k+1, & 0 \leq k < N \\ 2(N+1), & k = N \end{cases}$$

を得る。台形公式に基づく cosine 変換によって、これらのことを確認する。

```

C      TEST PROBLEMS OF SUBROUTINE FCOSTS
      DIMENSION X(129),W(63)
      EXTERNAL F
      COMMON L,AN,J
      M=7
      MW=6
      N=2**M+1
      AN=FLOAT(N-1)
      CALL TRIGQS(W,MW,ICON)
      DO 30 L=1,2
      DO 10 J=1,N
      X(J)=F(FLOAT(J-1)/AN)
10 CONTINUE
      CALL FCOSTS(X,M,W,MW,ICON)
      CT=2.0/AN
      DO 20 I=1,N
      X(I)=X(I)*CT
20 CONTINUE
      WRITE(6,600) L,N,(X(I),I=1,N)
30 CONTINUE
600 FORMAT(///8X,9HPROBLEM (,I1,1H),4X,2HN=,I3/1X/(1H ,4F15.06))
      STOP
      END
      FUNCTION F(P)
      COMMON L,AN,J
      SGN=1.0
      IF(MOD(J,2).EQ.0) SGN=-SGN
      GO TO (10,20),L

```

```

C   PROBLEM (1)
10  F=AN+0.5
    IF(J.EQ.1) RETURN
    F=SGN*0.5
    RETURN
C   PROBLEM (2)
20  IF(J.EQ.1) GO TO 21
    IF(SGN.LT.0.0) GO TO 22
    F=(AN+1.0)*0.5
    RETURN
21  F=(AN*AN+3.0*AN+1.0)*0.5
    RETURN
22  F=-(AN+1.0)*0.5-0.5/SINHP(P)**2
    RETURN
    END

```

台形公式に基づく高速 cosine 変換の数値例

k	問題(1)	問題(2)
0	1.000000	1.000000
1	1.000000	2.000000
2	1.000000	3.000000
3	1.000000	4.000000
⋮	⋮	⋮
125	1.000000	125.000000
126	1.000000	125.999999
127	1.000000	127.999999
128	2.000000	257.999999

注：番号 k は出力データの順番である。

次に sine 変換の使用例を示す。奇関数 $\{\cos \frac{\theta}{2} - \cos(N + \frac{1}{2})\theta\} / 2 \sin \frac{\theta}{2}$ を、点 $\theta_j = \frac{\pi}{N} j$, $1 \leq j < N$ で標本化し、 $N-1$ 個のデータ

$$X_j = \begin{cases} 0, & j \text{ は偶数} \\ \cot \frac{\theta_j}{2}, & j \text{ は奇数} \end{cases}$$

を入力すれば、このフーリエ係数は、すべて 1 である。すなわち、

$$\frac{2}{N} C_k = 1, \quad 1 \leq k \leq N-1$$

また、 $\{(N+1)\sin N\theta - N\sin(N+1)\theta\} / 4 \sin^2 \frac{\theta}{2}$ に対する $N-1$ 個の標本

$$X_j = \frac{1}{2} (-1)^{j-1} N \cot \frac{\theta_j}{2}, \quad 1 \leq j \leq N-1$$

を入力すれば、

$$\frac{2}{N} C_k = k, \quad 1 \leq k \leq N-1$$

を得る。

```

C   TEST PROBLEMS OF SUBROUTINE FSINTS
    DIMENSION X(128),W(63)
    EXTERNAL F
    COMMON L,AN,J
    M=7
    MW=6
    N=2**M-1
    AN=FLOAT(N+1)
    CALL TRIGGS(W,MW,ICON)
    DO 30 L=1,2
    X(L)=0.0
    DO 10 J=1,N
    X(J+1)=F(FLDAT(J)/AN)
10  CONTINUE
    CALL FSINTS(X,M,W,MW,ICON)
    CT=2.0/AN
    DO 20 I=1,N
    X(I+1)=X(I+1)*CT
20  CONTINUE
    WRITE(6,600) L,N,(X(I+1),I=1,N)
30  CONTINUE
600 FORMAT(////8X,9HPROBLEM (,I1,1H),4X,2HN=,I3/1X/(1H ,4F15.06))
    STOP
    END
    FUNCTION F(P)
    COMMON L,AN,J
    SGN=1.0
    IF(MOD(J,2).NE.0) SGN=-SGN
    GO TO (10,20),L
C   PROBLEM (1)
10  F=0.0
    IF(SGN.GT.0.0) RETURN
    F=COTHP(P)
    RETURN
C   PROBLEM (2)
20  F=0.5*SGN*COTHP(P)*(-AN)
    RETURN
    END

```

台形公式に基づく高速 sine 変換の数値例

k	問題(1)	問題(2)
1	1.000000	1.000000
2	1.000000	2.000000
3	1.000000	3.000000
4	1.000000	4.000000
⋮	⋮	⋮
125	1.000000	124.999996
126	1.000000	125.999998
127	1.000000	126.999998

注：番号 k は出力データの順番である。

FCOSCS / D
 FCOSOS / D
 FSINOS / D

Fourier Cosine Series of Even Function Defined on The Closed Interval $(0, \pi)$
 (FCOSCS/D)

閉区間 $(0, \pi)$ で与えられた偶関数の cosine 級数展開 (FCOSCS/D)

Fourier Cosine Series of Even Function Defined on The Open Interval $(0, \pi)$
 (FCOSOS/D)

開区間 $(0, \pi)$ で与えられた偶関数の cosine 級数展開 (FCOSOS/D)

Fourier Sine Series of Odd Function Defined on The Open Interval $(0, \pi)$
 (FSINOS/D)

開区間 $(0, \pi)$ で与えられた奇関数の sine 級数展開 (FSINOS/D)

作成	鳥居 達生	1978年12月
形式	サブルーチン	言語 ; FORTRAN サイズ ; 112, 114, 115, 117, 91, 93行

(1) 概要

周期 2π の関数 $f(t)$ が、偶関数あるいは奇関数ならば、半周期 $[0, \pi]$ だけで $f(t)$ を与えればよい。 $f(t)$ を cosine 級数に展開する場合区間の端点を標本点に用いる方法と用いない方法に分かれる。前者が、FCOSCS/D であり、後者が FCOSOS/D である。sine 級数展開の場合、端点は標本点に用いない。

関数 $f(t)$ を入力すれば、要求精度に応じて自動的に展開項数を決定し、フーリエ係数を出力する。中点公式に基づく高速 cosine(sine)変換が計算法の基礎となっているので、能率的である。

(2) 使用法

CALL FCOSCS/D (F, EPSA, EPSR, NMIN, NMAX, A, N, ERR, ICON)

CALL FCOSOS/D (F, EPSA, EPSR, NMIN, NMAX, A, N, ERR, ICON)

CALL FSINOS/D (F, EPSA, EPSR, NMIN, NMAX, A, N, ERR, ICON)

引数	型と種類	属性	内容
F	実数型 関数副プログラム	入力	1変数の周期関数(偶関数あるいは奇関数)を関数副プログラムとして使用者が定義する。この関数の定義域は、FCOSCS/Dの場合、閉区間 $[0, \pi]$ であり、FCOSOS/D、FSINOS/Dの場合、開区間 $(0, \pi)$ でのよい。
EPSA EPSR	実数型	入力	求めるフーリエ級数の誤差限界。EPSA ≥ 0 、EPSR ≥ 0 は、それぞれ絶対誤差、相対誤差による要求精度。保存される。
NMIN NMAX	整数型	入力	展開項数の下限、上限。保存される。 FCOSCS/Dの場合 $0 \leq NMIN \leq NMAX \leq 1025$ 。 FCOSOS/D及びFSINOS/Dの場合 $0 \leq NMIN \leq NMAX \leq 1023$ 。

引 数	型 と 種 類	属 性	内 容
A N	実 数 型 1 次 元 配 列 整 数 型	出 力	配列Aの大きさ $\geq NMAX$ 。A上にN個のフーリエ係数が番号順に格納される。使用した標本数もNである。Nは FCOSC/Dの場合 2^n+1 ， FCOSO/D及びFSINO/Dの場合 2^n-1 の形の整数値をとる。 標本数の制限はNMINよりNMAXが優先する。
ERR	実 数 型	出 力	求めたフーリエ級数の絶対誤差（推定値）。
ICON	整 数 型	出 力	ICON = 0 ならば次の意味で正常。 入力関数 $f(t)$ のN次のフーリエ級数を $P_N(t)$ とすれば、 $ f(t)-P_N(t) \leq \max\{EPSA, EPSR*\ f\ \}$ 。 ただし $\ f\ = \max_{0 \leq j \leq N} f(\frac{\pi}{N}j) $ 。 ICON = 10000 : 要求精度がきびしすぎるため $P_N(t)$ は上の条件は満たさないが、計算誤差の限界内にある。正常とみなしてよい。 ICON = 20000 : 異常。N $\leq NMAX$ の下で要求精度が得られない。 ICON = 30000 : パラメータ・エラー。

(3) 性能

入力関数 $f(t)$ を標本化する時間を除けば、入力がデータの高速 cosine(sine) 変換と同じ計算時間である。

(4) 計算法

逐次近似方式の台形公式に基づく高速 cosine(sine) 変換法である。ただし、FCOSO は、区間 $(0, \pi)$ の端点を標本点に用いないように修正されている。

求めたフーリエ係数の誤差は、末尾2項の係数の絶対値和で推定している。丸め誤差の伝播誤差は計算機の機械精度の最小単位を u として、

$$16u \|f\|$$

で評価した。プログラムの中では u として FUNCTION 副プログラム AMACH を引用している。

本サブルーチンの内部に三角関数表のため大きさ511の実数型1次元配列と関数 $f(t)$ を2進逆順に標本化するために、大きさ256の整数型1次元配列をもっている。本ルーチンが呼ばれる場合、これらの定数表は初回だけ計算される。定数表の大きさを2倍にすれば標本数の上限も2倍にゆるめることができる。

(5) 使用例

1. 閉区間 $[0, \pi]$ で与えられた偶関数の cosine 級数展開の使用例
cosine 関数の母関数

$$f(\theta) = \frac{1-t^2}{1-2t \cos \theta + t^2}$$

を展開し、

$$= 1 + 2 \sum_{n=1}^{\infty} t^n \cos n\theta$$

が得られることを確認する。 $t = 0.5$ としたときのプログラムを示す。

```

C      EXAMPLE FOR SUBROUTINE FCOSCS
      DIMENSION A(257)
      EXTERNAL F
      COMMON T
      T=0.5
      EPSA=1.0E-5
      EPSR=EPSA
      NMAX=257
      NMIN=0
      CALL FCOSCS(F, EPSA, EPSR, NMIN, NMAX, A, N, ERR, ICON)
      WRITE(6,600) N, ERR, ICON, (A(I), I=1, N)
600   FORMAT(1H0,4X,2HN=,I3,5X,4HERR=,E10.3,5X,5HICON=,I5/1H0,4X,
      *7HARRAY A/(1H ,4F15.06))
      STOP
      END
      FUNCTION F(P)
      COMMON T
      F=(1.0-T*T)/(1.0-2.0*T*COS(P)+T*T)
      RETURN
      END

```

2. 開区間 $(0, \pi)$ で与えられた奇関数の sine 級数展開の使用例

$$\frac{\sin \theta}{1 - 2t \cos \theta + t^2} = \sum_{k=1}^{\infty} t^{k-1} \sin k\theta$$

```

C      EXAMPLE FOR SUBROUTINE FSINOS
      DIMENSION A(257)
      EXTERNAL F
      COMMON T
      T=0.5
      EPSA=1.0E-5
      EPSR=EPSA
      NMAX=257
      NMIN=0
      CALL FSINOS(F, EPSA, EPSR, NMIN, NMAX, A, N, ERR, ICON)
      WRITE(6,600) N, ERR, ICON, (A(I), I=1, N)
600   FORMAT(1H0,4X,2HN=,I3,5X,4HERR=,E10.3,5X,5HICON=,I5/1H0,4X,
      *7HARRAY A/(1H ,4F15.06))
      STOP
      END
      FUNCTION F(P)
      COMMON T
      F=SIN(P)/(1.0-2.0*T*COS(P)+T*T)
      RETURN
      END

```

3. 開区間 $(0, \pi)$ で与えられた偶関数の cosine 級数展開の使用例

区間の端点を標本点に使用できない場合、本ルーチンを使用する。偶関数

$$f(\theta) = \frac{\alpha}{2} \cdot \frac{1 + \tan^2 \frac{\theta}{2}}{\alpha^2 + \tan^2 \frac{\theta}{2}}$$

を cosine 級数に展開すれば、

$$= \sum_{k=0}^{\infty} \left(\frac{1-\alpha}{1+\alpha} \right)^k \cos k\theta$$

となる。 $\alpha = 1/3$ とおき、 $f(\theta)$ を $(0, \pi)$ で展開するプログラムを次に示す。

```

C      EXAMPLE FOR SUBROUTINE FCOSOS
      DIMENSION A(257)
      EXTERNAL F
      COMMON ALPHA
      ALPHA=1.0/3.0
      EPSA=1.0E-5
      EPSR=EPSA
      NMAX=257
      NMIN=0
      CALL FCOSOS(F, EPSA, EPSR, NMIN, NMAX, A, N, ERR, ICON)
      WRITE(6,600) N, ERR, ICON, (A(I), I=1, N)
600   FORMAT(1H0,4X,2HN=,I3,5X,4HERR=,E10.3,5X,5HICON=,I5/1H0,4X,
*7HARRAY A/(1H ,4F15.06))
      STOP
      END
      FUNCTION F(P)
      COMMON ALPHA
      Q=TAN(P*0.5)**2
      F=ALPHA*0.5*(1.0+Q)/(ALPHA**2+Q)
      RETURN
      END

```

以上まとめて計算結果を示す。

k	問題 5. 1	問題 5. 2	問題 5. 3	k	問題 5. 1	問題 5. 2	問題 5. 3
0	2.000000	—	1.000000	:	:	:	:
1	1.000000	1.000000	0.500000	30	0.000000	0.000000	0.000000
2	0.500000	0.500000	0.250000	31	0.000000	0.000000	—
:	:	:	:	32	0.000000	—	—
14	0.000121	0.000122	0.000061	誤差の 推定値	0.715 E - 06	0.317 E - 06	0.351 E - 06
15	0.000061	0.000061	0.000031				
16	0.000031	0.000031	0.000015				

注：番号 k は出力データの順番である。

FCHB1S / D

FCHB2S / D

FCHB3S / D

FCHBOS / D

Fourier Expansion of Functions by Chebyshev Polynomials of First Kind
(FCHB1S/D)

第1種チェビシェフ多項式による関数のフーリエ展開 (FCHB1S/D)

Fourier Expansion of Functions by Chebyshev Polynomials of Second Kind
(FCHB2S/D)

第2種チェビシェフ多項式による関数のフーリエ展開 (FCHB2S/D)

Fourier Expansion of Functions by Shifted Chebyshev Polynomials
(FCHB3S/D)

ずらしチェビシェフ多項式による関数のフーリエ展開 (FCHB3S/D)

Fourier Expansion of Functions on The Open Interval by First Chebyshev Polynomials
(FCHBOS/D)

第1種チェビシェフ多項式による開区間上で与えられた関数のフーリエ展開 (FCHBOS/D)

作 成	鳥居 達生 1978年7月
形 式	サブルーチン 言語; FORTRAN サイズ; 98, 99, 78, 79, 91, 92, 101, 103 行

(1) 概 要

有限区間(開区間あるいは閉区間)で与えられた関数 $f(t)$ を要求精度 ϵ に基づきチェビシェフ級数展開する。計算法の基本は、周期関数の cosine 級数展開 (sine 級数) と同じである。

閉区間 $[-1, 1]$ で滑らかな関数 $f(t)$ を第1種チェビシェフ多項式で展開するのが FCHB1 である。

$$f(t) \approx \sum_{0 \leq k \leq N} C_k T_k(t) = \sum_{0 \leq k \leq N} C_k \cos k\theta$$

ここで、 $t = \cos \theta$ であって次数 $N = N(\epsilon)$ は2のべきの値をとる。

開区間 $(-1, 1)$ で滑らかな関数 $f(t)$ を第2種チェビシェフ多項式で展開するのが FCHB2 である。

$$f(t) \approx \sum_{0 \leq k \leq N-2} C_k U_k(t) = \sum_{0 < k < N} C_{k-1} \frac{\sin k\theta}{\sin \theta}$$

閉区間 $[0, 1]$ で滑らかな関数は、ずらしチェビシェフ多項式系で展開される。

$$f(t) \approx \sum_{0 \leq k \leq N} C_k T_k^*(t) = \sum_{0 \leq k \leq N} C_k \cos k\theta$$

ただし、 $t = \cos^2 \frac{\theta}{2}$

与えられた区間の両端を標本点にとることができない関数 $f(t)$ を第1種チェビシェフ多項式系で展開したい場合、FCHBO を用いればよい。

$$f(t) = \sum_{0 \leq k \leq N-2} C_k T_k(t)$$

(2) 使用法

CALL FCHB1S/D (F, EPSA, EPSR, NMIN, NMAX, A, N, ERR, ICON)

CALL FCHB2S/D (F, EPSA, EPSR, NMIN, NMAX, A, N, ERR, ICON)

CALL FCHB3S/D (F, EPSA, EPSR, NMIN, NMAX, A, N, ERR, ICON)

CALL FCHBOS/D (F, EPSA, EPSR, NMIN, NMAX, A, N, ERR, ICON)

引 数	型 と 種 類	属 性	内 容
F	実 数 型 関数副プログラム	入 力	1変数の関数を関数副プログラムとして使用者が定義する。FCHB1S/D, FCHB2S/D, FCHB3S/D, FCHBOS/Dに応じて、関数の定義域は、 $[-1, 1]$, $(-1, 1)$, $[0, 1]$, $(-1, 1)$ でなければならない。
EPSA EPSR	実 数 型	入 力	求めるチェビシェフ級数の誤差限界。EPSA, EPSR ≥ 0 は、それぞれ絶対誤差、相対誤差による要求精度。保存される。
NMIN NMAX	整 数 型	入 力	標本数の下限、上限。保存される。 FCHB1S/D, FCHB3S/Dの場合 $0 \leq NMIN \leq NMAX \leq 1025$ FCHB2S/D, FCHBOS/Dの場合 $0 \leq NMIN \leq NMAX \leq 1023$
A N	実 数 型 1 次 元 配 列 整 数 型	出 力	配列Aの大きさ $\geq NMAX$ 。配列A上にN個のチェビシェフ展開係数が番号順に格納される。Nは、FCHB1S/D, FCHB3S/Dの場合 2^n+1 , FCHB2S/D, FCHBOS/Dの場合 2^n-1 の型の正の整数値をとる。
ERR	実 数 型	出 力	求めたチェビシェフ級数の誤差の上限 (注参照)。
ICON	整 数 型	出 力	ICON = 0 ならば次の意味で正常。 誤差 $\leq \max\{EPSA, EPSR * \ f\ \}$ ICON = 10000 : 要求精度がきびしすぎるため、上の条件は満たされないが、打ち切り誤差は丸め誤差の水準 (計算誤差の限界) にある。 ICON = 20000 : 異常。標本数が、その上限 NMAX に達しても、打ち切り誤差が要求精度あるいは丸め誤差の水準まで低下しない。 ICON = 30000 : パラメータ・エラー。

注：FCHB1S/D, FCHB3S/D, FCHBOS/D におけるノルムの定義は $\|f\|_\infty = \max_j |f(x_j)|$, x_j は標本点

FCHB2S/D においては、 $\|f\|_1 = \sum |C_k|$, C_k は $f(t)$ の第2種チェビシェフ展開係数である。それぞれの打ち切り誤差は、このノルムに従って測られる。

(3) 性能

関数 $f(t)$ を標本化する時間を除けば、台形公式に基づく高速 cosin (sine) 変換と同じ演算回数である。

(4) 計算法

閉区間 $[0, \pi]$ で与えられた偶関数 $f(\cos \theta)$, $f(\cos^2 \frac{\theta}{2})$ に対する高速 cosine 変換が FCHB1S, FCHB3S にほかならない。両端を標本点に採用しない $f(\cos \theta)$ の cosine 変換は FCHBOS に対応する。

奇関数 $f(\cos \theta) \sin \theta$ に対する高速 sine 変換が FCHB2S である。求めたチェビシェフ級数の誤差は、

末尾2項の係数の絶対値和で推定している。

各サブルーチンは、三角関数表のために1次元配列をもっている(FCHBOS, FCHB1S, FCHB2S は511語, FCHB3S は1023語)。これは cosine(sine) 変換と標本点に共用される。各サブルーチンが初めて呼ばれたときにだけ、これらの定数表は計算され、以後保存される。

(5) 使用例

第1種チェビシェフ多項式の母関数

$$\frac{1-t^2}{1-2tx+t^2} = 2 \sum_{k=0}^{\infty} t^k T_k(x), \quad 0 < t < 1$$

を与えられた要求精度の下で第1種チェビシェフ多項式系でフーリエ展開する。

以下は、パラメータ t を 1/4, 2/4, 3/4 と動かした場合の FCHB1S の使用例である。要求精度は絶対誤差で 10^{-5} である。標本数の下限, 上限等については以下のプログラムに記されている。

```

C      TEST FOR SUBROUTINE FCHB1S.
      DIMENSION A(257)
      EXTERNAL F
      COMMON T
      AEPS=1.0E-05
      REPS=0.0
      NMIN=0
      NMAX=257
      T=0.25
      H=0.25
1     CONTINUE
      CALL FCHB1S (F,AEPS,REPS,NMIN,NMAX,A,N,ERR,ICON)
      WRITE(6,600) N,ERR,ICON,T,(A(I),I=1,N)
      T=T+H
      IF(T.LT.1.0) GO TO 1
600   FORMAT(1H0,4X,2HN=,I3,5X,4HERR=,E10.3,5X,5HICON=,I5/1H0,4X,
      *7HARRAY A,5X,2HT=,F5.2/(1H ,4F15.06))
      STOP
      END
      FUNCTION F(P)
C      GENERATING FUNCTION OF CHEBYSHEV POLYNOMIALS OF FIRST KIND.
      COMMON T
      F=(1.0-T*T)/(1.0-2.0*T*P+T*T)
      RETURN
      END

```

DATE 82.02.23 TIME 12:41:08

NAME FCOSOS

第1種チェビシェフ多項式の母関数の展開

k	$t = 1/4$	$t = 1/2$	$t = 3/4$
0	2.000000	2.000000	2.000000
1	0.500000	1.000000	1.500000
⋮	⋮	⋮	⋮
8	0.000031	0.007813	0.200226
9	0.000008	0.003906	0.150169
⋮	⋮	⋮	⋮
16	0.000000	0.000031	0.020045

k	$t = 1/4$	$t = 1/2$	$t = 3/4$
17		0.000015	0.015034
⋮		⋮	⋮
32		0.000000	0.000201
33		0.000000	0.000151
⋮			⋮
64			0.000000
65			0.000000
項 数	17	33	65
誤 差 推定値	0.397 E - 06	0.715 E - 06	0.167 E - 05

注：番号 k は出力データの順番である。

第2種チェビシェフ多項式の母関数

$$\frac{1}{1-2tx+t^2} = \sum_{k=0}^{\infty} t^k U_k(x)$$

を用いてサブルーチン FCHB2S をテストする。

```

C      TEST FOR SUBROUTINE FCHB2S
      DIMENSION A(255)
      EXTERNAL F
      COMMON T
      AEPS=1.0E-05
      REPS=0.0
      NMAX=255
      NMIN=0
      T=0.25
      H=0.25
1     CONTINUE
      CALL FCHB2S(F,AEPS,REPS,NMIN,NMAX,A,N,ERR,ICON)
      WRITE(6,600) N,ERR,ICON,T,(A(I),I=1,N)
      T=T+H
      IF(T.LT.1.0) GO TO 1
600  FORMAT(1H0,4X,2HN=,13,5X,4HERR=,E10.3,5X,5HICON=,15/1H0,4X,
      *7HARRAY A,5X,2HT=,F5.2/(1H,4F15.06))
      STOP
      END
      FUNCTION F(P)
C      GENERATING FUNCTION OF CHEBYSHEV POLYNOMIALS OF SECOND KIND.
      COMMON T
      F=1.0/(1.0-2.0*T*P+T*T)
      RETURN
      END

```

第2種チェビシェフ多項式の母関数の展開

k	$t = 1/4$	$t = 1/2$	$t = 3/4$
0	1.000000	1.000000	1.000000
1	0.250000	0.500000	0.750000
2	0.062500	0.250000	0.562500
⋮	⋮	⋮	⋮
15		0.000031	0.013363
16		0.000015	0.010023
17		0.000008	0.007517
⋮		⋮	⋮
31			0.000134
32			0.000100

k	$t = 1/4$	$t = 1/2$	$t = 3/4$
33			0.000075
⋮			⋮
62			0.000000
項 数	15	31	63
誤差の 推定値	0.159 E - 06	0.238 E - 06	0.477 E - 06

注：番号 k は出力データの順番である。

ずらしチェビシェフ多項式系を用いて、次の二つの関数を展開する。

$$f(x) = \frac{1-t^2}{1-2t(2x-1)+t^2}, \quad 0 \leq x \leq 1$$

$$= -2 \sum_{k=0}^{\infty} t^{-k} T_k^*(x), \quad t > 1$$

$$g(x) = \frac{1}{1+x^2}, \quad -t \leq x \leq t$$

$$= \frac{2t^{-1}}{\sqrt{1+t^{-2}}} \sum_{k=0}^{\infty} (-1)^k (t^{-1} + \sqrt{1+t^{-2}})^{-2k} T_k^*\left(\frac{x}{t}\right)^2$$

関数 $g(x)$ の変域は $[-t, t]$ であって、かつ偶関数であるから変数変換 $y = (x/t)^2$ をほどこした。パラメータ t は、2, 4, 8 と動かしている。要求精度は絶対誤差で 10^{-5} である。

```

C      TEST PROBLEMS OF SUBROUTINE FCHB3S
      DIMENSION A(257)
      EXTERNAL F
      COMMON T,L
      AEPS=1.0E-05
      REPS=0.0
      NMIN=0
      NMAX=257
      DO 10 L=1,2
      T=2.0
      1 CONTINUE
      CALL FCHB3S (F,AEPS,REPS,NMIN,NMAX,A,N,ERR,ICON)
      WRITE(6,601) L
601  FORMAT(1H0,4X,9HPROBLEM (,I1,1H),)
      WRITE(6,600) N,ERR,ICON,T,(A(I),I=1,N)
600  FORMAT(1H0,4X,2HN=,I3,5X,4HERR=,E10.3,5X,5HICON=,I5/1H0,4X,
      *7HARRAY A,5X,2HT=,F5.2/(1H ,4F15.06))
      T=T+T
      IF(T.LE.8.0) GO TO 1
10 CONTINUE
      STOP
      END
      FUNCTION F(P)
      COMMON T,L
      GO TO (10,20),L
C      PROBLEM (1)
10 CONTINUE
C      GENERATING FUNCTION OF SHIFTED CHEBYSHEV POLYNOMIALS.
      Q=P+P-1.0
      F=(1.0-T*T)/(1.0-2.0*T*Q+T*T)
      RETURN
C      PROBLEM (2)
20 CONTINUE
C      APPLY THE VARIABLE TRANSFORMATION.
      Q=T*SQRT(P)
      F=1.0/(1.0+Q*Q)
      RETURN
      END

```

関数 $1/(1+x^2)$ を $\{T_k^*((\frac{x}{t})^2)\}$ で展開した結果を示す。

k	$t = 2$	$t = 4$	$t = 8$
0	0.894427	0.485071	0.248069
1	-0.341641	-0.295705	-0.193322
2	0.130495	0.180265	0.150656
⋮	⋮	⋮	⋮
15	-0.000001	-0.000289	-0.005891
16	0.000000	0.000176	0.004591
17		0.000108	-0.003578
⋮		⋮	⋮
31		0.000000	-0.000109
32		0.000000	0.000085
33		0.000000	-0.000066
⋮			⋮
63			0.000000
64			0.000000
項 数	17	33	65
誤差の 推定値	0.905 E - 06	0.272 E - 06	0.238 E - 06

注：番号 k は出力データの順番である。

開区間で定義された関数を第1種チェビシェフ級数に展開するには、サブルーチン FCHBOS を使えばよい。端点を標本点に使用しないので、使用するものと比べると一般に精度はわるい。FCHB1S と比較のため、母関数

$$f(x) = \frac{1-t^2}{1-2xt+t^2} = 2 \sum' t^k T_k(x)$$

を区間 $[-1, 1]$ で展開する。また、 $(0, \infty)$ で定義される関数

$$g(x) = \frac{(1-t^2)(1+x)}{2\{(1-t^2)+(1+t)^2x\}} = \sum_{k=0}^{\infty} t^k T_k\left(\frac{1-x}{1+x}\right)$$

を変数変換 $y = (1-x)/(1+x)$ によって $[-1, 1]$ にうつしチェビシェフ級数展開する。

パラメータ t を $1/4, 2/4, 3/4$ と動かした場合の計算例である。

```

C      TEST PROBLEMS OF SUBROUTINE FCHBOS
      DIMENSION A(255)
      COMMON T,L
      EXTERNAL F
      AEPS=1.0E-05
      REPS=0.0
      NMIN=0
      NMAX=255
      DO 10 L=1,2
      T=0.25
      H=0.25
      1 CONTINUE
      CALL FCHBOS(F,EPSA,EPSR,NMIN,NMAX,A,N,ERR,ICON)
      WRITE(6,601) L
      WRITE(6,600) N,ERR,ICON,T,(A(I),I=1,N)
601  FORMAT(1H0,4X,9HPROBLEM (,I1,1H),)
600  FORMAT(1H0,4X,2HN=,I3,5X,4HERR=,E10.3,5X,5HICON=,I5/1H0,4X,
      *7HARRAY A,5X,2HT=,F5.2/(1H ,4F15.06))

```

```

T=T+H
IF(T.LT.1.0) GO TO 1
10 CONTINUE
STOP
END
FUNCTION F(P)
COMMON T,L
GO TO (10,20),L
10 CONTINUE
PROBLEM (1)
C GENERATING FUNCTION OF CHEBYSHEV POLYNOMIALS OF FIRST KIND.
F=(1.0-T*T)/(1.0-2.0*T*P+T*T)
RETURN
C PROBLEM (2)
20 CONTINUE
C APPLY THE VARIABLE TRANSFORMATION
Q=(1.0-P)/(1.0+P)
F=(1.0-T*T)*(1.0+Q)*0.5/((1.0-T)**2+(1.0+T)**2*Q)
RETURN
END

```

有理式 $g(x)$ を $\left\{ T_k \left(\frac{1-x}{1+x} \right) \right\}$ で展開した例である。

k	$t = 1/4$	$t = 1/2$	$t = 3/4$
0	1.000000	1.000000	1.000000
1	0.250000	0.500000	0.750000
2	0.062500	0.250000	0.562500
⋮	⋮	⋮	⋮
15		0.000031	0.013363
16		0.000015	0.010023
17		0.000008	0.007517
⋮		⋮	⋮
31			0.000134
32			0.000100
33			0.000075
⋮			⋮
62			0.000000
項数	15	31	63
誤差の 推定値	0.195 E - 06	0.351 E - 06	0.811 E - 06

注：番号 k は出力データの順番である。

AQDCCS / D

AQDCOS / D

Automatic Quadrature of Closed Type by Clenshaw-Curtis Method (AQDCCS/D)

クレンショウ・カーチス法による自動積分（閉じた積分公式）(AQDCCS/D)

Automatic Quadrature of Open Type by Clenshaw-Curtis Method (AQDCOS/D)

クレンショウ・カーチス法による自動積分（開いた積分公式）(AQDCOS/D)

作成	鳥居 達生	1978年7月
形式	サブルーチン 言語 ; FORTRAN サイズ ; 106, 107, 104, 105行	

(1) 概要

有限区間 (a, b) で滑らかで有界な関数 $f(x)$ の積分 $\int_a^b f(x) dx$ の近似値を要求精度で自動的に求める。この方法の基礎は $f(x)$ をいったん $[a, b]$ でチェビシエフ級数展開し、項別積分することである。したがって、この級数の収束が速い程、この積分法は少ない標本数で所要の精度に到達する。関数の滑らかさが増す程、そのチェビシエフ級数の収束は速い。

被積分関数が閉区間 $[a, b]$ で定義されているならば、両端を標本点に用いる閉じた積分公式を用いる方がよい。開区間で与えられるならば開いた積分公式を使用せざるを得ない。

(2) 使用法

CALL AQDCCS/D (A, B, F, EPSA, EPSR, NMIN, NMAX, S, ERR, N, ICON)

CALL AQDCOS/D (A, B, F, EPSA, EPSR, NMIN, NMAX, S, ERR, N, ICON)

引数	型と種類	属性	内容
A B	実数型	入力	A, Bは積分区間の下限, 上限。保存される。
F	実数型 関数副プログラム	入力	被積分関数を1変数の関数副プログラムとして使用者が定義する。
EPSA EPSR	実数型	入力	要求精度, EPSA, EPSRはそれぞれ絶対誤差, 相対誤差の限界 ≥ 0 。保存される。
NMIN NMAX	整数型	入力	標本数の下限, 上限。NMAX \geq NMIN ≥ 0 AQDCCS/Dの場合 NMAX ≤ 1025 AQDCOS/Dの場合 NMAX ≤ 1023 保存される。
S ERR	実数型	出力	Sは求める積分の近似値。ERRはその絶対誤差の推定値。
N	整数型	出力	Sを求めるに使用した標本数。
ICON	整数型	出力	ICON = 0 : 正常。 ICON = 10000 : 要求精度がきびしすぎる。使用している計算機では限界の精度が得られているので正常と考えてよい。

引数	型と種類	属性	内容
			ICON = 20000 : 異常。標本数その限界 NMAX まで増しても要求精度が得られない。 ICON = 30000 : パラメータ・エラー。

(3) 性能

被積分関数のチェビシェフ級数展開には、中点式に基づく高速 cosine 変換を用いているので、標本数を N とするとき、実数乗算回数は、約 $\frac{N}{2} \log_2 N$ である。

(4) 計算法

区間 $[a, b]$ を 1 次変換によって $[-1, 1]$ にうつし、被積分関数 $f(t)$ をチェビシェフ級数展開し、その項別積分を行う。

$$\begin{aligned} \int_{-1}^1 f(t) dt &= \frac{1}{2} \int_{-1}^1 (f(t) + f(-t)) dt \\ &= \int_{-1}^1 \sum_{k=0}^{\infty} a_{2k} T_{2k}(t) dt \\ &= a_0 - 2 \left(\frac{a_2}{1 \cdot 3} + \frac{a_4}{3 \cdot 5} + \frac{a_6}{5 \cdot 7} + \dots \right) \end{aligned}$$

閉じた積分公式、開いた積分公式の誤差評価は、それぞれの標本数を $N+1$ 、 $N-1$ とするとき、

$$(|a_{N-2}| + |a_N|)/N$$

及び、

$$4(|a_{N-4}| + |a_{N-2}|)/N$$

で行っている。係数 4 は便宜的なものである。

それぞれの積分法における“相対誤差”は各絶対誤差の評価を被積分関数のノルム $\|f(t) + f(-t)\|_{\infty}$ で割ったものである。ただし、関数のノルムは、

$$\|f\|_{\infty} = \max_j |f(x_j)|, \quad x_j \text{ は標本点}$$

とする。また、丸め誤差の水準（計算誤差あるいは伝播誤差）は、機械精度の最小単位を u として

$$16u \|f(t) + f(-t)\|$$

で評価した。安全係数 16 は経験的に定めた。以上準備して収束の判定法について述べる。

要求精度 ϵ_a （絶対誤差）と ϵ_r （相対誤差）が与えられたとき、

$$\text{絶対誤差の評価値} \leq \max \{ \epsilon_a, \text{計算誤差} \}$$

$$\text{相対誤差の評価値} \leq \epsilon_r$$

の少なくともいずれか一つの条件を満たしたならば収束したと判定する。満たさなければ閉じた積分公式の場合標本数を 17, 33, 65, … 開いた積分公式においては 15, 31, 63, … と倍々に増していく。

$\epsilon_a = \epsilon_r = 0$ とおけば、最高の精度（すなわち、打ち切り誤差に比べ丸め誤差が支配的）の結果が得られる。

上述の収束の判定法の妥当性は被積分関数の性質に依存する。被積分関数 $f(t)$ が十分滑らかならば、標本数も少なく（ 2^5 前後）誤差評価も適切であるが、 $f(t)$ が微分不可能ならば標本数は増大し、誤差の評価値は実際より過少となる傾向がある。 $f(t)$ が実軸上の区間 $[-1, 1]$ で解析的であっても、その

特異点が $[-1, 1]$ に接近すれば同様の困難性が生ずる。

(5) 使用例

解析解がわかっている次の3種の問題

$$(1) \int_{-1}^1 \frac{1-t^2}{1-2tx+t^2} dx = \left(\frac{1}{t}-t\right) \log \frac{1+t}{1-t} \quad t = 1/2, 3/4, 15/16$$

$$(2) \int_{-1}^1 \frac{a}{a^2+x^2} dx = 2 \tan^{-1} \frac{1}{a} \quad a = 1, 1/4, 1/16$$

$$(3) \int_{-1}^1 \cos ax dx = \frac{2}{a} \sin a \quad a = 4, 16, 64$$

でテストする。

要求精度 ϵ を 10^{-2} , 10^{-4} , 10^{-6} , ... と変えつつ上の積分を計算し, 真値, 計算値, 誤差, 誤差評価, 標本数, 計算が正常に行われたか否かの指標 (ICON) を印字するプログラムを示す。

```

C      TEST PROBLEMS FOR SUBROUTINE AQDCCS AND AQDCOS.
C      1978.11.15
      DIMENSION PARAM(3,3)
      DATA PARAM/0.5,0.75,0.9375,1.0,0.25,0.0625,4.0,16.0,64.0/
      COMMON T,J
      EXTERNAL F
      ZERO=AMACH(ZERO)
      EPSA=1.0E-02
      EPSR=0.0
      NMIN=0
      NMAX=1025
      A=-1.0
      B=1.0
      10 WRITE(6,600) EPSA
600  FORMAT(1H0/4X,32HPERMISSIBLE ABSOLUTE ERROR BOUND,E15.3/)
      DO 20 J=1,3
      DO 20 I=1,3
      T=PARAM(I,J)
      CALL AQDCCS(A,B,F,EP SA,EP SR,NMIN,NMAX,S,ERR,N,ICON)
      TS=TRUE(T,J)
      ERROR=TS-S
      WRITE(6,601) J,I,T,TS,S,ERROR,ERR,N,ICON
601  FORMAT(1H ,2I4,F8.4,2F15.06,2E13.03,2I8,5X,6HAQDCCS)
      CALL AQDCOS(A,B,F,EP SA,EP SR,NMIN,NMAX,S,ERR,N,ICON)
      ERROR=TS-S
      WRITE(6,602) J,I,T,TS,S,ERROR,ERR,N,ICON
602  FORMAT(1H ,2I4,F8.4,2F15.06,2E13.03,2I8,5X,6HAQDCOS/)
      20 CONTINUE
      EPSA=EP SA*1.0E-02
      IF(EP SA.GT.ZERO) GO TO 10
      STOP
      END
      FUNCTION F(P)
      COMMON T,J
      GO TO (1,2,3),J
      1 F=(1.0-T*T)/(1.0-2.0*T*P+T*T)
      RETURN
      2 F=T/(T*T+P*P)
      RETURN
      3 F=COS(T*P)
      RETURN
      END
      FUNCTION TRUE(P,J)
      GO TO (1,2,3),J
      1 TRUE=(1.0/P-P)*ALOG((1.0+P)/(1.0-P))
      RETURN

```

```

2 TRUE=2.0*ATAN(1.0/P)
  RETURN
3 TRUE=2.0*SIN(P)/P
  RETURN
  END

```

要求精度 10^{-5} の下の計算結果

問題	パラメータ	種別	積分の計算値	誤差	誤差評価	標本数	正常・異常
(1)	t = 1/4	AQDCCS	1.647918	0.0	0.781 E - 06	33	0
		AQDCOS	1.647918	0.0	0.781 E - 06	31	0
	t = 3/4	AQDCCS	1.135114	0.596 E - 07	0.166 E - 05	65	10000
		AQDCOS	1.135114	0.298 E - 07	0.166 E - 05	63	10000
	t = 15/16	AQDCCS	0.443557	0.104 E - 06	0.714 E - 05	257	10000
		AQDCOS	0.443548	0.858 E - 05	0.647 E - 05	127	10000
(2)	a = 1	AQDCCS	1.570796	- 0.298 E - 07	0.531 E - 06	17	0
		AQDCOS	1.570796	- 0.298 E - 07	0.472 E - 06	31	0
	a = 1/4	AQDCCS	2.651635	0.596 E - 07	0.184 E - 05	65	10000
		AQDCOS	2.651635	0.119 E - 06	0.184 E - 05	63	10000
	a = 1/16	AQDCCS	3.016755	0.179 E - 06	0.735 E - 05	257	10000
		AQDCOS	3.016755	0.238 E - 06	0.735 E - 05	255	10000
(3)	a = 4	AQDCCS	- 0.378401	0.0	0.469 E - 06	17	0
		AQDCOS	- 0.378401	0.0	0.476 E - 06	31	0
	a = 16	AQDCCS	- 0.035988	- 0.484 E - 07	0.477 E - 06	33	0
		AQDCOS	- 0.035988	- 0.829 E - 07	0.477 E - 06	63	0
	a = 64	AQDCCS	0.028751	- 0.424 E - 07	0.477 E - 06	129	0
		AQDCOS	0.028751	- 0.405 E - 07	0.477 E - 06	127	0

VCOSS / D

VSINS / D

Value of Cosine Series (VCOS/D)

cosine 級数の値 (VCOSS/D)

Value of Sine Series (VSIN/D)

sine 級数の値 (VSINS/D)

作 成	鳥居 達生 1978 年 12 月
形 式	サブルーチン 言語 ; FORTRAN サイズ ; 38, 39, 38, 39行

(1) 概 要

cosine 級数

$$\sum_{0 \leq k < N} a_k \cos k\theta$$

及び sine 級数

$$\sum_{k=1}^N a_k \sin k\theta$$

の値を計算する。

(2) 使用法

CALL VCOSS/D (A, N, T, F, ICON)

CALL VSINS/D (A, N, T, F, ICON)

引 数	型 と 種 類	属 性	内 容
A	実 数 型 1 次 元 配 列	入 力	配列Aの大きさ $\geq N$ 。 VCOSS/Dの場合、 a_0, a_1, \dots, a_{N-1} が、Aの上に格納されている。
N	整 数 型		VSINS/Dの場合、 a_1, \dots, a_N が、Aの上に格納されている。 A及びNは保存される。 $N \geq 1$
T	実 数 型	入 力	任意の実数。保存される。
F	実 数 型	出 力	$\theta = t$ における cosine 級数 (VCOSS) の値及び sine 級数 (VSINS) の値。
ICON	整 数 型	出 力	ICON=0: 正常。 ICON=30000: パラメータ・エラー。

(3) 計算法

cosine 級数の和は第1種チェビシェフ級数和と同じくクレンショウの方法で求められる。sine級数の和は第2種チェビシェフ級数和に $\sin \theta$ をかければよい。

(4) 使用例

1. cosine 級数の計算例

周期関数のフーリエ展開ができるならば、その項別積分は容易である。そこでサブルーチンVCOSSの使用例として sine 級数の積分を求める例を示すことにする。

sine 関数の母関数の項別積分は、

$$\int_0^\varphi \frac{\sin \theta}{1 - 2t \cos \theta + t^2} d\theta = \sum_{k=0}^{\infty} a_k \cos k\varphi$$

ただし、

$$a_k = -t^{k-1}/k, k \geq 1$$

$$a_0 = -2 \sum_{k=1}^{\infty} a_k$$

となるので、被積分関数をサブルーチン FSINOS を用いて展開し、項別積分する。

積分の上限 φ を $1/12\pi$, $2/12\pi$, ..., $6/12\pi$ と動かしながらサブルーチン VCOSS を用いて cosine 級数の値を求める。この積分の解析解は、

$$\frac{1}{2t} \log \left\{ \frac{1 - 2t \cos \varphi + t^2}{(1-t)^2} \right\}$$

である。

```

C      TEST FOR SUBROUTINE VCOSS
      DIMENSION A(256)
      EXTERNAL F
      COMMON T
      TRUE(P,T)=ALOG((1.0-2.0*T*COS(P)+T*T)/(1.0-T)**2)*0.5/T
      T=0.5
      NX=6
      HPI=2.0*ATAN(1.0)
      H=HPI/FLOAT(NX)
      EPSA=1.0E-05
      EPSR=0.0
      NMIN=0
      NMAX=255
      CALL FSINOS(F,EPSA,EPSR,NMIN,NMAX,A,N,ERR,ILL)
      NP1=N+1
      S=0.0
      DO 10 I=1,N
        K=NP1-I
        A(K+1)=-A(K)/FLOAT(K)
        S=A(K+1)+S
10     CONTINUE
      A(1)=-S-S
      THETA=H
      DO 20 I=1,NX
        CALL VCOSS(A,NP1,THETA,VA,ICON)
        ICON=ICON+ILL
        ERV=TRUE(THETA,T)-VA
        WRITE(6,600) I,VA,ERV,T,N,ICON
600    FORMAT(1H0,4X,I4,F15.06,E15.03,F8.3,2I8)
        THETA=THETA+H
20     CONTINUE
      STOP
      END
      FUNCTION F(P)
      COMMON T
      F=SIN(P)/(1.0-2.0*T*COS(P)+T*T)
      RETURN
      END

```

sine 母関数の展開とその積分

φ	積 分	誤 差
$\pi / 12$	0.127774	- 0.119 E - 07
$2 \pi / 12$	0.429115	- 0.217 E - 07
$3 \pi / 12$	0.775452	- 0.300 E - 07
$4 \pi / 12$	1.098612	- 0.274 E - 07
$5 \pi / 12$	1.377436	- 0.194 E - 07
$6 \pi / 12$	1.609438	- 0.190 E - 07

注：sine 級数展開のための要求精度 10^{-5} (入力)

パラメータ $t = 1/2$ (入力)

標 本 数 $N = 31$ (出力)

2. sine 級数の計算例

楕円積分

$$F(\varphi, \alpha) = \int_0^{\varphi} \frac{d\theta}{\sqrt{1 - \sin^2 \alpha \sin^2 \theta}}$$

の計算例を示す。簡単のため $\alpha = \pi/4$ とする。被積分関数を cosine 級数に展開し、これを項別積分すれば、定数項を除き sine 級数となるので、 φ を動かしながら、この和を求める。定数項の積分だけは別途計算して加えればよい。以下の例では、 $\varphi = 1/12\pi, 2/12\pi, \dots, 6/12\pi$ と動かしている。

```

C      TEST FOR SUBROUTINE VSINS.
      DIMENSION A(257)
      EXTERNAL F
      NX=6
      HPI=2.0*ATAN(1.0)
      H=HPI/FLOAT(NX)
      EPSA=1.0E-05
      EPSR=0.0
      NMIN=0
      NMAX=257
      CALL FCOSCS(F,EPSA,EPSR,NMIN,NMAX,A,N,ERR,ILL)
      A(N)=A(N)*0.5
      M=N-1
      CONST=A(1)*0.5
      DO 10 I=1,M
      A(I)=A(I+1)/FLOAT(I)
10    CONTINUE
      T=H
      DO 20 I=1,NX
      CALL VSINS(A,M,T,V,ICON)
      V=V+CONST*T
      ICON=ICON+ILL
      WRITE(6,600) I,V,N,ICON
600   FORMAT(1H0,4X,I4,F15.06,2I8)
      T=T+H
20    CONTINUE
      STOP
      END
      FUNCTION F(P)
      F=1.0/SQRT(1.0-0.5*SIN(P)**2)
      RETURN
      END
    
```

楕円積分

φ	積 分	備 考
$\pi/12$	0.263297	標本数 = 17
$2\pi/12$	0.535623	
$3\pi/12$	0.826018	真 値
$4\pi/12$	1.142429	1.14242906
$5\pi/12$	1.487885	
$6\pi/12$	1.854075	1.85407468

VCHB1S / D DCHB3S / D
 DCHB1S / D ICHB3S / D
 ICHB1S / D
 VCHB3S / D

Value of Chebyshev Series (VCHB1S/D)

チェビシェフ級数の値 (VCHB1S/D)

Differential Coefficient (DCHB1S/D)

チェビシェフ級数の微係数 (DCHB1S/D)

Value of Indefinite Integral (ICHB1S/D)

チェビシェフ級数の不定積分の値 (ICHB1S/D)

Value of Shifted Chebyshev Series (VCHB3S/D)

ずらしチェビシェフ級数の値 (VCHB3S/D)

Differential Coefficient (DCHB3S/D)

ずらしチェビシェフ級数の微係数 (DCHB3S/D)

Value of Indefinite Integral (ICHB3S/D)

ずらしチェビシェフ級数の不定積分の値 (ICHB3S/D)

作 成	鳥居 達生	1978 年 12 月
形 式	サブルーチン	言語 ; FORTRAN サイズ ; 75, 76, 75, 76, 75, 76, 80, 81, 80, 81, 80, 81行

(1) 概 要

第 1 種チェビシェフ多項式系の N 項の級数

$$\sum_{0 \leq k < N} a_k T_k(x)$$

に関して、それぞれ次の計算を行う。

1. 任意の点 $x \in [-1, 1]$ における級数 (1) の値を求める。
2. 点 x における微係数を計算する。
3. 任意の点 $x \in [-1, 1]$ までの積分

$$\int_{-1}^x \left(\sum_{0 \leq k < N} a_k T_k(t) \right) dt \quad \text{を求める。}$$

同様にずらしチェビシェフ多項式系の級数

$$\sum_{0 \leq k < N} a_k T_k^*(x)$$

に関して、級数の値、微係数、積分

$$\int_0^x (\sum' a_k T_k^*(t)) dt$$

を求めるのが、VCHB3S, DCHB3S, ICHB3S である。

(2) 使用法

CALL VCHB1S/D (A, N, X, F, ICON)
 CALL DCHB1S/D (A, N, X, F, ICON)
 CALL ICHB1S/D (A, N, X, F, ICON)
 CALL VCHB3S/D (A, N, X, F, ICON)
 CALL DCHB3S/D (A, N, X, F, ICON)
 CALL ICHB3S/D (A, N, X, F, ICON)

引数	型と種類	属性	内容
A	実数型 1次元配列	入力	配列Aの大きさ $\geq N$ 。A(1), A(2), ..., A(N) にフーリエ係数 a_0, a_1, \dots, a_{N-1} が格納されている。A, N は $N \geq 1$
N	整数型		
X	実数型	入力	$-1 \leq X \leq 1$ 。
F	実数型	出力	それぞれのサブルーチンの計算値。
ICON	整数型	出力	ICON = 0 : 正常。 ICON = 30000 : パラメータ・エラー。

(3) 計算法

クレンショウの算法を用いる、N 項のチェビシェフ級数

$$S_N(x) = \sum'_{0 \leq k < N} a_k T_k(x)$$

の点 $x \in [-1, 1]$ における値は、漸化式

$$b_N = 0$$

$$b_{N-1} = a_{N-1}$$

$$b_k = 2x b_{k+1} - b_{k+2} + a_k$$

$$k = N-2, N-1, \dots, 1, 0$$

$$S_N(x) = \frac{1}{2} (b_0 - b_2)$$

$$\text{あるいは} = x b_1 - b_2 + \frac{a_0}{2}$$

によって計算できる。N-1 次のチェビシェフ級数の和は N 回の乗算で求められる。

中間結果である数列 $\{b_k\}$ のためには、配列は使用しない。N 次のチェビシェフ級数の点 x における微係数

$$\sum_{k=1}^N a_k \frac{d}{dx} T_k(x) \Big|_{x=x}$$

は、漸化式

$$b_{N+1} = 0$$

$$b_N = N a_N$$

$$b_k = 2x b_k - b_{k+1} + k a_k$$

$$k = N-1, N-2, \dots, 1$$

$$\text{微係数} = b_1$$

によって得られる。チェビシェフ級数の不定積分は、

$$\int_{-1}^x \sum_{0 \leq k < N} a_k T_k(x) dx = \sum_{0 \leq k < N} a_k \int_{-1}^x T_k(x) dx = \sum_{k=1}^N \frac{a_{k-1} - a_{k+1}}{2k} (T_k(x) - (-1)^k)$$

$$\text{ただし, } a_{N+1} = a_N = 0$$

と表されるから求める積分値は、

$$b_{N+1} = 0, c_N = a_{N-1} / 2N$$

$$b_N = c_N, S_N = c_N$$

$$c_k = (a_{k-1} - a_{k+1}) / 2k$$

$$b_k = 2x b_{k+1} - b_{k+2} + c_k, S_{k-1} = c_k - S_k$$

$$k = N, N-1, \dots, 1$$

$$\text{積分値} = x b_1 - b_2 + S_1$$

によって計算できる。ずらしチェビシェフ多項式の場合も類似の方法で計算できる。

(4) 使用例

簡単のため、指数関数で、数値微分、積分をテストする。区間 $[-1, 1]$ で、

$$f(x) = e^x$$

を与えられた要求精度の下でチェビシェフ級数展開し、VCHB1S, DCHB1S, ICHB1Sを用いて、

$$f(x), f'(x), \int_{-1}^x f(x) dx$$

$$x = \frac{i}{4}, i = -4, -3, \dots, 3, 4$$

の各値を計算する。また、同じ指数関数を区間 $[0, 8]$ でずらしチェビシェフ級数展開し、

$$f(x), f'(x), \int_0^x f(x) dx, x = 0, 1, \dots, 8$$

の計算例を示す。このとき、区間 $[0, 8]$ を $[0, 1]$ に標準化するための変数変換が必要である。

```

C      TEST FOR SUBROUTINE VCHB1S,DCHB1S AND ICHB1S
      DIMENSION A(257)
      EXTERNAL F
      EPSA=1.0E-05
      EPSR=0.0
      NMIN=0
      NMAX=257
      CALL FCHB1S(F,EPSA,EPSR,NMIN,NMAX,A,N,ERR,ILL1)
      A(N)=A(N)*0.5
      H=0.25
      X=-1.0
10    CONTINUE
      CALL VCHB1S(A,N,X,V,ILL2)
      CALL DCHB1S(A,N,X,D,ILL3)
      CALL ICHB1S(A,N,X,V1,ICON)
      ICON=ICON+ILL1+ILL2+ILL3
      TRUEV=EXP(X)
      ERV=TRUEV-V
      ERD=TRUEV-D

```

```

        ERI=TRUEV-EXP(-1.0)-VI
        WRITE(6,600) X,V,ERV,D,ERD,VI,ERI,N,ICON
600    FORMAT(1H0,4X,F8.3,3(F15.06,E15.03),2I8)
        X=X+H
        IF(X.LE.1.0) GO TO 10
        STOP
        END
        FUNCTION F(P)
        F=EXP(P)
        RETURN
        END
C      TEST FOR SUBROUTINE VCHB3S,DCHB3S AND ICHB3S
        DIMENSION A(257)
        EXTERNAL F
        EPSA=1.0E-05
        EPSR=0.0
        NMIN=0
        NMAX=257
        CALL FCHB3S(F,EPSA,EPSR,NMIN,NMAX,A,N,ERR,ILL1)
        A(N)=A(N)*0.5
        Y=0.0
        H=1.0
10     CONTINUE
C      APPLY THE VARIABLE TRANSFORMATION
        X=Y/8.0
        CALL VCHB3S(A,N,X,V,ILL2)
        CALL DCHB3S(A,N,X,D,ILL3)
        CALL ICHB3S(A,N,X,VI,ICON)
        ICON=ICON+ILL1+ILL2+ILL3
        D=D/8.0
        VI=VI*8.0
        TRUEV=EXP(Y)
        ERV=TRUEV-V
        ERD=TRUEV-D
        ERI=TRUEV-1.0-VI
600    WRITE(6,600) Y,V,ERV,D,ERD,VI,ERI,N,ICON
        FORMAT(1H0,4X,F8.3,3(F15.06,E15.03),2I8)
        Y=Y+H
        IF(Y.LE.8.0) GO TO 10
        STOP
        END
        FUNCTION F(P)
C      APPLY THE VARIABLE TRANSFORMATION
        Q=8.0*P
        F=EXP(Q)
        RETURN
        END

```

e^x の $\{T_k(x)\}$ による展開, 級数和, 微係数及び不定積分

x	級数和	誤差	微係数	誤差	積分	誤差
-1.00	0.367879	-0.745 E - 08	0.367879	0.291 E - 06	0.000000	0.373 E - 08
-0.75	0.472367	-0.745 E - 08	0.472366	0.112 E - 06	0.104487	0.745 E - 08
-0.50	0.606531	0.149 E - 07	0.606531	-0.104 E - 06	0.238651	0.745 E - 08
-0.25	0.778801	-0.149 E - 07	0.778801	-0.596 E - 07	0.410921	0.0
0.00	1.000000	0.0	1.000000	0.134 E - 06	0.632121	-0.745 E - 08
0.25	1.284025	0.0	1.284026	-0.119 E - 06	0.916146	0.745 E - 08
0.50	1.648721	-0.298 E - 07	1.648721	-0.596 E - 07	1.280842	-0.224 E - 07
0.75	2.117000	0.0	2.117000	0.596 E - 07	1.749121	0.745 E - 08
1.00	2.718282	-0.596 E - 07	2.718281	0.119 E - 05	2.350402	-0.522 E - 07

注：展開のための要求精度 $\epsilon = 10^{-5}$

標本数 $N = 9$

e^x の $\{T_k^*(\frac{x}{8})\}$ による展開, 級数和, 微係数及び不定積分

x	級数和	誤差	微係数	誤差	積分	誤差
0.0	0.999996	0.381 E - 05	0.999893	0.107 E - 03	0.000008	-0.763 E - 05
1.0	2.718304	-0.219 E - 04	2.718166	0.115 E - 03	1.718302	-0.200 E - 04
2.0	7.389103	-0.468 E - 04	7.389107	-0.507 E - 04	6.389076	-0.201 E - 04
3.0	20.085506	0.305 E - 04	20.085672	-0.135 E - 03	19.085560	-0.229 E - 04
4.0	54.598145	0.572 E - 05	54.597870	0.280 E - 03	53.598206	-0.553 E - 04
5.0	148.413208	-0.496 E - 04	148.413406	-0.248 E - 03	147.413169	-0.114 E - 04
6.0	403.428611	0.183 E - 03	403.428878	-0.839 E - 04	402.428771	0.229 E - 04
7.0	1096.63305	0.916 E - 04	1096.63201	0.113 E - 02	1095.63314	0.0
8.0	2980.958	0.0	2980.9678	-0.983 E - 02	2979.95788	0.0122 E - 03

注：展開のための要求精度 $\epsilon = 10^{-5}$

標本数 $N = 17$

VCHB2S / D

ICHB2S / D

Value of Second Kind Chebyshev Series (VCHB2S/D)

第2種チェビシェフ級数の値 (VCHB2S/D)

Value of Indefinite Integral (ICHB2S/D)

第2種チェビシェフ級数の不定積分の値 (ICHB2S/D)

作 成	鳥居 達生 1978年12月
形 式	サブルーチン 言語; FORTRAN サイズ; 47, 48, 47, 48行

(1) 概 要

第2種チェビシェフ級数

$$\sum_{0 \leq k < N} a_k U_k(x) \tag{1}$$

の点 x における値と積分

$$\int_{-1}^x \sum_k a_k U_k(x) dx \tag{2}$$

を求める。

(2) 使用法

CALL VCHB2S/D (A, N, X, F, ICON)

CALL ICHB2S/D (A, N, X, F, ICON)

引 数	型 と 種 類	属 性	内 容
A	実 数 型 1 次 元 配 列	入 力	配列Aの大きさ $\geq N$ 。A(1), A(2), ..., A(N) にフーリエ係数 a_0, a_1, \dots, a_{N-1} が格納されている。A, Nは保存される。 $N \geq 1$
X	実 数 型	入 力	$-1 \leq X \leq 1$ 。保存される。
F	実 数 型	出 力	それぞれのサブルーチンの計算値。
ICON	整 数 型	出 力	ICON = 0 : 正常。 ICON = 30000 : パラメータ・エラー。

(3) 計算法

第2種チェビシェフ多項式系が従う漸化式は、第1種の場合と初期条件を除けば同じであるから級数

(1) の値は、

$$b_N = 0$$

$$b_{N-1} = a_{N-1}$$

$$b_k = 2x b_{k+1} - b_{k+2} + a_k$$

$$k = N-2, N-3, \dots, 1, 0$$

$$\text{級数 (1) の値} = b_1$$

で求められる。また、不定積分 (2) は、

$$\sum_{k=1}^N \frac{a_{k-1}}{k} T_k(x) - \sum_{k=1}^N (-1)^k \frac{a_{k-1}}{k}$$

で求められる。したがって、第 1 種チェビシエフ級数の不定積分(ICHB1S)にならって計算すればよい。

(4) 使用例

指数関数 e^x を $[-1, 1]$ で第 2 種チェビシエフ級数で展開 (FCHB2S を用いる) して、この級数の値及び積分を求める。点 x は区間 $[-1, 1]$ を 8 等分した分点にとった。

```

C      TEST FOR SUBROUTINE VCHB2S AND ICHB2S
      DIMENSION A(255)
      EXTERNAL F
      EPSA=1.0E-05
      EPSR=0.0
      NMIN=0
      NMAX=255
      CALL FCHB2S(F, EPSA, EPSR, NMIN, NMAX, A, N, ERR, ILL1)
      H=0.25
      X=-1.0
10     CONTINUE
      CALL VCHB2S(A, N, X, VA, ILL2)
      CALL ICHB2S(A, N, X, VI, ICON)
      ICON=ICON+ILL1+ILL2
      TRUEV=EXP(X)
      ERV=TRUEV-VA
      ERI=TRUEV-EXP(-1.0)-VI
      WRITE(6,600) X, VA, ERV, VI, ERI, N, ICON
600    FORMAT(1H0,4X,F8.3,2(F15.06,E15.03),2I8)
      X=X+H
      IF(X.LE.1.0) GO TO 10
      STOP
      END
      FUNCTION F(P)
      F=EXP(P)
      RETURN
      END
  
```

e^x の $\{U_k(x)\}$ による展開、級数和と不定積分

x	級数和	誤差	積分	誤差
-1.00	0.367879	0.0	0.000000	-0.373 E - 08
-0.75	0.472367	0.745 E - 08	0.104487	0.745 E - 08
-0.50	0.606531	-0.0	0.238651	0.745 E - 08
-0.25	0.778801	-0.298 E - 07	0.410921	-0.745 E - 08
0.00	1.000000	0.0	0.632121	-0.745 E - 08
0.25	1.284025	0.0	0.916146	-0.745 E - 08
0.50	1.648721	-0.298 E - 07	1.280842	-0.224 E - 08
0.75	2.117000	0.0	1.749121	0.745 E - 08
1.00	2.718282	-0.596 E - 07	2.350402	-0.522 E - 07

注：展開のための要求精度 $\epsilon = 10^{-5}$

標本数 $N = 15$

DRCH1S / D

DRCH3S / D

IICH1S / D

IICH3S / D

Derivative of First Kind Chebyshev Series (DRCHB1/DRCH1D)

第1種チェビシェフ級数の導関数 (DRCH1S/D)

Derivative of Shifted Chebyshev Series (DRCHB3/DRCH3D)

ずらしチェビシェフ級数の導関数 (DRCH3S/D)

Indefinite Integral of First Kind Chebyshev Series (IICHB1/IICH1D)

第1種チェビシェフ級数の不定積分 (IICH1S/D)

Indefinite Integral of Shifted Chebyshev Series (IICHB3/IICH3D)

ずらしチェビシェフ級数の不定積分 (IICH3S/D)

作成	鳥居 達生	1978年12月
形式	サブルーチン	言語 ; FORTRAN サイズ ; 24, 24, 24, 24, 26, 26, 26, 27行

(1) 概要

第1種チェビシェフ級数

$$\sum_{0 \leq k < N} a_k T_k^*(x)$$

の項別微分, 積分をチェビシェフ級数に表す。同様にずらしチェビシェフ級数

$$\sum_{0 \leq k < N} a_k T_k^*(x)$$

の項別微分, 積分を $\{T_k^*(x)\}$ で展開する。

(2) 使用法

CALL DRCH1S (A, NA, B, NB, ICON)

CALL DRCH1D (A, NA, B, NB, ICON)

CALL IICH1S (A, NA, B, NB, ICON)

CALL IICH1D (A, NA, B, NB, ICON)

CALL DRCH3S (A, NA, B, NB, ICON)

CALL DRCH3D (A, NA, B, NB, ICON)

CALL IICH3S (A, NA, B, NB, ICON)

CALL IICH3D (A, NA, B, NB, ICON)

引数	型と種類	属性	内容
A NA	実数型 1次元配列 整数型	入力	DRCHB1S/DRCH1D, IICHB1S/IICH1Dの場合: 第1種 チェビシェフ級数の係数がAに格納されている。項数 NA ≥ 1 DRCHB3S/DRCH3D, IICHB3S/IICH3Dの場合: ずら しチェビシェフ多項式の級数である。
B NB	実数型 1次元配列 整数型	出力	項別積分あるいは微分した級数の係数が配列Bに格納される。 NB ≥ 1は出力の係数の個数。
ICON	整数型	出力	ICON = 0 : 正常。 ICON = 30000 : パラメータ・エラー。

(3) 計算法

N項のチェビシェフ級数を項別積分して

$$\int_{-1}^x \sum'_{0 \leq k < N} a_k T_k(x) dx = \sum'_{0 \leq k < N} b_k T_k(x)$$

とすれば、係数間に、

$$b_k = (a_{k-1} - a_{k+1}) / 2k, k \geq 1$$

$$b_0 = 2 \sum_{k=1}^{N+1} (-1)^{k-1} b_k$$

ただし、 $a_{N+1} = a_N = 0$

が成り立つ。項別微分の場合は、係数 $\{b_k\}$ を与えて $\{a_k\}$ を求める。ずらしチェビシェフ多項式系で展開する場合には、

$$\int_{-1}^x \sum' a_k T_k^*(x) dx = \sum' b_k T_k^*(x)$$

の係数間の関係

$$b_k = (a_{k-1} - a_{k+1}) / 4k, k \geq 1$$

を用いればよい。

(4) 使用例

三角関数をチェビシェフ級数展開すれば、ベッセル関数が現れる。

$$\cos ax = J_0(a) + 2 \sum_{k=1}^{\infty} (-1)^k J_{2k}(a) T_{2k}(x)$$

$$\sin ax = 2 \sum_{k=1}^{\infty} (-1)^k J_{2k+1}(a) T_{2k+1}(x)$$

右辺の項別積分、微分を行う。同様に $\cos ax$, $\sin ax$ をずらしチェビシェフ多項式系で展開し、項別積分、微分をするプログラムを以下に示す。積分定数は $x = -1$ (ずらしチェビシェフ多項式を用いる場合は $x = 0$) で級数和が0となるよう定めた。

```

C      TEST FOR SUBROUTINE IICH1S AND DRCH1S.
      DIMENSION A(258),B(258),C(258)
      EXTERNAL F
      COMMON L,T
      DATA EPSA,EPSR,NMIN,NMAX/0.0,0.0,0,257/

```

```

T=10.0
DO 10 L=1,2
CALL FCHB1S(F,EPSA,EPSR,NMIN,NMAX,A,N,ERR,ILL1)
CALL IICH1S(A,N,B,NB,ILL2)
CALL DRCH1S(A,N,C,NC,ICON)
ICON=ICON+ILL1+ILL2
WRITE(6,601) L,T,ICON
601 FORMAT(////7X,9HPROBLEM (,I2,1H),5X,1HT,F08.3,6X,4HICON,I8)
WRITE(6,600) (I,A(I),B(I),C(I),I=1,NB)
600 FORMAT(1H0/(1H ,I8,3F25.06))
CALL VCHB1S(B,NB,-1.0,VB,ICON)
WRITE(6,603) VB
603 FORMAT(//8X,26HCHECK OF INTEGRAL CONSTANT,E25.5)
10 CONTINUE
STOP
END
FUNCTION F(P)
COMMON L,T
GO TO (1,2),L
1 F=SIN(T*P)
RETURN
2 F=COS(T*P)
RETURN
END

```

数値積分ルーチンの選び方

名大プログラム・ライブラリーには、一次元と多次元、有限区間と無限区間、固定則積分と自動積分など、種々の場合に即して多数のすぐれた数値積分ルーチンが用意されているので、これを以下に示す指針に基づいて慎重に選択すれば、精度と速度の両面において多大の効果を上げることができる。簡単のために推奨ルーチンの名前は単精度用のもので代表させることにする。

(A) 一次元有限区間

1. 変化の穏やかな解析関数

(1) 固定則積分 GASNS

(2) 自動積分 QDAPBS, DEFINS, AQNN9S

2. 振動型の解析関数 QDAPBS

3. ピーク型の関数 AQNN9S

4. 端に特異点をもつ解析関数 DEFINS

5. 特異点や不連続点をもつ関数 AQNN9S

6. 挙動のよく分からない関数 AQNN9S

7. 周期関数の一周期積分 TRAPZS

(B) 一次元半無限区間の積分 $\int_0^{\infty} e^{-x} f(x) dx$ で $f(x)$ が穏やかな関数の場合

1. 固定則積分 GSLNS

2. 自動積分 HINFAS, HINFES

(C) 一次元全無限区間

1. $\int_{-\infty}^{\infty} e^{-x^2} f(x) dx$ の形で $f(x)$ が穏やかな関数の場合

(1) 固定則積分 GSHNS

(2) 自動積分 INFINS

2. $\int_{-\infty}^{\infty} f(x) dx$ の形で $f(x)$ が急減少の場合 TRAPZS

(D) 多次元固定則積分

1. 関数入力 MQPRRS

2. データ入力 MQNCDS

3. 高次元 MQFSRS

(E) 多次元自動積分 AQMDS, AQNDS

なお、必要に応じて積分区間を分割したり、変数変換によって上下限を0や1のような正確に表現できる値にするなどの前処理を施すことが結果の精度を高めるのに役立つ。

GASNS / D

GSLNS / D

GSHNS / D

Gaussian Quadrature

ガウス型数値積分

作成	秦野 甯世	1980年8月
形式	サブルーチン	言語; FORTRAN サイズ; 50, 51, 58, 60, 68, 69行

(1) 概要

ガウスの数値積分公式により、1次元の積分値を求める。

GASNS/D…ガウス・ルジャンドル公式により有限区間積分

$$Y = \int_a^b f(x) dx$$

を求める。

GSLNS/D…ガウス・ラゲール公式により半無限区間積分

$$Y = \int_0^{\infty} e^{-x} \cdot f(x) dx$$

を求める。

GSHNS/D…ガウス・エルミート公式により、全無限区間積分

$$Y = \int_{-\infty}^{\infty} e^{-x^2} \cdot f(x) dx$$

を求める。

(2) 使用法

CALL GASNS/D (A, B, FUN, N, Y, ICON)

CALL GSLNS/D (FUN, N, Y, ICON)

CALL GSHNS/D (FUN, N, Y, ICON)

引数	型と種類*	属性	内容
A	実数型	入力	定積分の下限。
B	実数型	入力	定積分の上限。
FUN	実数型 関数副プログラム	入力	被積分関数の名前。これに対する実引数としての関数は、使用者が積分変数だけの1変数の関数副プログラムとして用意しなければならない。
N	整数型	入力	分点数。 GASNS, GSLNS, GSHNS…… $1 \leq N \leq 20$ 。 GASND…… $1 \leq N \leq 50$ 。 GSLND, GSHND…… $1 \leq N \leq 38$ 。

引数	型と種類*	属性	内容
Y	実数型	出力	定積分の値が出力される。
ICON	整数型	出力	ICON = 0: 正常終了。 ICON = 30000: 入力引数Nの制限が破られた。

* 倍精度用のサブルーチンの場合には、実数型をすべて倍精度実数型とする。

(3) 使用例

```

C***** EXAMPLE FOR GASNS *****
EXTERNAL FUN
A=0.0
B=1.0
E=EXACT(A,B)
DO 1000 N=1,30
CALL CLOCKM(ITA)
CALL GASNS(A,B,FUN,N,Y,ICON)
CALL CLOCKM(ITB)
ERR=E-Y
IT=ITB-ITA
WRITE(6,610) N,ICON,Y,ERR,IT
610 FORMAT(1H,'GASNS',13,' POINTS ICON =',16,E25.15,E10.3,15)
1000 CONTINUE
2000 STOP
END
FUNCTION FUN(X)
FUN=EXP(-X)
RETURN
END
FUNCTION EXACT(A,B)
EXACT=-EXP(-B)+EXP(-A)
RETURN
END

```

TGLEGS / D / Q

TGLAGS / D / Q

TGHERS / D / Q

TGLOBS / D

Tables of Weights and Sample Points by Gaussian Quadrature Formula
ガウスの数値積分公式による重率と分点の値

作 成	秦野 甯世	1980年8月
形 式	サブルーチン	言語 ; FORTRAN サイズ ; 35, 36, 36, 36, 37, 37, 45, 46, 46, 41, 42行

(1) 概 要

ガウスの数値積分公式による重率と分点の値を計算する。

1. TGLEGS/D/Q…ガウス・ルジャンドルの公式

$$\int_{-1}^1 f(x) dx = \sum_{k=1}^n W_k f(x_k) + E_n \quad (1)$$

における分点 x_k と重率 W_k を求める。

2. TGLAGS/D/Q…ガウス・ラゲールの公式

$$\int_0^{\infty} e^{-x} \cdot f(x) dx = \sum_{k=1}^n W_k f(x_k) + E_n \quad (2)$$

における分点 x_k と重率 W_k を求める。

3. TGHERS/D/Q…ガウス・エルミート公式

$$\int_{-\infty}^{\infty} e^{-x^2} \cdot f(x) dx = \sum_{k=1}^n W_k f(x_k) + E_n \quad (3)$$

における分点 x_k と重率 W_k を求める。

4. TGLOBS/D……ガウス・ロバット公式

$$\int_{-1}^1 f(x) dx = W_1 \cdot f(-1) + \sum_{k=2}^{n-1} W_k f(x_k) + W_n f(1) + E_n \quad (4)$$

における分点 x_k と重率 W_k を求める。

(2) 使用法

CALL TGLEGS/D/Q (N, X, W, EPS, ICON)

CALL TGLAGS/D/Q (N, X, W, EPS, ICON)

CALL TGHERS/D/Q (N, X, W, EPS, ICON)

CALL TGLOBS/D (N, X, W, EPS, ICON)

引 数	型 と 種 類*	属 性	内 容
N	整 数 型	入 力	分点数 n 。 TGLEGS/D/Q …… $2 \leq N \leq 37$ 。

引 数	型 と 種 類*	属 性	内 容
			TGLAGS/D/Q..... $1 \leq N \leq 39$ 。 TGHERS/D/Q..... $1 \leq N \leq 57$ 。 TGLOBBS/D..... $1 \leq N \leq 20$ 。
X	実 数 型 1 次 元 配 列	出 力	大きさN。分点 x_k の値が出力される ($k = 1, 2, \dots, n$)。
W	実 数 型 1 次 元 配 列	出 力	大きさN。重率 w_k の値が出力される ($k = 1, 2, \dots, n$)。
EPS	実 数 型	入 力	分点 x_k を求めるとき (例えば TGLEGS では、ルジャンドル多項式 $P_n(x)$ の零点) の Newton 法における収束判定定数。
ICON	整 数 型	出 力	ICON = 0: 正常終了。 ICON = 10000: EPS が小さすぎるので次の値に引上げて計算を行った。 単精度..... 10^{-6} , 倍精度..... 10^{-15} , 4 倍精度..... 10^{-32} 。 ICON = 30000: 入力引数Nの制限が破られた。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

参 考 文 献

- 1) 山内, 宇野, 一松; “電子計算機のための数値計算法Ⅲ”, 培風館, p. 279 (1971)。

AQNN5S / D

AQNN7S / D

AQNN9S / D

Adaptive Quadrature Based on Newton-Cotes 5 (7, 9) Point Rule

ニュートン・コーツ5(7, 9)点則に基づく適応型自動積分

作成	二宮 市三 1978年2月
形式	サブルーチン 言語; FORTRAN サイズ; 93, 94, 99, 100, 103, 104行

(1) 概要

被積分関数 $f(x)$, 下限 a , 上限 b , 要求精度 ϵ が与えられたとき, 適応型自動積分法を用いて, 定積分 $\int_a^b f(x) dx$ を絶対誤差 ϵ 以内の精度で計算する。適応型自動積分法というのは, 被積分関数の変化の緩急に順応して, サンプル値を取る密度を調節し, 合理的に計算を進める方法である。本ルーチンは, ニュートン・コーツ5(7, 9)点則に基づくものであるが, 従来からの同様なものに比べ, 誤差の推定法, 誤差の各小部分区間への配分, 不連続点, 代数的特異点の検出など, 多くの新しい工夫が加えられているので, 信頼性が高く, サンプル数も少ない。²⁾

(2) 使用法

CALL AQNN5S/D (A, B, FUNC, S, EPS, LF, NF, ILL)

CALL AQNN7S/D (A, B, FUNC, S, EPS, LF, NF, ILL)

CALL AQNN9S/D (A, B, FUNC, S, EPS, LF, NF, ILL)

引数	型と種類*	属性	内容
A	実数型	入力	定積分の下限。
B	実数型	入力	定積分の上限。A < B でなければならない。
FUNC	実数型 関数副プログラム	入力	被積分関数の名前。これに対する実引数としての関数は, 使用者が積分変数だけの1変数の関数副プログラムとして, 用意しなければならない。
S	実数型	出力	定積分の値が出力される。
EPS	実数型	入力	要求精度を表す正数。 単精度の場合 $10^{-3} \sim 10^{-6}$ 倍精度の場合 $10^{-5} \sim 10^{-15}$ の程度が標準である。
LF	整数型	入力	関数のサンプル回数の上限。LF > 12。数千程度が適当。
NF	整数型	出力	関数のサンプル回数。NF > LF となったとき, 計算を中断してルーチンより脱出する。
ILL	整数型	出力	ルーチン内の計算の状況を示す。ルーチン内で0にセットされ, 次の各状態が発生する度にそれぞれに応じて一定の数が加えられる。

引 数	型 と 種 類*	属 性	内 容
			(1) 小部分区間の長さが極度に小さくなったとき, 1 (2) 不連続点を検出したとき, 10 (3) 対数特異点を検出したとき, 10 (4) 代数特異点を検出したとき, 1000 (5) $NF > LF$ となったとき, 10000 (6) 代数特異点の次数が-1以下であるとき, 20000 (7) 入力に関する制限が破られたとき, 30000 以上のうち, (5), (6), (7)が発生すると, 計算は中断される。

* 倍精度用のサブルーチンの場合は, 実数型をすべて倍精度実数型とする。

(3) 性 能

被積分関数その変化の特徴によって, 次の五つの型に分類する。

- (1) 平滑型 関数が滑らかで, 変化がおだやかである。
- (2) ピーク型 急峻なピークや谷がある。
- (3) 振動型 波長の短かい, 激しい振動がある。
- (4) 不連続型 関数値や微係数に不連続点がある。
- (5) 特異型 対数特異点, 代数特異点がある。

本ルーチンは, その計算法の局所性により, 平滑型は勿論のこと, ピーク型に特に強い。また, 不連続型, 特異型も異常を示す点が積分区間の端や中央などの検出し易い場所にはこれを検出し, 効果的に処理される。振動型に対しては, 他の型に比較してやや苦手である。ともかく, どの型に対しても, 場合によっては被積分関数の評価回数が多くなることはあっても, 妥当な積分値を与えるという意味で, 頑丈であると言える。

3種類のルーチンの中では信頼性の高さから9点則の利用を推奨する。Kahaner の21個のテスト問題¹⁾に対する実験結果は, 次表のとおりである。ただし, 信頼度というのは, 定積分の計算値が実際に要求精度を満足した場合の比率である。

要求精度	10^{-3}		10^{-6}		10^{-9}	
	信頼度	平均 サンプル数	信頼度	平均 サンプル数	信頼度	平均 サンプル数
AQNN 5 D	95 %	61	90 %	106	90 %	346
AQNN 7 D	86	51	90	105	86	237
AQNN 9 D	95	82	95	123	90	234
QNC 7 *	86	79	86	201	81	437
QUAD **	95	149	90	269	86	465

* QNC7は, Kahaner による7点法に基づくサブルーチン。

** QUADは, Kahaner による10点法に基づくサブルーチン。

(4) 使用例

a の値を 0.1 から 0.1 刻みに 0.9 まで変えて, 定積分 $\int_0^1 (x^{-a} + 1 + \sin x) dx$ の値を AQNN9S で求める場合のプログラムの断片を次に示す。

```

C      MAIN PROGRAM
      COMMON A
      EXTERNAL FUN
      DO 10 I=1,9
      A=FLOAT(I)*0.1
      CALL AQNN9S(0.0,1.0,FUN,S,1.E-4,5000,NF,IND)
      :
10    CONTINUE
      STOP
      END

C      FUNCTION SUBPROGRAM FOR INTEGRAND
      FUNCTION FUN(X)
      COMMON A
      FUN=1.+SIN(X)
      IF(X.GT.0.) FUN=X**(-A)+FUN
      RETRUN
      END

```

この例のように、被積分関数に副変数がある場合（例ではA）には、これをコモン領域に入れて、メイン・プログラムと被積分関数副プログラムの間の連絡を行う。また、特異点で関数値が ∞ になる場合は、これを適当な有限値（例えば0）で置き換えることが本ルーチンを利用する場合には必要である。

(5) 備考

1. 本ルーチンの場合に限らないが、定積分を計算するには、必要があれば適当な変数変換により、積分区間の長さや積分値の絶対値とともに1の程度の数にすることが望ましい。被積分関数副プログラムの名前はメイン・プログラムで EXTERNAL 宣言しなければならない。
2. 特異点などはできる限り積分区間の端にくるように、しかも、なるべくこれを原点にとることが精度を高める上で望ましい。なお、使用例の説明を参照のこと。
3. 使用法の ILL の項にあるように、本ルーチンでの ILL の値は、 $ILL \geq 10000$ の場合以外は $ILL \neq 0$ でも必ずしも得られた積分値が無効ということはない。例えば、代数特異点を検出して、これに応ずる処置が取られた場合には $ILL = 1000$ となるが、積分値は正しいことが多い。このような場合、得られた積分値に自信が持てないときには、2種類のルーチンを用いて、それぞれの計算値を比較するの一法であろう。

参考文献

- 1) D.K. Kahaner; "Comparison of Numerical Quadrature Formulas," J.R. Rice, ed. : Mathematical Software, Academic Press, pp.229-259 (1971).
- 2) 二宮市三; "適応型ニュートン・コーツ積分法の改良", 情報処理, Vol.21, No.5, pp.504-512 (1980).
- 3) I. Ninomiya; "Improvement of Adaptive Newton-Cotes Quadrature Methods," JIP, Vol.3, No.3, pp.162-170, (1980).

DEFINS / D

IMTDES / D

Numerical Quadrature by Double Exponential Formulas —Finite Interval—

二重指数関数型公式による有限区間積分

作 成	秦野 甯世	1977 年 3 月
形 式	サブルーチン	言語 ; FORTRAN サイズ ; 264, 265, 136, 137 行

(1) 概 要

被積分関数 $f(x)$, 下限 a , 上限 b , 要求精度 ϵ が与えられたとき, 高橋・森の二重指数関数型積分公式¹⁾²⁾³⁾を用いて, 定積分 $\int_a^b f(x)dx$ を絶対誤差 ϵ 以内の精度で計算する自動積分ルーチンである。特に, 積分区間の端点で $x^{-\alpha}$ ($0 < \alpha < 1$) 型の特異点があってもわずかな計算量で高精度の結果が得られる。ただし, $f(x)$ は端点を除いて解析的であることを前提としている。

(2) 使用法

CALL DEFINS/D (A, B, F, S, EPS, N, ILL)

CALL IMTDES/D (A, B, F, S, EPS, N, ILL)

引 数	型 と 種 類*	属 性	内 容
A, B	実 数 型	入 力	定積分の上限, 下限。A \asymp B
F	実 数 型 関数副プログラム	入 力	被積分関数の名前。これに対する関数副プログラムは, 積分変数一つだけを引数に持つものを利用者が用意しなければならない。
S	実 数 型	出 力	定積分の値が出力される。ILL = 0 又は 30000 以外のときは最後に得られた近似値が出力されている。
EPS	実 数 型	入 力	要求精度を表す正数(ϵ)。単精度では 10^{-5} , 倍精度では 10^{-10} 程度が標準である。
N	整 数 型	出 力	関数の実際の評価回数。
ILL	整 数 型	出 力	ルーチン内での計算の状況を示す。 DEFINS/D の場合: ルーチン内で最初 0 にセットされ, 次の各状態が発生する度にそれぞれに応じて一定の値が加えられる。 (1) 区間の下限の方で, 関数値が急激に増大しているため要求精度が自動的に引下げられたとき, 1 (2) 区間の上限の方で, (1)の事象が発生したとき, 2 (3) ルーチンの許容標本点数の最大を用いても収束しないとき, 10000 (4) 入力引数に対する制限が破られたとき, 30000 IMTDES/D の場合: ILL = 0 のとき正常終了, ILL = 10000 のときルーチンの許容標本点数の最大を用いても収束しない, ILL = 30000 のとき入力引数に対する制限が破られたため計算を全く行わない。

* 倍精度用ルーチンの場合には, 実数型をすべて倍精度実数型とする。

(3) 計算法

定積分 $I = \int_{-1}^1 f(x) dx$ において, x に変換

$$x = \phi(t)$$

を行うと定積分 I は

$$I = \int_{t_0}^{t_n} f(\phi(t))\phi'(t) dt$$

と書ける。これに台形則を用いて I を求める。変換式は, DEFINS (D) の場合は⁽²⁾

$$x = \tanh\left(\frac{\pi}{2} \sinh t\right), -1 \leq x \leq 1, -\infty \leq t \leq \infty$$

を, IMTDES (D) の場合は³⁾

$$x = \tanh\left\{\frac{\pi}{2} \sinh \frac{\pi}{2} \left(\frac{1}{1-t} - \frac{1}{1+t}\right)\right\}, -1 \leq x, t \leq 1$$

を用いる。

(4) 性能

関数の評価回数が少ないのが特長である。被積分関数の変化の仕方が滑らかでおだやかなものや比較的ゆるやかな振動をするものについては効率が良い。特に端点で $x^{-\alpha}$ ($0 < \alpha < 1$) の特異性をもっているものに対しては, 他のルーチンにみられない高効率を示す。ただし, この場合, 10桁以上の精度を得ることが難しいこともある。区間の中央部に高いピークを持つもの, 不連続点を持つものに対しては不得意である(参考文献⁵⁾の表1 (p. 210)を参照)。

次表に Kahaner の21個のテスト問題⁴⁾に対する実験結果を示す。ただし, 信頼度というのは定積分の計算値が実際に要求精度を満足した場合の比率である。

要求精度	10 ⁻³		10 ⁻⁶		10 ⁻⁹	
	信頼度	平均評価回数	信頼度	平均評価回数	信頼度	平均評価回数
DEFIND	95%	81	90%	114	86%	138
IMTDED	86	59	90	113	81	131

(5) 使用例

```
C.... EXAMPLE OF DEFINS....
EXTERNAL FUN
A=-1.0
B=1.0
EPS=1.0E-4
CALL DEFINS(A,B,FUN,S,EPS,NF,ILL)
WRITE(6,610) ILL,S,NF
10 CONTINUE
STOP
610 FORMAT(1H ,10X,5HILL= ,15,3X,2HS=,E22.14,
*3X,2HN=,15)
END
FUNCTION FUN(X)
P=(1.0+X)*(1.0-X)
FUN=0.0
IF(P.GT.0.0) FUN=1.0/SQRT(P)
RETURN
END
```

参考文献

- 1) H. Takahashi and M. Mori; "Double Exponential Formulas for Numerical Integration," *Bull. R.I.M.S., Kyoto Univ.*, **9**, pp.721-741 (1974).
- 2) 森 正武; "曲線と曲面", p.24, 教育出版社 (1974)。
- 3) 森 正武; "IMT 型二重指数関数型数値積分公式について", 数値計算のアルゴリズムの研究会予稿集, 京都大学数理解析研究所 (1976)。
- 4) D.K. Kahaner; "Comparison of Numerical Quadrature Formulas," J.R. Rice, ed., *Mathematical Software*, Academic Press, pp.229-259 (1971).
- 5) 二宮市三, 桑野甯世; "新登録 SSL プログラム", 名古屋大学大型計算機センターニュース, Vol. 8, No. 3, pp.209-263 (1977)。

QDAPBS / D

A Quadrature of Interpolatory Type Increasing the Sample Points with Arithmetic Progression

等差数列的に標本点を増す補間型積分法

作成	長谷川 武光 1977年4月
形式	サブルーチン 言語 ; FORTRAN サイズ ; 142, 302行

(1) 概要

被積分関数 $f(x)$ 、積分区間の下限 a 、上限 b が与えられたとき、定積分 $\int_a^b f(x)dx$ をセンターの計算機で得られる限度の精度で計算する自動積分ルーチンである。本ルーチンは、標本点を等差数列的（8点ずつ）に増すので、標本を無駄にする可能性が少なく効率が良い。滑らかな被積分関数に対して精度の良い方法である。

(2) 使用法

CALL QDAPBS/D (A, B, F, S, EPS, N, ILL)

引数	型と種類*	属性	内容
A	実数型	入力	定積分の下限。
B	実数型	入力	定積分の上限。
F	実数型 関数副プログラム	入力	被積分関数の名前。これに対する実引数としての関数は、使用者が積分変数だけの1変数の関数副プログラムとして用意しなければならない。
S	実数型	出力	定積分の値が出力される。ILL = 10000 のときは、最後に得られた近似値が出力されている。
EPS	実数型	出力	積分値 S の誤差の推定値。
N	整数型	入出力	入力としては標本数の下限。出力としては実際に使用した標本数。入力として、N = 16 (QDAPBS) 又は N = 32 (QDAPBD) とするのが良い。
ILL	整数型	出力	ILL = 0 : 正常終了。 ILL = 10000 : 標本点を 200 点 (QDAPBS) 又は 512 点 (QDAPBD) 使用しても収束しないとき。 ILL = 30000 : B ≤ A のとき。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 計算法

積分区間(A, B)を[-1, 1]に変換し、この区間内にチェビシェフ分布するよう標本点を等差数列的に追加しながら補間式の列をつくり、これを積分する。

参考文献

- 1) 二宮市三, 桑野甯世; "新登録 SSL プログラム", 名古屋大学大型計算機センターニュース, Vol. 8, No. 3,

2) 鳥居達生,長谷川武光,二宮市三;“等差数列のに標本数を増す補間の自動積分法”『情報処理』,Vol.19, No.3, pp. 248—255 (1978)。

— Interval estimation of the variance of the normal distribution by the method of the automatic interpolation of the sample size.

鳥居達生,長谷川武光,二宮市三の論文(1978年)の要約。

鳥居達生,長谷川武光,二宮市三の論文(1978年)の要約。

— Interval estimation of the variance of the normal distribution by the method of the automatic interpolation of the sample size.

鳥居達生

鳥居達生,長谷川武光,二宮市三の論文(1978年)の要約。この論文は、正規分布の分散の区間推定に、標本数を自動的に増やす補間法を用いる。この方法は、標本数を増やすことで、分散の推定精度を向上させる。また、この方法は、標本数を増やすことで、分散の推定精度を向上させる。また、この方法は、標本数を増やすことで、分散の推定精度を向上させる。

鳥居達生

鳥居達生,長谷川武光,二宮市三の論文(1978年)の要約。

鳥居達生,長谷川武光,二宮市三の論文(1978年)の要約。

鳥居達生,長谷川武光,二宮市三の論文(1978年)の要約。この論文は、正規分布の分散の区間推定に、標本数を自動的に増やす補間法を用いる。この方法は、標本数を増やすことで、分散の推定精度を向上させる。また、この方法は、標本数を増やすことで、分散の推定精度を向上させる。また、この方法は、標本数を増やすことで、分散の推定精度を向上させる。

鳥居達生

鳥居達生,長谷川武光,二宮市三の論文(1978年)の要約。この論文は、正規分布の分散の区間推定に、標本数を自動的に増やす補間法を用いる。この方法は、標本数を増やすことで、分散の推定精度を向上させる。また、この方法は、標本数を増やすことで、分散の推定精度を向上させる。また、この方法は、標本数を増やすことで、分散の推定精度を向上させる。

HINFAS / D

HINFES / D

Numerical Quadrature by Double Exponential Formulas —Semiinfinite Interval— 二重指数関数型公式による半無限区間積分

作成	桑野 甯世	1977年5月
形式	サブルーチン	言語 ; FORTRAN サイズ ; 261, 262, 261, 262行

(1) 概要

被積分関数 $f(x)$ と要求精度 ϵ が与えられたとき、高橋・森の二重指数関数型積分公式¹⁾を用いて半無限区間定積分 $\int_0^{\infty} f(x)dx$ を絶対誤差 ϵ 以内の精度で計算する自動積分ルーチンである。特に HINFES (D) は、おとなしい関数 $g(x)$ を用いて $f(x) = g(x)e^{-x}$ の形で表される場合に適した公式を用いている。 $f(x)$ が $x \rightarrow \infty$ での 0 への収束が遅い場合、 $x = 0$ で $x^{-\alpha}$ ($0 < \alpha < 1$) 程度の特異性を持つ場合でも高精度の解が得られる。

(2) 使用法

CALL HINFAS/D (F, S, EPS, N, ILL)

CALL HINFES/D (F, S, EPS, N, ILL)

引数	型と種類*	属性	内容
F	実数型 関数副プログラム	入力	被積分関数の名前。これに対する副プログラムは、積分変数一つだけを引数に持つものを利用者が用意しなければならない。
S	実数型	出力	定積分の値が出力される。ILL = 0 又は 30000 以外のときは、最後に得られた近似値が出力されている。
EPS	実数型	入力	要求精度を表す正数 (ϵ)。単精度では 10^{-5} 、倍精度では 10^{-8} 程度が適当である。保存される。
N	整数型	出力	関数の実際の評価回数。
ILL	整数型	出力	ルーチン内の計算の状況を示す。ルーチン内で最初 0 にセットされ、次の各状態が発生する度にそれぞれに応じて一定の値が加えられる。 (1) $x \rightarrow 0$ の付近で、関数値が急激に増大しているため要求精度が自動的に引き下げられたとき、1 (2) $x \rightarrow \infty$ の方で、関数値の 0 への収束が遅いため要求精度が自動的に引き下げられたとき、2 (3) ルーチンの許容標本点数の最大を用いても収束しないとき、10000 (4) EPS < 0 を指定したとき、30000

* 倍精度用ルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 計算法

定積分 $\int_0^{\infty} f(x)dx$ において積分変数 x を次のように変換したものに、台形則を用いる。HINFAS(D)

の場合¹⁾は,

$$x = \exp\left(\frac{\pi}{2} \sinh t\right), \quad -\infty \leq t \leq \infty$$

を, HINFES(D)の場合¹⁾は,

$$x = \exp\left\{\frac{\pi}{2}(t - \exp(-t))\right\}, \quad -\infty \leq t \leq \infty$$

を用いる。

(4) 性能

$x \rightarrow \infty$ での関数値の0への収束が遅いものや Gauss-Laguerre の公式でうまく求まらないようなものに対しても, かなり良い結果が得られる。ただし, そのようなものに対して10桁以上の精度を得ることは難しい(参考文献²⁾の表1(p. 210)を参照)。

参 考 文 献

- 1) 森 正武; “曲線と曲面”, p.24, 教育出版社(1974)。
- 2) 二宮市三, 桑野甯世; “新登録 SSL プログラム”, 名古屋大学大型計算機センターニュース, Vol.8, No.3, pp. 209—263 (1977)。

INFINS / D

Numerical Quadrature by Double Exponential Formulas —Infinite Interval— 二重指数関数型公式による無限区間積分

作 成	秦野 甯世	1977年4月
形 式	サブルーチン 言語; FORTRAN サイズ; 250, 251行	

(1) 概 要

被積分関数 $f(x)$ と要求精度 ϵ が与えられたとき、高橋・森の二重指数関数型積分公式¹⁾を用いて無限区間定積分 $\int_{-\infty}^{\infty} f(x) dx$ を絶対誤差 ϵ 以内の精度で計算する自動積分ルーチンである。 $f(x)$ が $x \rightarrow \pm \infty$ で 0 への収束が遅い場合でも高精度の解が得られる。

(2) 使用法

CALL INFINS/D (F, S, EPS, N, ILL)

引 数	型 と 種 類*	属 性	内 容
F	実 数 型 関数副プログラム	入 力	被積分関数の名前。これに対する関数副プログラムは、積分変数一つだけを引数に持つものを利用者が用意しなければならない。
S	実 数 型	出 力	定積分の値が出力される。ILL=0 又は30000以外のときは、最後に得られた近似値が出力されている。
EPS	実 数 型	入 力	要求精度を表す正数(ϵ)。単精度では 10^{-5} 、倍精度では 10^{-8} 程度が適当である。
N	整 数 型	出 力	関数の実際の評価回数。
ILL	整 数 型	出 力	ルーチン内の計算の状況を示す。ルーチン内で最初0にセットされ、次の各状態が発生する度にそれぞれに応じて一定の値が加えられる。 (1) $x \rightarrow -\infty$ の方で、関数値の0への収束が遅いため要求精度が自動的に引き下げられたとき, 1 (2) $x \rightarrow +\infty$ の方で、(1)の状態が発生したとき, 2 (3) ルーチンの許容標本点数の最大を用いても収束しないとき, 10000 (4) EPS < 0 を指定したとき, 30000

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 計算法

定積分 $\int_{-\infty}^{\infty} f(x) dx$ において、積分変数 x を次のように変換したものに台形則を用いる¹⁾。

$$x = \sinh\left(\frac{\pi}{2} \sinh t\right), \quad -\infty \leq x, t \leq \infty$$

(4) 備考

$x \rightarrow \pm \infty$ での関数値の 0 への収束が遅いものや Gauss-Hermite の公式でうまく求まらないようなものに対しても、かなり良い結果が得られる。ただし、原点付近で高いピークを持つもの、激しく振動するものに対しては不得意である (参考文献²⁾ の表 1 (p. 210) を参照)。

参考文献

- 1) 森 正武; “曲線と曲面”, p.24, 教育出版社 (1974)。
- 2) 二宮市三, 秦野甯世; “新登録 SSL プログラム”, 名古屋大学大型計算機センターニュース, Vol.8, No.3, pp. 209-263 (1977)。

TRAPZS / D

Numerical Quadrature by Trapezoidal Rule —Infinite Interval—

台形則による無限区間積分

作成	秦野 甯世	1977年4月
形式	サブルーチン 言語 ; FORTRAN サイズ ; 81, 82行	

(1) 概要

被積分関数 $f(x)$ と要求精度 ϵ が与えられたとき、台形則により定積分 $\int_{-\infty}^{\infty} f(x) dx$ を絶対誤差 ϵ 以内の精度で計算する自動積分ルーチンである。 $f(x)$ が $x \rightarrow \pm \infty$ で 0 への収束が速い場合に有効である。

(2) 使用法

CALL TRAPZS/D (F, S, H, EPS, N, MAXF, ILL)

引数	型と種類*	属性	内容
F	実数型 関数副プログラム	入力	被積分関数の名前。これに対する関数副プログラムは、積分変数一つだけを引数に持つものを、利用者が用意しなければならない。
S	実数型	出力	定積分値が出力される。ILL=0 又は 30000 以外のときは、最後に得られた近似値が出力されている。
H	実数型	入出力	入力としてはきざみ幅の初期値。計算の途中では半分ずつに減少させて行き、最終段階でのきざみ幅が出力される。 H>0
EPS	実数型	入力	要求精度を表す正数(ϵ)。単精度では 10^{-5} 、倍精度では 10^{-15} 程度が最小限である。
N	整数型	出力	関数の実際の評価回数。
MAXF	整数型	入力	関数の評価回数の上限。 $5 \leq \text{MAXF}$
ILL	整数型	出力	ルーチン内の計算の状況を示す。ルーチン内で 0 にセットされ、次の各状態が発生する度にそれぞれに応じて一定の値が加えられる。 (1) $x \rightarrow -\infty$ の方で、関数値の 0 への収束が遅くて MAXF 以内で要求精度が得られないため、要求精度を自動的に大きくしたとき、1 (2) $x \rightarrow +\infty$ の方で、(1)の状態が発生したとき、2 (3) $N < \text{MAXF}$ になっても収束しないとき、10000 (4) 入力引数に対する制限が破られているとき、30000

* 倍精度用のサブルーチンの場合、実数型をすべて倍精度実数型とする。

(3) 性能

$f(x)$ が $x \rightarrow \pm \infty$ での 0 への収束が速い場合には、Gauss-Hermite の公式や二重指数関数型公式でうまく解が求まらない場合でも、有効である(参考文献¹⁾の表 1 (p. 210)を参照)。

(4) 備考

UNIONION

ILL = 1 又は 2 で終了したときには、きざみ幅の初期値 H を大きくするとよい。

UNIONION

参考文献

UNIONION

- 1) 二宮市三, 秦野甯世; "新登録 SSL プログラム", 名古屋大学大型計算機センターニュース, Vol.8, No.3, pp. 209-263 (1977)。

UNIONION

ROMBGS / D

Romberg Quadrature

ロンバーグ積分

作成	二宮 市三 1977年4月
形式	サブルーチン 言語; FORTRAN サイズ; 30, 31行

(1) 概要

古典的ロンバーグ積分法による自動数値積分ルーチンである。すなわち、積分区間の下端 a 、上端 b 、被積分関数 $f(x)$ 、収束判定常数 ϵ を与えて、定積分 $\int_a^b f(x)dx$ を ϵ の精度で計算する。

(2) 使用法

CALL ROMBGS/D (A, B, F, S, EPS, ILL)

引数	型と種類*	属性	内容
A	実数型	入力	積分区間の下端。
B	実数型	入力	積分区間の上端。
F	実数型 関数副プログラム	入力	被積分関数の名称。これに対する実引数としての関数は、使用者が積分変数だけの1変数の関数副プログラムとして用意しなければならない。
S	実数型	出力	定積分の値が入る。ILL = 1 のときは最後に得られた近似値が入っている。
EPS	実数型	入力	収束判定常数。連続する二つの近似値の差の絶対値が EPS より小となったとき、収束したものと判定する。EPS > 0
ILL	整数型	出力	ILL = 0: 正常終了。 ILL = 30000: EPS ≤ 0。 ILL = 1: 積分区間を ROMBGS では8192分割, ROMBGD では16384分割してもなお収束しないとき。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 使用例

副変数を含む関数の積分の例として

$$J_0(x) = \frac{1}{\pi} \int_0^\pi \cos(x \sin \theta) d\theta$$

を考えてみよう。副変数 x を 0.1 から 0.1 刻みに 5.0 まで動かして $J_0(x)$ を求めるには、次のようにすればよい。

```
C      MAIN PROGRAM
C      COMMON X
C      EXTERNAL BES
C      PI=3.141593
C      X=0.0
C      DO 1 J=1, 50
C      X=X+0.1
```

```

CALL R0MBGS(0.0,PI,BES,S,1.0E-6,ILL)
B=S/PI
:
1 CONTINUE
:
END

C   FUNCTION SUBPROGRAM FOR INTEGRAND
FUNCTION BES(THETA)
COMMON X
BES=COS(X*SIN(THETA))
RETURN
END

```

(4) 備考

本ルーチンは、被積分関数が穏やかな場合だけに適する。被積分関数が激しい変化をする場合には、適応型自動積分ルーチンなど他のルーチンを用いる方がよい。

MQPRRS / D

Multiple Quadrature by Product Rules

直積型公式による多重数値積分

作 成	二宮 市三 1979年3月
形 式	サブルーチン 言語; FORTRAN サイズ; 79, 80行

(1) 概 要

次元 n ($1 \leq n \leq 10$), 下限 $a_1, \dots, a_2, \dots, a_n$, 上限 b_1, b_2, \dots, b_n , 被積分関数 $f(x_1, x_2, \dots, x_n)$ が与えられたとき, n 次元多重定積分

$$\int_{a_1}^{b_1} w_1(x_1) dx_1 \int_{a_2}^{b_2} w_2(x_2) dx_2 \cdots \int_{a_n}^{b_n} w_n(x_n) dx_n \cdot f(x_1, x_2, \dots, x_n)$$

の値を, 種々の1次元の積分公式の直積公式を用いて計算する。1次元の積分公式としては,

- (1) ニュートン・コーツ則 ($w(x) = 1$)
- (2) ガウス・ロバットー則 ($w(x) = 1$)
- (3) ガウス・ルジャンドル則 ($w(x) = 1$)
- (4) ガウス・ラゲール則 ($w(x) = e^{-x}, a = 0, b = \infty$)
- (5) ガウス・エルミート則 ($w(x) = e^{-x^2}, a = -\infty, b = \infty$)

が用意されている。

(2) 使用法

CALL MQPRRS/D (N, A, B, FUN, MET, NPT, NDV, S, P, W, ISW, ILL)

引 数	型 と 種 類*	属 性	内 容
N	整 数 型	入 力	積分の多重度。 $1 \leq N \leq 10$
A	実 数 型 1 次 元 配 列	入 力	積分領域の下限を表す。無限積分に関する成分は任意。
B	実 数 型 1 次 元 配 列	入 力	積分領域の上限を表す。無限積分に関する成分は任意。
FUN	実 数 型 関数副プログラム	入 力	被積分関数。これに対する実引数としての関数は, 使用者が積分変数だけを変数とする関数副プログラムとして用意しなければならない。
MET	整 数 型 1 次 元 配 列	入 力	各々の座標軸方向に用いる積分法を表す。 MET = 1 ニュートン・コーツ則 MET = 2 ガウス・ロバットー則 MET = 3 ガウス・ルジャンドル則 MET = 4 ガウス・ラゲール則 MET = 5 ガウス・エルミート則
NPT	整 数 型 1 次 元 配 列	入 力	各々の座標軸方向に用いる積分法の分点数を表す。 $1 \leq NPT(I) \leq 20$ 。ただし, ニュートン・コーツ則に対しては, $1 \leq NPT(I) \leq 11$ 。

引 数	型 と 種 類*	属 性	内 容
NDV	整 数 型 1 次 元 配 列	入 力	各々の座標軸方向の辺の等分割数。1 ≤ NDV (1)。無限積分に関する成分は任意。
S	実 数 型	出 力	積分の近似値。
P	実 数 型 1 次 元 配 列	作 業 領 域	各座標軸方向の総分点数 (分点数×分割数) の和以上の大きさの1次元配列。
W	実 数 型 1 次 元 配 列	作 業 領 域	大きさは P に同じ。
ISW	整 数 型	入 力	ISW = 0 のとき分点と重率の計算を行う。ISW ≠ 0 のとき分点と重率の計算を省略し、直前のコールでの分点と重率を再利用する。
ILL	整 数 型	出 力	ILL = 0: 正常終了。 ILL = 30000: N に関する制限が破られた。 ILL = K: K 番目の軸方向に関する引数が制限を破った。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 使用例

補助変数 q の値を変えて、3重積分

$$\int_0^{\infty} e^{-x} dx \int_{-1}^1 dy \int_0^1 dz \cdot x^2 \sin(q+y)/(1+z^2)$$

の値を、 x 方向にはガウス・ラゲール10点則、 y 方向にはニュートン・コーツ3点則 (シンプソン則)、 z 方向にはガウス・ルジャンドル5点則を用いて計算する。ただし、 y 方向には区間 $[-1, 1]$ を10等分し、 z 方向には区間 $[0, 1]$ を2等分するものとする。

```

C      MAIN PROGRAM
1      DIMENSION A(3),B(3),MET(3),NPT(3),NDV(3),P(100),W(100)
2      EXTERNAL FUN
3      COMMON Q
4      N=3
5      A(2)=-1.0
6      B(2)=1.0
7      A(3)=0.0
8      B(3)=1.0
9      MET(1)=4
10     MET(2)=1
11     MET(3)=2
12     NPT(1)=10
13     NPT(2)=3
14     NPT(3)=5
15     NDV(2)=10
16     NDV(3)=2
17     DO 10 IQ=1,50
18     Q=FLOAT(IQ)/10.0
19     ISW=IQ-1
20     CALL MQPRRS(N,A,B,FUN,MET,NPT,NDV,S,P,W,ISW,ICON)
21     10 WRITE(6,610) Q,S,ICON
22     610 FORMAT(1H ,3H0 =,F5.1,2X,3HS =,E13.5,2X,6HICON =,16)
23     STOP
24     END

1      FUNCTION FUN(X)
2      COMMON Q
3      DIMENSION X(3)

```

```

4      FUN=X(1)**2*SIN(X(2)+Q)/(X(3)**2+1.0)
5      RETURN
6      END

```

この例のように、被積分関数に副変数がある場合（例では Q）には、これをコモン領域に入れて、主プログラムと被積分関数副プログラムとの間の連絡を行う。また、本例のように、同じ領域で同じ積分公式を繰り返し用いる場合には、ISW の機能を用いて、2 回目の引用以後は分点位置と重率の計算を省略するのがよい。

(4) 備考

1. 被積分関数副プログラムの名前は主プログラムで、EXTERNAL 宣言しなければならない。
2. ガウス・ラゲール則とガウス・エルミート則の場合、A, B, NDV は任意でよいので、対応する成分を定義する必要はない。
3. ニュートン・コーツ則では分点数は11まで、他の公式では20まで用意されているが、むやみに分点数の多い公式を用いるよりは、区間をいくつかに分け、それぞれの小区間で比較的点数の少ない公式を複合的に用いる方が一般的に無難である。
4. 本サブルーチンでは分点位置と重率を計算するために次のようなスレーブ・サブルーチンを引用している。
 - (1) ニュートン・コーツ則 (TNCOTS/D)
 - (2) ガウス・ロバットー則 (TGLOBS/D)
 - (3) ガウス・ルジャンドル則 (TGLEGS/D)
 - (4) ガウス・ラゲール則 (TGLAGS/D)
 - (5) ガウス・エルミート則 (TGHERS/D)

MQNCDS / D

Multiple Quadrature by Product of Newton-Cotes Rules (Data Input)

ニュートン・コーツ則の直積による多重数値積分（データ入力）

作成	二宮 市三 1981年4月
形式	サブルーチン 言語 ; FORTRAN サイズ ; 60, 61行

(1) 概要

n 次元 ($1 \leq n \leq 10$) の超直方体領域 ($a_i \leq x_i \leq b_i, i = 1, \dots, n$) の等間隔格子点の上で、被積分関数 f の値がデータとして与えられたとき、 n 次元多重積分

$$\int_{a_1}^{b_1} dx_1 \int_{a_2}^{b_2} dx_2 \cdots \int_{a_n}^{b_n} dx_n f(x_1, x_2, \dots, x_n)$$

の値を、複合ニュートン・コーツ則の直積によって計算する。MQNCDS(D)は単(倍)精度用である。

(2) 使用法

CALL MQNCDS/D (F, N, NC, NP, H, S, ILL)

引数	型と種類*	属性	内容
F	実数型 N次元配列	入力	被積分関数の格子点上の値を入力する。Fの配列宣言における各添字の値は、超直方体の対応する座標軸方向の分点数(分割数+1)に正確に等しくなければならない。
N	整数型	入力	積分の多重度。 $1 \leq N \leq 10$
NC	整数型 1次元配列	入力	各座標軸方向の辺の分割数。 $1 \leq NC(I), I = 1, \dots, N$
NP	整数型 1次元配列	入力	各座標軸方向に用いるニュートン・コーツ則の分点数。 $1 \leq NP(I) \leq 11, I = 1, \dots, N$ 。 $NC(I) \cdot (NP(I) - 1)$ が各座標軸方向の分割数になるようにする(使用例参照)。
H	実数型 1次元配列	入力	各座標軸方向の格子間隔。
S	実数型	出力	積分の近似値。
ILL	整数型	出力	ILL = 0 : 正常終了。 ILL = K : K番目の軸方向に関する入力引数が制限を犯した。 ILL = 30000 : $N \leq 0$ 又は $N > 10$ であった。

* 倍精度用のサブルーチンの場合、実数型をすべて倍精度実数型とする。

(3) 使用例

2次元正方形領域 $0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1$ を各軸方向に20等分し、各格子点上で関数 $f(x_1, x_2) = e^{x_2} \sin x_1$ の値をデータとして与え、これを本サブルーチンに入力して積分

$$\int_0^1 \int_0^1 f(x_1, x_2) dx_1 dx_2$$

を求める。各軸方向をそれぞれ独立に、2分割及び4分割し、それに応じてニュートン・コーツ11点則及び6点則を用いる。プログラム及び出力は次のとおりである。

```

00010 C *** EXAMPLE FOR MQNCDS ***
00020     GENERIC
00040     DIMENSION NC(2),NP(2),H(2),A(21,21)
00050     H(1)=0.05
00060     H(2)=0.05
00065     WRITE(6,620)
00070     DO 20 I=1,21
00080     F=SIN(FLOAT(I-1)/20.E0)
00090     DO 20 J=1,21
00095     20 A(I,J)=EXP(FLOAT(J-1)/20.E0)*F
00098     T=(1.-COS(1.E0))*(EXP(1.E0)-1.)
00100     DO 30 I1=2,4,2
00110     NC(1)=I1
00120     NP(1)=20/I1+1
00130     DO 30 I2=2,4,2
00140     NC(2)=I2
00150     NP(2)=20/I2+1
00160     CALL MQNCDS(A,2,NC,NP,H,S,ILL)
00180     E=S-T
00187     620 FORMAT(9X,4H ILL,7X,1HS,14X,1HT,14X,1HE)
00190     WRITE(6,610) I1,I2,ILL,S,T,E
00200     610 FORMAT(3I4,3E15.5)
00210     30 CONTINUE
00220     STOP
00230     END

```

		ILL	S	T	E
2	2	0	0.78989E+00	0.78989E+00	-0.23842E-05
2	4	0	0.78989E+00	0.78989E+00	-0.30398E-05
4	2	0	0.78989E+00	0.78989E+00	-0.30398E-05
4	4	0	0.78989E+00	0.78989E+00	-0.30398E-05

(4) 備考

分点数11までの公式が用意されているがむやみに分点数の多い公式を用いるよりは、各辺をいくつかに等分し (NC の値を大きくして)、比較的的分点数の少ない公式を複合的に用いる方が、一般に無難である。

MQFSRS / D

Multiple Quadrature by Fully Symmetric Rules

完全対称則による多重数値積分

作成	二宮 市三	1981年4月
形式	サブルーチン 言語 ; FORTRAN サイズ ; 250, 250 行	

(1) 概要

n 次元 ($1 \leq n \leq 50$) の超直方体 ($a_i \leq x_i \leq b_i, i = 1, \dots, n$) における多重積分

$$\int_{a_1}^{b_1} dx_1 \int_{a_2}^{b_2} dx_2 \cdots \int_{a_n}^{b_n} dx_n f(x_1, x_2, \dots, x_n)$$

の値を 3 次, 5 次, 7 次又は 9 次の完全対称則によって計算する。MQFSRS(D) は単 (倍) 精度用である。

(2) 使用法

CALL MQFSRS/D (N, A, B, FUN, MET, ND, S, ILL)

引 数	型 と 種 類*	属 性	内 容
N	整 数 型	入 力	積分の多重度。 $1 \leq N \leq 50$
A	実 数 型 1 次 元 配 列	入 力	積分領域の下限。
B	実 数 型 1 次 元 配 列	入 力	積分領域の上限。
FUN	実 数 型 関数副プログラム	入 力	被積分関数。これに対する実引数は、利用者が積分変数だけを引数とする関数副プログラムとして用意しなければならない。
MET	整 数 型 1 次 元 配 列	入 力	2 個の要素をもつ 1 次元配列。MET(1)は完全対称則の次数を表し、3, 5, 7, 9 のいずれかでなければならない。MET(2)は 7 次と 9 次のとみだけ用いられる。 その値は 1 又は 2 であって次の意味をもつ。 MET(2) = 1: 領域の端に近い分点を用いる。 MET(2) = 2: 領域の中央に近い分点を用いる。
ND	整 数 型 1 次 元 配 列	入 力	各座標軸方向の辺の等分割数。 $1 \leq ND(I), I = 1, \dots, N$
S	実 数 型	出 力	積分の近似値。
ILL	整 数 型	出 力	ILL = 0: 正常終了。 ILL = K: $ND(K) \leq 0$ であった。 ILL = 30000: N と MET が制限を犯した。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 計算法

簡単のために領域を区間 $[-1, 1]$ の直積 $[-1, 1]^n$ としよう。完全対称則は原点に対して完全対称

に（軸の交換及び軸方向の反転に関して）配置された、小数個の非零座軸成分をもつ一群の点を分点とする、非直積型の多重数値積分則である。各分点での関数値に乗ぜられる重率は、積分則が最も高い次数の多項式に対して正確になるように決められている。本サブルーチンで使用可能な積分則は3次、5次、7次及び9次のものである。完全対称則は精度はあまり良くないが、次元 n の増加に伴う分点数の増加が比較的緩慢であるから、かなりの高次元の場合にも適用可能である。分点数 F は次元 n によって次のように与えられる。

$$3 \text{ 次: } F = 2n+1 \quad 7 \text{ 次: } F = \frac{4}{3}n(n^2+2)+1$$

$$5 \text{ 次: } F = 2n^2+1 \quad 9 \text{ 次: } F = \frac{2}{3}n(n-2)(n^2-1)+4n(2n-1)+1$$

$N = 3 \sim 20$ に対する F の値は次表のようになる。

N	3	5	7	9
3	7	19	45	77
4	9	33	97	193
5	11	51	181	421
6	13	73	305	825
7	15	99	477	1485
8	17	129	705	2497
9	19	163	997	3973
10	21	201	1361	6041
11	23	243	1805	8845
12	25	289	2337	12545
13	27	339	2965	17317
14	29	393	3697	23353
15	31	451	4541	30861
16	33	513	5505	40065
17	35	579	6597	51205
18	37	649	7825	64537
19	39	723	9197	80333
20	41	801	10721	98881

重率は次元数 n に依存し、しかも n が増大するにつれて大きく振動する（次数が大きい程この傾向が大きい）ようになる。したがって完全対称則では良い精度は期待できない。高次元の滑らかな関数の低精度の積分に適するといえよう。

(4) 使用例

6次元空間の領域 $[-1, 1]^6$ での積分

$$\frac{1}{64} \int_{-1}^1 \int_{-1}^1 \cdots \int_{-1}^1 \cos \left(3(1-x_1)x_2x_3x_4x_5x_6 + \frac{1}{2} \right) dx_1 dx_2 \cdots dx_6$$

を計算する。領域を各座標軸方向に2分割して生ずる64個の小領域の各々に完全対称則を適用する。プログラム及び出力は次のとおりである。

```

C      EXAMPLE FOR MQFSRS
      DIMENSION A(6),B(6),MET(2),ND(6)
      EXTERNAL FUN
      N=6
      MET(2)=1
      EXACT=0.8585247
      DO 10 I=1,N
      ND(I)=2
      A(I)=-1.0

```

```

10 B(I)=1.0
   WRITE(6,600)
600 FORMAT(1H0,'TEST FOR MQFSRS'//
*1H , 'N ORDER',8X,'EXACT',7X,'RESURT'
*,3X,'ABS ERR',3X,'REL ERR'//)
   DO 20 J=3,9,2
   MET(1)=J
   CALL MQFSRS(N,A,B,FUN,MET,ND,S,ILL)
   AER=S-EXACT
   RER=AER/EXACT
20 WRITE(6,610) N,J,EXACT,S,AER,RER
610 FORMAT(1H ,I1,I6,2E13.5,2E10.2)
   STOP
   END
   FUNCTION FUN(X)
   DIMENSION X(6)
   FUN=COS((1.0-X(1))*X(2)*X(3)*X(4)*X(5)*X(6)
**3.0+0.5)/64.0
   RETURN
   END

```

TEST FOR MQFSRS

	N	ORDER	EXACT	RESURT	ABS ERR	REL ERR
	6	3	0.85852E+00	0.86449E+00	0.60E-02	0.69E-02
	6	5	0.85852E+00	0.85897E+00	0.44E-03	0.52E-03
	6	7	0.85852E+00	0.85769E+00	-0.84E-03	-0.97E-03
	6	9	0.85852E+00	0.85800E+00	-0.52E-03	-0.61E-03

参 考 文 献

- 1) J. McNamee & F. Stenger; "Construction of Fully Symmetric Numerical Integration Formulas", Numer. Math., Bd.10, pp.327-344 (1967).

AQMDS / D

Automatic Multiple Integration Based on the Interpolatory Type Quadrature Increasing the Sample Points with Arithmetical Progression

等差数列的に標本数を増す補間型積分に基づく自動多重積分

作成	長谷川 武光 1980年4月
形式	サブルーチン 言語; FORTRAN サイズ; 562, 563行

(1) 概要

曲線境界領域での多重積分

$$I = \int_{\phi_1}^{\psi_1} dx_1 \int_{\phi_2}^{\psi_2} dx_2 \cdots \int_{\phi_m}^{\psi_m} dx_m f(x_1, x_2, \dots, x_m)$$

の近似値 S を、与えられた絶対誤差 ϵ_a と相対誤差 ϵ_r に対して

$$|S - I| \leq \max(\epsilon_a, \epsilon_r |I|)$$

を満たす精度で計算する自動積分ルーチンである。積分法は、等差数列的に標本数を増す補間型積分法(ここでは、積分区間の両端を含まない、いわゆる開いた公式を用いる。これに対して QDAPBS/D は閉じた公式である。開いた公式を用いたのは、いくらかでも両端が特異に近い関数にも対処するためである。)を各座標軸方向に繰り返し用いる直積型自動多重積分法である。滑らかな関数及び振動型の関数に対して特に有効な方法である。

(2) 使用法

CALL AQMDS/D (M, LSUB, FUN, EPSA, EPSR, NMIN, NMAX, S, ERR, N, ICON)

引数	型と種類*	属性	内容
M	整数型	入力	積分の多重度。1 ≤ M ≤ 3
LSUB	サブルーチン副プログラム	入力	積分の下限と上限を計算するサブルーチン副プログラムの名前。第1引数(Kとする)には、いま積分している座標軸 x_k 方向の番号 k が入る。第2引数(Xとする)は、M個の要素をもつ1次元配列名で、X(1)とX(2)にはそれぞれ x_1 と x_2 の値が入る。第3引数(Aとする)は積分の下限、第4引数(Bとする)は積分の上限が入る。主プログラムで EXTERNAL 宣言を要する。
FUN	実数型関数副プログラム	入力	被積分関数の名前。実引数(Xとする)としてM個の要素をもつ1次元配列一つだけの関数でなければならない。X(i)には x_i の値が入る(1 ≤ i ≤ M)。主プログラムで EXTERNAL 宣言を要する。
EPSA EPSR	実数型	入力	積分の近似値 S に対する要求絶対誤差 ϵ_a (EPSA)及び相対誤差 ϵ_r (EPSR)。EPSA ≥ 0, EPSR ≥ 0。
NMIN NMAX	整数型	入力	各座標軸方向での積分に対する被積分関数 FUN の評価回数の下限(NMIN)と上限(NMAX)。NMIN = 7, NMAX = 100 (AQMDSの場合), NMAX = 511 (AQMDDの場合)が適当。

引 数	型 と 種 類*	属 性	内 容
			NMAX ≥ 511 と指定したとき NMAX = 511 となる。
S	実 数 型	出 力	積分の近似値。
ERR	実 数 型	出 力	S に対する絶対誤差の推定値。
N	整 数 型	出 力	被積分関数 FUN の実際の全評価回数。
ICON	整 数 型	出 力	ICON = 0: 正常終了。 ICON = 30000: パラメータ・エラー。 各座標軸方向に積分するとき、関数評価回数が NMAX に達しても収束しない場合、その座標軸が x_3 なら ICON = 200, x_2 なら ICON = 2000, x_1 なら ICON = 20000。要求精度が小さすぎて、ある座標軸方向に積分をしてその積分の近似値の精度が、計算機の丸め誤差のレベルに達してしまったとき、その座標軸が x_3 なら ICON = 100, x_2 なら ICON = 1000, x_1 なら ICON = 10000。以上の事象が重なって起こった場合の ICON の値は、それぞれの ICON の値の和である。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 使用例

パラメータ p の値を変えて 3 重積分

$$\int_{-1}^1 dx_1 \int_{-2}^2 dx_2 \int_{-3}^3 dx_3 \frac{1}{48(4 - \cos px_1 - \cos px_2 - \cos px_3)}$$

を計算するプログラムを次に示す。

```

C      EXAMPLE ...AQMDS...
      EXTERNAL FUN,LSUB
      COMMON P
      EPSA=1.0E-4
      EPSR=0.0
      NMIN=7
      NMAX=100
      M=3
      DO 10 IP=1,10
      P=FLOAT(IP)*0.5
      CALL AQMDS(M,LSUB,FUN,EPSA,EPSR,NMIN,NMAX,S,ERR,N,ICON)
10     WRITE(6,100) P,S,ERR,N,ICON
100    FORMAT(1H,'P=',F4.1,5X,'S=',E15.7,5X,'ERR=',E10.2,5X,'N=',I7,5X,
*       'ICON=',I5)
      STOP
      END

C      FUNCTION FUN(X)
      DIMENSION X(3)
      COMMON P
      FUN=1.0/(4.0-COS(P*X(1))-COS(P*X(2))-COS(P*X(3)))/48.0
      RETURN
      END

C      SUBROUTINE LSUB(K,X,A,B)
      DIMENSION X(3)
      GO TO (1,2,3),K
1     A=-1.0
      B=1.0
      RETURN
2     A=-2.0
      B=2.0
      RETURN
3     A=-3.0

```

B=3.0
RETURN
END

この例のように、被積分関数にパラメータ（例では p ）がある場合には、これをコモン領域に入れて主プログラムとの連絡を図る。

(4) 性能

EPSA = 1.0 D-7 として、倍精度サブルーチン AQMDD を用いて、次の三通りの 3 重積分についてテストした結果を示す。

$$A \int_D \int_{i=1}^3 \frac{p}{x_i^2 + p^2} dx_1 dx_2 dx_3, \quad p = 1, \frac{1}{2}, \frac{1}{4}$$

$$B \int_D \int_{i=1}^3 \frac{1-p^2}{1-2px_i+p^2} dx_1 dx_2 dx_3, \quad p = \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$$

$$C \int_D \int_{i=1}^3 \cos px_i dx_1 dx_2 dx_3, \quad p = 8, 16, 32$$

ここで領域 D は $[-1, 1]^3$ である。

問 題	A			B			C		
p	1	1/2	1/4	1/4	1/2	3/4	8	16	32
標本数	12,167	59,487	350,847	11,215	29,791	223,543	29,663	65,151	272,199

各々の問題のパラメータ p の三通りの値は、積分が易しい、やや難しい、難しい、のそれぞれに対応している。

(5) 備考

1. 本ルーチンを何回もコールする場合、1次元用の積分の重みと標本点を初めの1回のみ計算するので、2回目以降コールする際には、計算時間が若干なくなる。
2. ICON = 0, 30000 以外のときは、積分の結果は要求精度を満足しないが、引数 ERR によって絶対誤差の推定ができる。
3. ICON = 200, 2000, 20000 のときは、NMAX を大きくすることにより要求精度に達することがある。
4. 多重積分の場合、一般に使用する標本数の数が大きくなり、したがって丸め誤差の累積が著しくなる。そこで、標準的には EPSA と EPSR が $0.5 E-4$ より小なら AQMDD を用いるとよい。NMAX ≥ 60 で AQMDS を用いて精度が十分得られないとき、AQMDD を用いると精度が良くなることがある。

参 考 文 献

- 1) 鳥居達生, 長谷川武光, 二宮市三; “等差数列的に標本数を増す補間的自動積分法”, 情報処理, Vol. 19, No. 3, pp. 248~255 (1978)。
- 2) 長谷川武光, 鳥居達生, 二宮市三; “標本数の節減を図る自動多重積分法”, 情報処理学会第21回全国大会予稿集, pp. 951 (1980)。
- 3) 二宮市三; “新登録数値計算ソフトウェアについて”, 名古屋大学大型計算機センターニュース, Vol. 10, No. 3,

pp. 278~308 (1979).

4) 長谷川武光; “等差数列的に標本数を増す補間型積分に基づく自動多重積分”, 名古屋大学大型計算機センター
ニュース, Vol.11, No. 4, p. 413 (1980).

© V. S. GOLUB
S. A. SOBEL
S. A. SOBEL
© V. S. GOLUB

Авторы благодарят за внимание к рукописи
рецензента.

Получено в редакцию 11.03.80. Принято в печать
14.04.80. Выходит в свет 15.05.80.

Продолжение

См. стр. 278

1980

$$\int_{-1}^1 \int_{-1}^1 \dots \int_{-1}^1 f(x_1, x_2, \dots, x_n) dx_1 dx_2 \dots dx_n$$

где $f(x_1, x_2, \dots, x_n)$ — функция, заданная на единичном кубе

$E_n = \{x_1, x_2, \dots, x_n \mid x_i \in [-1, 1], i=1, 2, \dots, n\}$

и n — натуральное число.

В работе рассматриваются алгоритмы вычисления интеграла (1) с помощью квадратурной формулы Гаусса по n переменным

$$I_n = \int_{-1}^1 \dots \int_{-1}^1 f(x_1, x_2, \dots, x_n) dx_1 dx_2 \dots dx_n$$

с использованием метода последовательного интегрирования по частям. В работе также рассматриваются алгоритмы вычисления интеграла (1) с помощью квадратурной формулы Гаусса по n переменным

$$I_n = \int_{-1}^1 \dots \int_{-1}^1 f(x_1, x_2, \dots, x_n) dx_1 dx_2 \dots dx_n$$

с использованием метода последовательного интегрирования по частям.

Ключевые слова: автоматическое интегрирование, квадратурная формула Гаусса.

Продолжение

См. стр. 278

См. стр. 278

Получено в редакцию 11.03.80. Принято в печать 14.04.80. Выходит в свет 15.05.80.

См. стр. 278

Получено в редакцию 11.03.80. Принято в печать 14.04.80. Выходит в свет 15.05.80.

См. стр. 278

Получено в редакцию 11.03.80. Принято в печать 14.04.80. Выходит в свет 15.05.80.

См. стр. 278

AQNDS / D

AQ3DS / D

AQ2DS / D

AQ1DS / D

Multiple Quadrature by Automatic Methods

自動多重数値積分

作成	二宮 市三, 長谷川 武光, 秦野 甯世 1979年3月
形式	サブルーチン 言語; FORTRAN サイズ; 622, 623行

(1) 概要

多重積分

$$I = \int_{l_1}^{u_1} dx_1 \cdots \int_{l_n}^{u_n} dx_n \cdot f(x_1 \cdots x_n) \quad (1)$$

を求めることを考える。ここで,

$$u_1 = a, u_2 = u_2(x_1), u_3 = u_3(x_1, x_2) \cdots \quad (2)$$

$$l_1 = b, l_2 = l_2(x_1), l_3 = l_3(x_1, x_2) \cdots$$

とする。今、積分 I の近似値 S に対する絶対誤差及び相対誤差の上限, ϵ_a, ϵ_r が与えられたとき,

$$|S - I| \leq \max(\epsilon_a, \epsilon_r |I|) \quad (3)$$

を満たす S を求める。積分法としては、1変数に対する各種の自動積分法を各次元方向に繰り返し用いる直積型多重積分法を用いる。1変数に対する自動積分公式としては、

- (1) ニュートン・コーツ9点則
- (2) クレンショウ・カーチス型積分法一分点を等比数列的(公比 $\sqrt{2}$)に追加する公式
- (3) 二重指数関数型積分公式(有限区間)
- (4) 二重指数関数型積分公式(半無限区間)
- (5) 二重指数関数型積分公式(全無限区間)

があり、任意の組み合わせが可能である。

(2) 使用法

CALL AQNDS/D (ME, M, AFUN, BFUN, FUN, EPSA, EPSR, NMIN, NMAX, S, ERR, N, ICON)

CALL AQ3DS/D (ME, AFUN, BFUN, FUN, EPSA, EPSR, NMIN, NMAX, S, ERR, N, ICON)

CALL AQ2DS/D (ME, AFUN, BFUN, FUN, EPSA, EPSR, NMIN, NMAX, S, ERR, N, ICON)

CALL AQ1DS/D (ME, AFUN, BFUN, FUN, EPSA, EPSR, NMIN, NMAX, S, ERR, N, ICON)

引数	型と種類*	属性	内容
ME	整数型 1次元配列	入力	M個の要素をもつ1次元配列。各々の次元方向に対して用いる積分公式を番号で指定する。 x_1 に対して ME(1)に、 x_n に対して ME(n) に指定する。保存される。 1 のとき ニュートン・コーツ9点則 2 のとき クレンショウ・カーチス型公式 3 のとき 二重指数関数型公式(有限区間) 4 のとき 二重指数関数型公式(半無限区間) 5 のとき 二重指数関数型公式(全無限区間) $1 \leq ME \leq 5$
M	整数型	入力	積分の多重度。AQ1DS/D のときは $M=1$ 、AQ2DS/D のときは $M=2$ 、AQ3DS/D のときは、 $M=3$ を指定したものとみなす。保存される。 $1 \leq M \leq 3$
AFUN BFUN	実数型 関数副プログラム	入力	定積分の下限(AFUN)、上限(BFUN)を計算する関数副プログラムの名前。各々は二つの引数をもつ。第1引数(Xとする)はM個の要素をもつ1次元配列名。X(1)には、 x_1 、X(2)には x_2 、X(3)には x_3 の値がセットされている。第2引数(Kとする)には、今、積分を実行している次元方向の番号がセットされている。すなわち x_1 についての積分を実行しているときは $K=1$ となっている。定積分の領域が積分変数の関数で定義される場合にはこれらの引数の値を引用して、AFUN、BFUN の値を定義する。いずれも、呼び出し側で EXTERNAL 宣言が必要。
FUN	実数型 関数副プログラム	入力	被積分関数 f を計算する関数副プログラム名。実引数(Xとする)として、M個の要素をもつ1次元配列一つだけの関数でなければならない。X(1)には x_1 、X(2)には x_2 の値がセットされている。呼び出し側で EXTERNAL 宣言が必要。
EPSA EPSR	実数型	入力	積分の近似値 S に対する絶対誤差の上限 ϵ_a 、相対誤差の上限 ϵ_r 。保存される。 $\epsilon_a, \epsilon_r \geq 0$
NMIN NMAX	整数型	入力	各次元方向ごとの f の評価回数下限、上限。保存される。NMIN = 10, NMAX = 100 が適当。ただし、各積分公式に対するコンポーネント・ルーチン内固有の値に反する場合は自動的に置きかえる。
S	実数型	出力	積分の近似値。
ERR	実数型	出力	S に対する絶対誤差の推定値。
N	整数型	出力	f の実際の評価回数。
ICON	整数型	出力	コンディション・コード。 x_1 に対する積分の終了状態は10000位に、 x_2 は100位、 x_3 は1位に表示される。各位について、次の各事象が発生する毎に一定の数が増えらる。 (1) 正常のとき 0 (2) NMAX を越えても収束しないとき 2 (3) (2)の事象が NMAX/10回を越えたとき 20 (4) (2)(3)以外の何らかの障害が発生したとき 1 (5) (4)の事象が NMAX/10回を越えたとき 10 以上の場合は S、ERR の値は各々近似値、誤差の推定値の意味をもつ。なお、 $MAX = \min(NMAX ** M, 1000000)$

引数	型と種類*	属性	内容
ICON	整数型	出力	として N がこれを超えたときは上記の値に5000を加える。 ICON = 30000のときはパラメータ・エラー。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 使用例

定積分

$$\int_0^1 dx_1 \int_0^{1-x_1} \frac{dx_2}{\sqrt{x_1+x_2}}$$

を x_2 について、二重指数関数型公式を x_1 についてクレンショウ・カーチス型公式を用いて、計算する場合のプログラムを次に示す。

```

C *** EXAMPLE (AQ2DS) ***
EXTERNAL FUN,AFUN,BFUN
DIMENSION ME(2)
ME(1)=2
ME(2)=3
CALL AQ2DS(ME,AFUN,BFUN,FUN,1.E-3,1.E-3,10,100,S,ERR,N,ICON)
WRITE(6,610) S,ERR,ICON
610 FORMAT(1H,'S =',F13.5,' ERR =',E10.2,' ICON =',I6)
STOP
END
FUNCTION AFUN(X,K)
AFUN=0.0
RETURN
END
FUNCTION BFUN(X,K)
DIMENSION X(2)
BFUN=1.0
IF(K.EQ.2) BFUN=BFUN-X(1)
RETURN
END
FUNCTION FUN(X)
DIMENSION X(2)
Y=ABS(X(1)+X(2))
IF(Y.LT.1.E-70) GO TO 2
FUN=1.0/SQRT(Y)
1 RETURN
2 FUN=0.0
GO TO 1
END

```

(4) 備考

1変数に対する自動積分公式及び対応するコンポーネント・サブルーチンについての詳細は参考文献^{1),2)}を参照されたい。積分公式の選択法についても参考文献^{1),4)}が役立つ。

参考文献

- 1) 二宮, 桑野; “新登録 SSL プログラム”, 名古屋大学大型計算機センターニュース, Vol. 8, No. 3, p. 209 (1977)。
- 2) 長谷川, 鳥居, 二宮; “等比数列的に標本数を増す Clenshaw-Curtis 型積分法”, 情報処理学会第19回大会予稿集, p. 391 (1978)。
- 3) 桑野, 長谷川, 二宮; “自動多重積分サブルーチンの作成”, 情報処理学会第20回全国大会予稿集, p. 447 (1979)。
- 4) 二宮, 桑野; “ライブラリー・プログラムの利用状況とプログラムの選択について”, 名古屋大学大型計算機センターニュース, Vol. 11, No. 4, p. 399 (1980)。

RK4S/D/Q/C/B

Solution of Initial Value Problems of Systems of Ordinary Differential Equations of The First Order by The Classic Runge-Kutta Method of The Fourth Order

4 次 の 古 典 的 ル ン ゲ ・ ク ッ タ 法 に よ る 連 立 1 階 常 微 分 方 程 式 の 初 期 値 問 題 の 数 値 解 法

作 成	二宮 市三 1979 年 3 月
形 式	サブルーチン 言語 ; FORTRAN サイズ ; 27, 28, 28, 28, 29 行

(1) 概 要

n 元連立 1 階常微分方程式

$$y'_i = f_i(x, y_1, y_2, \dots, y_n), \quad i = 1, 2, \dots, n$$

初期条件

$$y_i(x_0) = \eta_i, \quad i = 1, 2, \dots, n$$

独立変数 x の刻み幅 h 及び積分段数 m が与えられたとき,

$$x_r = x_0 + rh, \quad r = 1, 2, \dots, m$$

での数値解

$$y_i(x_r), \quad i = 1, 2, \dots, n$$

を 4 次 の 古 典 的 ル ン ゲ ・ ク ッ タ 法 で 計 算 し,

$$y_i(x_m), y'_i(x_m), \quad i = 1, 2, \dots, n$$

を出力する。

(2) 使用法

CALL RK4S/D/Q/C/B (X, H, Y, N, DY, DIFFUN, NSTEP, NFUN, ILL)

引 数	型 と 種 類*	属 性	内 容
X**	実 数 型	入 出 力	独立変数 x の初期値 x_0 を入力すると, m 段後の値 $x_m = x_0 + mh$ が出力される。
H**	実 数 型	入 力	独立変数の刻み幅 h 。 $H \neq 0$ 。 $H < 0$ でもよい。
Y*	実 数 型 1 次 元 配 列	入 出 力	解の初期値を入力すると $x = x_m$ での解の値が出力される。大きき N の 1 次元配列。
N	整 数 型	入 力	方程式の元数。 $N > 0$
DY*	実 数 型 1 次 元 配 列	出 力	$x = x_m$ での解の微係数の値が初めの N 個の成分に格納されている。大きき $3N$ の 1 次元配列。
DIFFUN	サブルーチン名	入 力	X, Y の関数として微係数の値を計算するためのサブルーチン。 DIFFUN (X, Y, DY) の形のサブルーチンとして利用者が用意しなければならない。
NSTEP	整 数 型	入 力	積分段数 m 。 $m \geq 1$
NFUN	整 数 型	入 出 力	最初の引用のときは NFUN = 0 と入力しなければならない。以後は微係数サブルーチンの引用回数 の 累 計 が 出 力 さ れ る。

引 数	型 と 種 類*	属 性	内 容
ILL	整 数 型	出 力	ILL=0: 正常終了。 ILL=30000: N ≤ 0, NSTEP ≤ 0 又は H = 0.0。

* RK4D (Q, C, B) の場合は, Y, DYを倍精度実数型 (4倍精度実数型, 複素数型, 倍精度複素数型) とする。

** RK4D (Q, C, B) の場合は, X, Hを倍精度実数型 (4倍精度実数型, 実数型, 倍精度実数型) とする。

(3) 使用例

0次のベッセルの微分方程式

$$y'' + y'/x + y = 0$$

において, 変数変換

$$y_1 = y, y_2 = -y'$$

を行ってえられる連立方程式

$$\begin{cases} y_1' = -y_2 \\ y_2' = y_1 - y_2/x \end{cases}$$

を, 初期条件

$$y_1(0) = 1, y_2(0) = 0$$

の下で解く。厳密解は

$$y_1 = J_0(x), y_2 = J_1(x)$$

である。xの刻み $h = 0.1$ とし5ステップ毎に途中結果と誤差を印刷し, 50ステップ積分する。

```

DIMENSION Y(2),DY(2),E(2)
EXTERNAL BES
N=2
X=0.0
H=0.1
Y(1)=1.0
Y(2)=0.0
NFUN=0
NSTEP=5
WRITE(6,600) X,Y
DO 10 I=1,10
CALL RK4S(X,H,Y,N,DY,BES,NSTEP,NFUN,ILL)
E(1)=Y(1)-BJ0(X)
E(2)=Y(2)-BJ1(X)
10 WRITE(6,600) X,Y,E
600 FORMAT(1H,10X,F5.1,2E15.7,2E11.3)
STOP
END
SUBROUTINE BES(X,Y,DY)
DIMENSION Y(2),DY(2)
DY(1)=-Y(2)
IF(ABS(X).LT.1.0E-2) DY(2)=0.5-X*X*0.1875
IF(ABS(X).GE.1.0E-2) DY(2)=Y(1)-Y(2)/X
RETURN
END

```

(4) 備 考

1. 導関数を計算するサブルーチンの名前を主プログラムで EXTERNAL 宣言しなければならない。
2. 最初の引用及び導関数の値が不連続的に変化する点での引用において, NFUN = 0 と入力することが必要である。NFUN は入出力両方に用いられるので, この変数の場所に定数を用いてはいけない。

3. 4 次のルンゲ・クッタ法の局所打切誤差は、刻み幅 h に対して

$$\varepsilon_l = \alpha h^5$$

の形で与えられる。 α の値は方程式に依存するので一概に何ともいえないが、解の変化がゆるやかな場合には 1 の程度と見てよい。このことと積分区間の長さなどを考え合わせて、 h を選ぶべきである。たとえば使用例のような場合には、4～5桁の精度の解をうるためには $h = 0.1$ 位が適当で、 $h = 0.01$ などとすると、打切誤差が不必要に小さい（単精度では丸めの誤差の最小単位より小さくなる）ばかりか、積分回数が増えるので丸めの誤差が増大し、かえって良い結果がえられない。

4. 本サブルーチンは解の変化がゆるやかで、刻み幅 h を変更する必要があまりない場合に適する。解の動きが急激で刻み幅の選択が困難な場合には、刻み幅の自動調節機能をもつ方法(RKF4AS/D, RKM4AS/D, ODEBSS/D)を採用すべきである。

RKF4AS / D

Solution of Initial Value Problems for Systems of First Order Differential Equations by the Rungu-Kutta-Fehlberg Fourth Order Method

ルンゲ・クッタ・フェールベルク 4 次法による連立 1 階常微分方程式の初期値問題の解法

作 成	二宮 市三	1980 年 4 月
形 式	サブルーチン 言語 ; FORTRAN サイズ ; 77, 78 行	

(1) 概 要

n 元連立 1 階常微分方程式

$$y'_i = f_i(x, y_1, y_2, \dots, y_n), i = 1, 2, \dots, n,$$

初期条件

$$y_i(x_0) = \eta_i, i = 1, 2, \dots, n,$$

独立変数 x の刻み幅の初期値 h_0 、積分段数 m 及び独立変数の目標値 x_e が与えられたとき、ルンゲ・クッタ・フェールベルクの 4 次及び 5 次法の組み合わせに基づく自動刻み幅調節算法を用いて、出力点 $x_f = \min(x_m, x_e)$ での解とその微係数の値

$$y_i(x_f), y'_i(x_f), i = 1, 2, \dots, n$$

を出力する。ただし、 x_m は初期値 x_0 、 h_0 から m 段の積分計算の後に到達する独立変数の値である。

(2) 使用法

CALL RKF4AS/D (X, H, Y, N, DY, EPS, DIFFUN, NSTEP, XEND, ERR, NFUN, ILL)

引 数	型 と 種 類*	属 性	内 容
X	実 数 型	入 出 力	独立変数の初期値 x_0 を入力すると、最終到達値 x_f が出力される。
H	実 数 型	入 出 力	刻み幅の初期値 h_0 を入力すると、 x_f からさらに積分を進める場合の適正な刻み幅 h が出力される。
Y	実 数 型 1 次 元 配 列	入 出 力	解の初期値を入力すると、 x_f での解の値が出力される。大きさ N の 1 次元配列。
N	整 数 型	入 力	方程式の元数 n 。 $0 < N \leq 100$
DY	実 数 型 1 次 元 配 列	出 力	x_f での解の微係数が初めの N 個の成分に格納されて出力される。大きさ $6N$ の 1 次元配列。
EPS	実 数 型	入 力	解の許容誤差。
DIFFUN	サブルーチン名	入 力	X, Y の関数として微係数を計算するためのサブルーチン。DIFFUN (X, Y, DY) の形式のサブルーチンとして利用者が用意する。
NSTEP	整 数 型	入 力	積分段数 m 。NSTEP ≤ 0 と与えると $m = \infty$ の場合と同じ扱いとなり、必ず $x_f = x_e$ となる。
XEND	実 数 型	入 力	独立変数の目標値 x_e 。 (XEND-X)*H > 0 でなければならない。

引 数	型 と 種 類*	属 性	内 容
ERR	実 数 型 1 次 元 配 列	出 力	最終段での解の各成分の打切絶対誤差の推定値。
NFUN	整 数 型	入 出 力	入力：初めて本ルーチンをコールするとき又は方程式に不連続的な変化があるときには NFUN = 0 とする。 最終段だけを再計算したいときは NFUN < 0 とする。 出力：NFUN ≤ 0 としたとき以後の微係数の計算回数の累計が出力される。
ILL	整 数 型	出 力	ILL = 0 : 正常終了。 ILL = 10000 : 刻み幅を変更して10回反復しても所定の精度が達成されなかった。 ILL = 30000 : 引数に対する制限が破られた。 上記以外の値は1回のコールにおいて、精度の達成に失敗したために行った反復の総数を表わす。

* 倍精度用のサブルーチンの場合には、実数型をすべて倍精度実数型とする。

(3) 計算法

参考文献¹⁾を参照のこと。

(4) 使用例

初期値問題

$$\begin{cases} y'_1 = y_1 / (2(x+1)) - 2xy_2, \\ y'_2 = y_2 / (2(x+1)) + 2xy_1, \end{cases} \begin{cases} y_1(0) = 1 \\ y_2(0) = 0 \end{cases}$$

を解くためのプログラムを下に示す。厳密解は

$$\begin{cases} y_1 = \sqrt{x+1} \cos x^2 \\ y_2 = \sqrt{x+1} \sin x^2 \end{cases}$$

である。

```

C      TEST FOR RKF4AD
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(2),DY(12),ERR(2),Z(2)
      EXTERNAL RHS
      X=0.DO
      H=0.1D0
      Y(1)=1.DO
      Y(2)=0.DO
      N=2
      EPS=1.D-7
      NS=0
      NF=0
      XE=2.5D0
      CALL RKF4AD(X,H,Y,N,DY,EPS,RHS,NS,XE,ERR,NF,ILL)
      XX=X*X
      S=DSQRT(X+1.DO)
      Z(1)=DCOS(XX)*S
      Z(2)=DSIN(XX)*S
      WRITE(6,600) H,X,(Y(J),Z(J),ERR(J),J=1,2),NF,ILL
600  FORMAT(1H ,2D13.5,2(2D15.7,D11.3),2I8)
      STOP
      END

```

```

C      SUBROUTINE FOR DERIVATIVES
      SUBROUTINE RHS(X,Y,DY)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(2),DY(2)
      DY(1)=Y(1)*0.5D0/(X+1.D0)-Y(2)*X*2.D0
      DY(2)=Y(2)*0.5D0/(X+1.D0)+Y(1)*X*2.D0
      RETURN
      END

```

(5) 備 考

1. 本ルーチンの使い方は大別すると、次の二通りになる。
 - (1) 目標値での解のみを出力する。そのためには、使用例のように XEND に目標値を入れ、NSTEP = 0 とする。
 - (2) 目標値までの途中の結果を出力する。そのためには二つの方法がある。
 - (a) XEND に目標値を入れたままにしておいて、NSTEP に比較的小さい正の値を入れて反復コールする。この場合、サブルーチンから戻ったときの X の値は、刻み幅自動調節のために不規則な値になる。
 - (b) 目標値までの間に適当に（例えば等間隔に）出力点を設定し、XEND にこの出力点を順次入れながら反復コールする。この場合には、規則的な点での出力がえられるので好都合だが、出力点をあまり密にとると、出力点での脱出の度ごとに強制的な刻み幅の変更が行われ、本来の刻み幅自動調節の機能を損なうことになる。
2. EPS で制御し、ERR で推定できるのは、局所打切誤差であって真の誤差ではない。
3. EPS は解の各成分について、その大きさが 1 以内なら絶対誤差の意味に、1 を越える場合には、その大きさの最大値に対する相対誤差の意味に用いられる。
4. NFUN の入力方法に注意すること。
5. ILL の値が 0, 10000, 30000 以外の値をとる場合でも解の値は必ずしも不正確とは限らない。
6. 本ルーチンとほとんど同じ使用法の姉妹ルーチン、RKM4AS/D, ODEBSS/D があるので、適宜取捨選択して利用されたい。

参 考 文 献

- 1) E. Fehlberg; "Klassische Runge-Kutta-Formeln vierter und niedrigerer Ordnung mit Schrittweiten-Kontrolle und ihre Anwendung auf Wärmeleitungs-probleme", Computing, Vol.6, pp.61-71 (1970).

RKM4AS / D

Solution of Initial Value Problems for Systems of First Order Differential Equations by the Runge-Kutta-Merluzzi Fourth Order Method

ルンゲ・クッタ・メルルッツィ 4 次法による連立 1 階常微分方程式の初期値問題の解法

作 成	二宮 市三 1980 年 4 月
形 式	サブルーチン 言語 ; FORTRAN サイズ ; 79, 80 行

(1) 概 要

n 元連立 1 階常微分方程式

$$y'_i = f_i(x, y_1, y_2, \dots, y_n), \quad i = 1, 2, \dots, n,$$

初期条件

$$y_i(x_0) = \eta_i, \quad i = 1, 2, \dots, n,$$

独立変数 x の刻み幅の初期値 h_0 、積分段数 m 及び独立変数の目標値 x_e が与えられたとき、累積打切誤差評価能力を持つメルルッツィの 4 次のルンゲ・クッタ法に基づく自動刻み幅調節算法を用いて、出力点 $x_f = \min(x_m, x_e)$ での解

$$y_i(x_f), \quad i = 1, 2, \dots, n$$

を出力する。ただし、 x_m は初期値 x_0 、 h_0 から m 段の積分計算の後に到達する独立変数の値である。

(2) 使用法

CALL RKM4AS/D (X, H, Y, N, DY, EPS, DIFFUN, NSTEP, XEND, ERR, NFUN, ILL)

引 数	型 と 種 類*	属 性	内 容
X	実 数 型	入 出 力	独立変数の初期値 x_0 を入力すると、最終到達値 x_f が出力される。
H	実 数 型	入 出 力	刻み幅の初期値 h_0 を入力すると、 x_f からさらに積分を進める場合の適正な刻み幅 h が出力される。
Y	実 数 型 1 次 元 配 列	入 出 力	解の初期値を入力すると、 x_f での解の値が出力される。大きさ N の 1 次元配列。
N	整 数 型	入 力	方程式の元数 n 。 $0 < N \leq 100$
DY	実 数 型 1 次 元 配 列	作 業 領 域	大きさ $6N$ の 1 次元配列。
EPS	実 数 型	入 力	解の許容誤差。
DIFFUN	サブルーチン名	入 力	X, Y の関数として微係数を計算するためのサブルーチン。DIFFUN(X, Y, DY) の形式のサブルーチンとして利用者が用意する。
NSTEP	整 数 型	入 力	積分段数 m 。 $NSTEP \leq 0$ と与えると $m = \infty$ の場合と同じ扱いとなり、必ず $x_f = x_e$ となる。
XEND	実 数 型	入 力	独立変数の目標値 x_e 。 (XEND-X)*H > 0 でなければならない。

引 数	型 と 種 類*	属 性	内 容
ERR	実 数 型 1 次 元 配 列	出 力	最終段での解の各成分の打切絶対誤差の推定値。
NFUN	整 数 型	入 出 力	入力：初めて本ルーチンをコールするとき又は方程式に不連続的な変化があるときには NFUN = 0 とする。 最終段だけを再計算したいときは NFUN < 0 とする。 出力：NFUN ≤ 0 としたとき以後の微係数の計算回数の累計が出力される。
ILL	整 数 型	出 力	ILL = 0 : 正常終了。 ILL = 10000 : 刻み幅を変更して10回反復しても所定の精度が達成されなかった。 ILL = 30000 : 引数に対する制限が破られた。 上記以外の値は1回のコールにおいて、精度の達成に失敗したために行った反復の総数を表わす。

* 倍精度用のサブルーチンの場合には、実数型をすべて倍精度実数型とする。

(3) 計算法

参考文献¹⁾を参照のこと。

(4) 使用例

初期値問題

$$\begin{cases} y_1' = y_1 / (2(x+1)) - 2xy_2, \\ y_2' = y_2 / (2(x+1)) + 2xy_1, \end{cases} \quad \begin{cases} y_1(0) = 1 \\ y_2(0) = 0 \end{cases}$$

を解くためのプログラムを下に示す。厳密解は

$$\begin{cases} y_1 = \sqrt{x+1} \cos x^2 \\ y_2 = \sqrt{x+1} \sin x^2 \end{cases}$$

である。

```

C      TEST FOR RKM4AD
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(2),DY(12),ERR(2),Z(2)
      EXTERNAL RHS
      X=0.00
      H=0.100
      Y(1)=1.00
      Y(2)=0.00
      N=2
      EPS=1.0D-7
      NS=0
      NF=0
      XE=10.00
      CALL RKM4AD(X,H,Y,N,DY,EPS,RHS,NS,XE,ERR,NF,ILL)
      XX=X*X
      S=DSQRT(X+1.00)
      Z(1)=DCOS(XX)*S
      Z(2)=DSIN(XX)*S
      WRITE(6,600) H,X,(Y(J),Z(J),ERR(J),J=1,2),NF,ILL
600  FORMAT(1H ,2D13.5,2(2D15.7,D11.3),2I8)
      STOP
      END

```

```

C      SUBROUTINE FOR DERIVATIVES
      SUBROUTINE RHS(X,Y,DY)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(2),DY(2)
      DY(1)=Y(1)*0.5D0/(X+1.D0)-Y(2)*X*2.D0
      DY(2)=Y(2)*0.5D0/(X+1.D0)+Y(1)*X*2.D0
      RETURN
      END

```

(5) 備 考

1. 本ルーチンの使い方は大別すると、次の二通りになる。
 - (1) 目標値での解のみを出力する。そのためには、使用例のように XEND に目標値を入れ、NSTEP = 0 とする。
 - (2) 目標値までの途中の結果を出力する。そのためには二つの方法がある。
 - (a) XEND に目標値を入れたままにしておいて、NSTEP に比較的にか小さい正の値を入れて反復コールする。この場合、サブルーチンから戻ったときの X の値は、刻み幅自動調節のために不規則な値になる。
 - (b) 目標値までの間に適当に（例えば等間隔に）出力点を設定し、XEND にこの出力点を順次入れながら反復コールする。この場合には規則的な点での出力がえられるので好都合だが、出力点をあまり密にとると、出力点での脱出の度ごとに強制的な刻み幅の変更が行われ、本来の刻み幅自動調節の機能を損なうことになる。
2. EPS で制御し、ERR で推定できるのは、累積打切誤差であるが、真の誤差ではない。
3. EPS は解の各成分について、その大きさが 1 以内なら絶対誤差の意味に、1 を超える場合には、その大きさの最大値に対する相対誤差の意味に用いられる。
4. NFUN の入力方法に注意すること。
5. ILL の値が 0, 10000, 30000 以外の値をとる場合でも解の値は必ずしも不正確とは限らない。
6. 本ルーチンとほとんど同じ使用法の姉妹ルーチン、RKF4AS/D, ODEBSS/D があるので、適宜取捨選択して利用されたい。

参 考 文 献

- 1) P. Merluzzi et al; "Runge-Kutta Integration Algorithms with Built-in Estimation of the Accumulated Truncation Error", Computing, Vol.20, pp.1-16 (1978).

ODEBSS / D

Solution of Initial Value Problems for Systems of First Order Differential Equations by the Rational Extrapolation Method

有理補外法による連立1階常微分方程式の初期値問題の解法

作成	二宮 市三 1980年4月
形式	サブルーチン 言語; FORTRAN サイズ; 146, 147行

(1) 概要

n 元連立1階常微分方程式

$$y'_i = f_i(x, y_1, y_2, \dots, y_n), \quad i = 1, 2, \dots, n,$$

初期条件

$$y_i(x_0) = \eta_i, \quad i = 1, 2, \dots, n,$$

独立変数 x の刻み幅の初期値 h_0 , 積分段数 m 及び独立変数の目標値 x_e が与えられたとき, プリルシュ・ストアの有理補外法に基づく自動刻み幅調節算法を用いて, 出力点 $x_f = \min(x_m, x_e)$ での解とその微係数の値

$$y_i(x_f), y'_i(x_f), \quad i = 1, 2, \dots, n$$

を出力する。ただし, x_m は初期値 x_0, h_0 から m 段の積分計算の後に到達する独立変数の値である。

(2) 使用法

CALL ODEBSS/D (X, H, Y, N, DY, EPS, DIFFUN, NSTEP, XEND, ERR, NFUN, IND)

引数	型と種類*	属性	内容
X	実数型	入出力	独立変数の初期値 x_0 を入力すると, 最終到達値 x_f が出力される。
H	実数型	入出力	刻み幅の初期値 h_0 を入力すると, x_f からさらに積分を進める場合の適正な刻み幅 h が出力される。
Y	実数型 1次元配列	入出力	解の初期値を入力すると, x_f での解の値が出力される。大きさ N の1次元配列。
N	整数型	入力	方程式の元数 n_0 。 $0 < N \leq 100$
DY	実数型 1次元配列	出力	x_f での解の微係数が出力される。大きさ N の1次元配列。
EPS	実数型	入力	解の許容誤差。
DIFFUN	サブルーチン名	入力	X, Y の関数として微係数を計算するためのサブルーチン。DIFFUN (X, Y, DY) の形式のサブルーチンとして利用者が用意する。
NSTEP	整数型	入力	積分段数 m 。NSTEP ≤ 0 と与えると $m = \infty$ の場合と同じ扱いとなり, 必ず $x_f = x_e$ となる。
XEND	実数型	入力	独立変数の目標値 x_e 。 (XEND - X) * H > 0 でなければならない。

引数	型と種類*	属性	内容
ERR	実数型 1次元配列	出力	最終段での解の各成分の打切絶対誤差の推定値。
NFUN	整数型	入出力	入力: 初めて本ルーチンをコールするとき又は方程式に不連続的な変化があるときには NFUN = 0 とする。 最終段だけを再計算したいときは NFUN < 0 とする。 出力: NFUN ≤ 0 としたとき以後の微係数の計算回数の累計が出力される。
IND	整数型	入出力	入力: IND = 0 とすると有理式補外法が使われる。 IND ≠ 0 とすると多項式補外法が使われる。 出力: IND = 0: 正常終了。 IND = 10000: 刻み幅を半分にして6回反復しても所要の精度が達成されなかった。 IND = 30000: 引数エラー。 上記以外の値は1回のコールにおいて精度の達成に失敗したために行った反復の総数を表わす。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 計算法

参考文献¹⁾を参照のこと。

(4) 使用例

初期値問題

$$\begin{cases} y'_1 = y_1 / (2(x+1)) - 2xy_2, \\ y'_2 = y_2 / (2(x+1)) + 2xy_1, \end{cases} \quad \begin{cases} y_1(0) = 1 \\ y_2(0) = 0 \end{cases}$$

を解くためのプログラムを下に示す。厳密解は

$$\begin{cases} y_1 = \sqrt{x+1} \cos x^2 \\ y_2 = \sqrt{x+1} \sin x^2 \end{cases}$$

である。

```

C      TEST FOR ODEBSD
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(2),DY(2),ERR(2),Z(2)
      EXTERNAL RHS
      X=0.00
      H=0.100
      Y(1)=1.00
      Y(2)=0.00
      N=2
      EPS=1.0-7
      NS=0
      NF=0
      XE=2.500
      IND=0
      CALL ODEBSD(X,H,Y,N,DY,EPS,RHS,NS,XE,ERR,NF,IND)
      XX=X*X
      S=DSQRT(X+1.00)
      Z(1)=DCOS(XX)*S
      Z(2)=DSIN(XX)*S
      WRITE(6,600) H,X,(Y(J),Z(J),ERR(J),J=1,2),NF,IND
600  FORMAT(1H ,2D13.5,2(2D15.7,D11.3),2I8)
      STOP
      END

```

```

C      SUBROUTINE FOR DERIVATIVES
      SUBROUTINE RHS(X,Y,DY)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(2),DY(2)
      DY(1)=Y(1)*0.5D0/(X+1.D0)-Y(2)*X*2.D0
      DY(2)=Y(2)*0.5D0/(X+1.D0)+Y(1)*X*2.D0
      RETURN
      END

```

(5) 備考

1. 本ルーチンの使い方は大別すると、次の二通りになる。
 - (1) 目標値での解のみを出力する。そのためには、使用例のように XEND に目標値を入れ、NSTEP = 0 とする。
 - (2) 目標値までの途中の結果を出力する。そのためには二つの方法がある。
 - (a) XEND に目標値を入れたままにしておいて、NSTEP に比較的に小さい正の値を入れて反復コールする。この場合、サブルーチンから戻ったときの X の値は、刻み幅自動調節のために不規則な値になる。
 - (b) 目標値までの間に適当に（例えば等間隔に）出力点を設定し、XEND にこの出力点を順次入れながら反復コールする。この場合には規則的な点での出力がえられるので好都合だが、出力点をあまり密にとると、出力点での脱出の度ごとに強制的な刻み幅の変更が行われ、本来の刻み幅自動調節の機能を損なうことになる。
2. EPS で制御し、ERR で推定できるのは、局所打切誤差であって真の誤差ではない。
3. EPS は解の各成分について、その大きさが 1 以内なら絶対誤差の意味に、1 を超える場合には、その大きさの最大値に対する相対誤差の意味に用いられる。
4. NFUN の入力方法に注意すること。
5. IND の値が 0, 10000, 30000 以外の値を取る場合でも解の値は必ずしも不正確とは限らない。
6. 本ルーチンとほとんど同じ使用法の姉妹ルーチン RKF4AS/D, RKM4AS/D, があるので、適宜取捨選択して利用されたい。

参考文献

- 1) R. Bulirsch and J. Stoer; "Numerical Treatment of Ordinary Differential Equations by Extrapolation", Numer. Math., Vol.8, pp.1-13 (1966).

SINH P / DSINH P / QSINH P
COSH P / DCOSH P / QCOSH P
TANH P / DTANH P / QTANH P
COTH P / DCOTH P / QCOTH P

Trigonometric Functions for the Argument $\frac{\pi}{2}x$

引数 $\frac{\pi}{2}x$ に対する三角関数

作 成	二宮 市三	1980年1月
形 式	関数	言語 ; アセンブラ (4倍精度のものは FORTRAN) サイズ ; 117, 152, 47, 117, 152, 47, 134, 174, 55, 134, 174, 55行

(1) 概 要

$\left. \begin{array}{l} \text{SINH P (DSINH P, QSINH P)} \\ \text{COSH P (DCOSH P, QCOSH P)} \\ \text{TANH P (DTANH P, QTANH P)} \\ \text{COTH P (DCOTH P, QCOTH P)} \end{array} \right\}$ は、単 (倍, 4倍) 精度実数 x に対して $\left\{ \begin{array}{l} \sin \frac{\pi}{2} x \\ \cos \frac{\pi}{2} x \\ \tan \frac{\pi}{2} x \\ \cot \frac{\pi}{2} x \end{array} \right\}$ を単 (倍, 4

倍) 精度で計算する。

(2) 使用法

1. SINH P (X), COSH P (X), TANH P (X), COTH P (X)

DSINH P (D), DCOSH P (D), DTANH P (D), DCOTH P (D)

QSINH P (Q), QCOSH P (Q), QTANH P (Q), QCOTH P (Q)

X (D, Q) は単 (倍, 4倍) 精度実数型の任意の式である。倍 (4倍) 精度の関数名は倍 (4倍) 精度の宣言を要する。

2. 引数の範囲

$$|X| \leq 2^{19} \doteq 5.2 \cdot 10^5, |D| \leq 2^{51} \doteq 2.2 \cdot 10^{15}, |Q| \leq 2^{106} \doteq 8.1 \cdot 10^{31}$$

3. エラー処理

範囲外の引数が与えられた場合及び関数値が ∞ になるような引数が与えられた場合は、エラーとしてメッセージを印刷し、関数値を0として計算を続行する。(FNERST p. xii参照)

(3) 備 考

1. 引数が π を因子として含む場合の三角関数の値は、SIN, COS など普通の基本外部関数を使っても計算できるが、本節の諸関数を使う方が次の理由で合理的である。

(1) π の値を書く必要がない。

(2) 乗算2回分だけ速度が速い。

(3) 精度がよい。

例えば、SIN (1.570796 * X) よりは SINH P (X) の方が、また、DCOS (3.1415926535897932D0 * X) よりは DCOSH P (X+X) の方がよい。COSH P (1.0) は正確に0となるが、COS (1.570796)

はそうはならない。

2. 関数名の末尾の HP は HALF PI の意味である。双曲線関数の外前の末尾の H は HYPERBOLIC の意味であるから混同しないようにする。
3. 引数に誤差がある場合は、引数の整数部分の桁数程度の桁数だけ関数値の末尾に誤差が入る。これは標準関数においても同様である。
4. TANHP, COTHP 等の極の附近の関数値については精度は保証できない。

ASINH / DASINH / QASINH
ACOSH / DACOSH / QACOSH
ATANH / DATANH / QATANH

Inverse Hyperbolic Function

逆双曲線関数

作成	二宮 市三	1974年4月	改訂	1977年4月
形式	関数	言語 ; FORTRAN サイズ ; 18, 26, 37, 11, 12, 11, 18, 24, 34行		

(1) 概要

ASINH (DASINH, QASINH)
ACOSH (DACOSH, QACOSH)
ATANH (DATANH, QATANH) } は, 単(倍, 4倍)精度実数 x に対して $\left\{ \begin{matrix} \sinh^{-1} x \\ \cosh^{-1} x \\ \tanh^{-1} x \end{matrix} \right\}$ を単(倍,

4倍)精度で計算する。ただし,

$$\sinh^{-1} x = \log (x + \sqrt{1+x^2})$$

$$\cosh^{-1} x = \log (x + \sqrt{x^2-1})$$

$$\tanh^{-1} x = \frac{1}{2} \log \frac{1+x}{1-x}$$

(2) 使用法

1. ASINH (X), ACOSH (X), ATANH (X)

DASINH (D), DACOSH (D), DATANH (D)

QASINH (Q), QACOSH (Q), QATANH (Q)

X (D, Q) は単(倍, 4倍)精度実数型の任意の式である。倍(4倍)精度の関数名は, 倍(4倍)精度の宣言を要する。

2. 引数の範囲

逆双曲線正弦関数には引数の制限はない。

逆双曲線余弦関数に対しては, $X \geq 1, D \geq 1, Q \geq 1$ 。

逆双曲線正接関数に対しては, $|X| < 1, |D| < 1, |Q| < 1$ 。

3. エラー処理

範囲外の引数が与えられた場合は, エラーとしてメッセージを印刷し, 関数値を0として計算を続行する。(FNERST p. XII参照)

(3) 計算法

1. ASINH (DASINH)

(1) $|x| < \frac{3}{4}$ ならば $\sinh^{-1} x$ を近似多項式によって計算する。

(2) $|x| \geq \frac{3}{4}$ ならば $y = |x|, \sinh^{-1} x = \text{sign } x \cdot \sinh^{-1} y, \sinh^{-1} y = \log (y + \sqrt{1+y^2})$ と計算す

る。

(3) $y \geq 4096$ ($y \geq 3 \cdot 10^8$)のときは $\sinh^{-1} y = \log 2y$ と計算する。

2. ACOSH (DACOSH)

(1) $x < 1$ ならばエラーとする。

(2) $1 < x < 4096$ ($1 < x < 3 \cdot 10^8$)ならば $\cosh^{-1} x = \log(x + \sqrt{x^2 - 1})$ と計算する。

(3) $x \geq 4096$ ($x \geq 3 \cdot 10^8$)ならば $\cosh^{-1} x = \log 2x$ と計算する。

3. ATANH (DATANH)

(1) $|x| \geq 1$ ならばエラーとする。

(2) $|x| \leq \frac{1}{3}$ ならば $\tanh^{-1} x$ を近似多項式によって計算する。

(3) $\frac{1}{3} < |x| < 1$ ならば $\tanh^{-1} x = \frac{1}{2} \log \frac{1+x}{1-x}$ と計算する。

(4) 備考

本節の関数は、すべて対数関数で定義される簡単な関数であるが、定義式のとおり計算すると、絶対値の小さい引数に対しては精度のよい値が得られない。本節の関数を用いれば、絶対値の小さい引数に対しては特別の処置がとられているので、精度の低下は起こらない。

EXP1 / DEXP1 / QEXP1
ALOG1 / DLOG1 / QLOG1

Functions $e^x - 1$ and $\log(1+x)$

関数 $e^x - 1$ 及び $\log(1+x)$

作成	二宮 市三	1981年4月, 1977年4月
形式	関数	言語; FORTRAN サイズ; 21, 25, 28, 18, 24, 34行

(1) 概要

EXP1 (DEXP1, QEXP1)
 ALOG1 (DLOG1, QLOG1) } は, 単(倍, 4倍)精度実数 x に対して $\left\{ \begin{matrix} e^x - 1 \\ \log(1+x) \end{matrix} \right\}$ を単(倍, 4倍)精度で計算する。

(2) 使用法

1. EXP1 (X), DEXP (D), QEXP1 (Q)

ALOG1 (X), DLOG1 (D), QLOG1 (Q)

X (D, Q) は単(倍, 4倍)精度実数型の任意の式である。倍(4倍)精度の関数名は倍(4倍)精度の宣言を要する。

2. 引数の範囲

EXP1 等については, $X \leq 174.673$, $D \leq 174.673$, $Q \leq 174.673$ 。

ALOG1 等については, $X > -1$, $D > -1$, $Q > -1$ 。

3. エラー処理

範囲外の引数が与えられた場合は, エラーとしてメッセージを印刷し, 関数値を0として計算を続行する。(FNERST p. xii参照)

(3) 計算法

1. EXP1/DEXP1/QEXP1

(1) $x < -18.421$ (DEXP1 の場合は $x < -41.447$, QEXP1 の場合は $x < -77.633$) のとき, $f(x) = -1$ とする。

(2) $|x| \leq 1$ のとき, 近似多項式 P, Q によって $e^x - 1 = 2xP(x^2) / [Q(x^2) - xP(x^2)]$ と計算する。

(3) (1), (2)以外の x に対しては, 定義式通り $e^x - 1$ と計算する。

2. ALOG1/DLOG1/QLOG1

(1) $x \leq -1$ ならばエラーとする。

(2) $-\frac{1}{2} \leq x < 1$ ならば $y = \frac{x}{x+2}$ とし, $\log(1+x) = \log \frac{1+y}{1-y}$ を y に関する近似多項式によって計算する。

(3) $-1 < x < -\frac{1}{2}$ 又は $x \geq 1$ ならば $\log(1+x)$ のまま計算する。

(4) 備考

本節の関数を標準関数によって定義式どおり計算すると, $x = 0$ の近傍で精度が損われる。

ALANGV / DLANGV

Langevin Function

ランジュバン関数

作成	二宮 市三	1981年4月
形式	関数	言語 ; 21, 26行

(1) 概要

ALANGV (DLANGV) は単 (倍) 精度実数 x に対してランジュバン関数 $L(x) = \coth x - 1/x$ を単 (倍) 精度で計算する。

(2) 使用法

1. ALANGV (X), DLANGV (D)

X (D) は単 (倍) 精度実数型の任意の式である。DLANGV は倍精度の宣言を必要とする。

2. 引数の範囲

引数に対する制限はない。

(3) 計算法

1. $|x| \leq 4$ のとき, 最良近似有利式 R によって $L(x) = xR(x^2)$ と計算する。

2. $4 < |x| < 10$ (DLANGV の場合 $4 < |x| < 20$) のとき

$L(x) = \text{sign}(x)(2e^{-|x|}/(1-e^{-|x|}) - 1/|x| + 1)$ と計算する。

3. $|x| \geq 10$ (DLANGV の場合 $|x| \geq 20$) のとき $L(x) = \text{sign}(x)(1 - 1/|x|)$ と計算する。

(4) 備考

$L(x)$ を定義式どおりに計算すると $x = 0$ の近傍で精度が損われる。

CABS1 / CDABS1 / CQABS1

Sum of Moduli of Real and Imaginary Parts of a Complex Number

複素数の実部と虚部の絶対値和

作成	二宮 市三	1980年1月
形式	関数	言語 ; アセンブラ サイズ ; 42行

(1) 概要

CABS1 (CDABS1, CQABS1) は単 (倍, 4 倍) 精度複素数 $z = x + iy$ に対して, $\|z\|_1 = |x| + |y|$ を単 (倍, 4 倍) 精度で計算する。

(2) 使用法

1. CABS1 (C), CDABS1 (B), CQABS1 (Z),

C (B, Z) は単 (倍, 4 倍) 精度複素数型の任意の式である。CDABS1 (CQABS1) は倍 (4 倍) 精度の宣言を要する。

2. 引数に対する制限はない。

(3) 備考

1. 複素数 $z = x + iy$ の普通の意味の絶対値 $\|z\|_2 = |z| = \sqrt{x^2 + y^2}$ を与えるものに, 基本外部関数 CABS (CDABS, CQABS) があるが, 平方根を含み, しかも定義式のまま計算できない事情があって, 計算速度がおそい。収束判定などで複素数の小ささを調べるためには, 簡単な絶対値和 $\|z\|_1$ で充分である。本節の関数の存在理由はここにある。ちなみに, 同じ ϵ による判定 $\|z\|_2 < \epsilon$, $\|z\|_1 < \epsilon$ では後者の方が強い判定である。すなわち, $\|z\|_1 < \epsilon$ であれば必ず $\|z\|_2 < \epsilon$ となっているのである。
2. CDABS1 (CQABS1) の代わりに DCABS1 (QCABS1) と書いてもよい。

FCTRL / DFCTRL / QFCTRL**FFCTR / DFFCTR / QFFCTR****Factorial and Double Factorial****階乗及び二重階乗**

作成	二宮 市三	1980年1月
形式	関数	言語; FORTRAN サイズ; 25, 41, 66, 36, 62, 106行

(1) 概要

FCTRL (DFCTRL, QFCTRL)
FFCTR (DFFCTR, QFFCTR)}は、整数 n に対して {階乗 $n!$
二重階乗 $n!!$ } を単(倍, 4倍)精度で計算する。

ただし、

$$n! = \begin{cases} 1 & n = 0 \\ n(n-1)\cdots 2 \cdot 1 & n > 0 \end{cases}$$

$$n!! = \begin{cases} 1 & n = 0 \\ n(n-2)\cdots 4 \cdot 2 & n \geq 2 \text{ (偶数)} \\ n(n-2)\cdots 3 \cdot 1 & n \geq 1 \text{ (奇数)} \end{cases}$$

(2) 使用法

1. FCTRL (N), DFCTRL (N), QFCTRL (N)
FFCTR (N), DFFCTR (N), QFFCTR (N)

N は整数型の任意の式である。倍 (4倍) 精度の関数名は、倍 (4倍) 精度の宣言を要する。

2. 引数の範囲

階乗については、 $0 \leq N \leq 57$ 。

二重階乗については、 $0 \leq N \leq 97$ 。

3. エラー処理

範囲外の引数が与えられた場合は、エラーとしてメッセージを印刷し、関数値を0として計算を続行する。(FNERST p. xii参照)

(3) 計算法

あらかじめ十分な精度で計算された数表を利用する。

COMB / DCOMB

Binomial Coefficient

二項係数

作成	二宮 市三	1982年4月
形式	関数	言語 ; FORTRAN サイズ ; 23, 25行

(1) 概要

COMB (DCOMB) は整数 m, n に対して、二項係数 $mCn = \binom{m}{n} = \frac{m!}{n!(m-n)!}$ を単 (倍) 精度で計算する。

(2) 使用法

1. COMB (M, N)

DCOMB (M, N)

M, N は整数型の任意の式である。

2. 引数の範囲

$$1 \leq M \leq 1000, \quad 0 \leq N \leq M$$

ただし、関数値がオーバーフローとなる範囲を除く。

3. エラー処理

範囲外の引数が与えられた場合はエラーとしてメッセージを印刷し、関数値を 0 として計算を続行する (FNERST p. xii 参照)

(3) 計算法

1. $k = \min(n, m-n)$ とする。

2. $m \leq 56, k > 3$ のとき FCTRL (DFCTRL) を引用して

$$mCk = \frac{m!}{k!(m-k)!} \text{ とする。}$$

3. 上記以外の場合は次のようにする。

$mC_0 = 1$ から出発し、漸化式

$$mC_r = mC_{r-1} \cdot \frac{m-r+1}{r}$$

を反復する。

ZETNO / DZETNO / QZETNO

Riemann Zeta Function

リーマン・ゼータ関数

作成	二宮 市三	1981年4月
形式	関数	言語; FORTRAN サイズ; 42, 62, 162行

(1) 概要

ZETNO (DZETNO, QZETNO) は正の整数 n に対して、リーマン・ゼータ関数 $\zeta(n)$ を単 (倍, 4倍) 精度で計算する。

$$\zeta(n) = \sum_{k=1}^{\infty} \frac{1}{k^n}$$

(2) 使用法

1. ZETNO (N), DZETNO (N), QZETNO (N)

N は整数型の任意の式である。DZETNO (QZETNO) は倍 (4倍) 精度の宣言を必要とする。

2. 引数の範囲

$$N \geq 1$$

3. エラー処理

範囲外の引数が与えられた場合は、エラーとしてメッセージを印刷し、関数値を0として計算を続行する。(FNERST p. XII参照)

(3) 計算法

あらかじめ十分な精度で計算された数表を利用する。

(4) 備考

1. $n = 1$ に対しては本来の $\zeta(1)$ は未定義であるので、その代わりにオイラーの定数 $\gamma = \lim_{k \rightarrow \infty} \left(\sum_{i=1}^k \frac{1}{i^n} - \log k \right)$ を値とする。

2. 十分大きい n に対しては、 $\zeta(n) \approx 1 + 2^{-n}$ となる。

3. ベルヌーイ数 B_{2n} は、 $B_{2n} = (-1)^{n+1} \frac{2(2n)!}{(2\pi)^{2n}} \zeta(2n)$ で与えられる。

参考文献

- 1) Handbook of Mathematical Functions, Dover, N.Y., p.804 (1970).

BERNO / DBERNO / QBERNO

Bernoulli Numbers

ベルヌーイ数

作成	二宮 市三	1981年4月
形式	関数	言語; FORTRAN サイズ; 57, 57, 57行

(1) 概要

BERNO (DBERNO, QBERNO) は正の整数 n に対して、ベルヌーイ数 B_{2n} を単 (倍, 4倍) 精度で計算する。

(2) 使用法

1. BERNO (N), DBERNO (N), QBERNO (N)

N は整数型の任意の式である。DBERNO (QBERNO) は倍 (4倍) 精度の宣言を必要とする。

2. 引数の範囲

$$1 \leq N \leq 48$$

3. エラー処理

範囲外の引数が与えられた場合は、エラーとしてメッセージを印刷し、関数値を 0 として計算を続行する。(FNERST p. xii 参照)

(3) 計算法

あらかじめ十分な精度で計算された数表を利用する。

(4) 備考

1. ベルヌーイ数 B_{2n} とリーマン・ゼータ関数 $\zeta(n)$ との間には、 $B_{2n} = (-1)^{n+1} \frac{2(2n)!}{(2\pi)^{2n}} \zeta(2n)$ という関係がある。
2. $n \rightarrow \infty$ のとき $|B_{2n}|$ はおおよそ $(n/e\pi)^{2n}$ の程度で増大する。

参考文献

- 1) Handbook of Mathematical Functions, Dover, N.Y., p.804 (1970).

GAMCO / DGAMCO / QGAMCO

Coefficients of Taylor Series for $1/\Gamma(x)$

$1/\Gamma(x)$ のテーラー級数展開係数

作成	二宮 市三	1981年4月
形式	関数	言語 ; FORTRAN サイズ ; 91, 91, 91行

(1) 概要

GAMCO (DGAMCO, QGAMCO) は正の整数 n に対して $1/\Gamma(x)$ のテーラー級数展開係数 C_n の値を単 (倍, 4 倍) 精度で計算する。ただし,

$$1/\Gamma(x) = \sum_{n=1}^{\infty} C_n x^n$$

(2) 使用法

1. GAMCO(N), DGAMCO(N), QGAMCO(N)

N は整数型の任意の式である。DGAMCO(QGAMCO) は倍 (4 倍) 精度の宣言を必要とする。

2. 引数の範囲

$$N \geq 1$$

3. エラー処理

範囲外の引数が与えられた場合は, エラーとしてメッセージを印刷し, 関数値を 0 として計算を続行する。(FNERST p. xii 参照)

(3) 計算法

あらかじめ十分な精度で計算された数表を利用する。

(4) 備考

1. $n \geq 80$ に対してはアンダーフローのため $C_n = 0$ とする。
2. C_2 はオイラーの定数 γ である。

RGAMA / DRGAMA

Reciprocal of Gamma Function

ガンマ関数の逆数

作成	二宮 市三	1981年4月
形式	関数	言語; FORTRAN サイズ; 41, 54行

(1) 概要

RGAMA (DRGAMA) は単 (倍) 精度実数 x に対して $1/\Gamma(x)$ を単 (倍) 精度で計算する。

(2) 使用法

1. RGAMA (X), DRGAMA (D)

X (D) は, 単 (倍) 精度実数型の任意の式である。DRGAMA は倍精度の宣言を必要とする。

2. 引数の範囲

$$X \geq -56, D \geq -56$$

3. エラー処理

範囲外の引数が与えられた場合は, エラーとしてメッセージを印刷し, 関数値を 0 として計算を続行する。(FNERST p. XII 参照)

(3) 計算法

- $-56 \leq x < 0$ ならば反転公式 $\frac{1}{\Gamma(x)} = \frac{\Gamma(1-x)\sin\pi x}{\pi}$ により $1/\Gamma(1-x)$ の計算に変換する。
- $0 \leq x \leq 32$ の場合は必要に応じて漸化式 $\Gamma(1+x) = x\Gamma(x)$ により, 引数を $1 \leq x \leq 2$ の範囲に変換し, 最良近似多項式 P を用いて $1/\Gamma(x) = P(t), t = x-1$ と計算する。
- $32 < x \leq 57$ ならば, $\log \Gamma(x)$ を近似式によって計算し, $\frac{1}{\Gamma(x)} = e^{-\log\Gamma(x)}$ とする。
- $x > 57$ ならば $\frac{1}{\Gamma(x)} = 0$ とする。

(4) 備考

- $\Gamma(x)$ は $x = 0, -1, -2, \dots$ に一位の極を持つが, $1/\Gamma(x)$ は無限遠点以外に特異点を持たない。いわゆる整関数である。
- ガンマ関数が分母に入っている有理式の計算には, ガンマ関数プログラムよりも本関数プログラムを用いる方が合理的である。

参考文献

- 1) Handbook of Mathematical Functions, Dover, N.Y., p.253 (1970).

CGAMMA / CDGAMA

Gamma Function for Complex Arguments

複素変数のガンマ関数

作成	二宮 市三	1977年4月
形式	関数	言語 ; FORTRAN サイズ ; 76, 77行

(1) 概要

CGAMMA (CDGAMA) は単 (倍) 精度複素数 z に対して, $\Gamma(z)$ を単 (倍) 精度複素数として計算する。

(2) 使用法

1. CGAMMA (C), CDGAMA (B)

C(B) は単 (倍) 精度複素数型の任意の式である。CGAMMA (CDGAMA) は単 (倍) 精度複素数の宣言を要する。

2. 引数の範囲

複素平面から 0 及び負の整数並びにガンマ関数の値がオーバーフローになる領域を除いたもの。

3. エラー処理

範囲外の引数が与えられた場合は, エラーとしてメッセージを印刷し, 関数値を 0 として計算を続行する。(FNERST p. XII 参照)

(3) 計算法

引数を $z = x + iy$ とする。

1. $|z| \leq 1$ 又は $|x| \leq 0.5, |y| \leq 1.0$ であれば, $1/\Gamma(z)$ をテーラー級数で計算する。

2. $x < 0$ ならば, 反転公式 $\Gamma(z) = \pi / ((1-z) \cdot \sin \pi z)$ によって, $x > 0$ の場合に還元する。

3. $x^2 + y^2 > 16(32)$ ならば $\log \Gamma(z)$ を, 漸近級数 $\log \Gamma(z) = (z - 1/2) \cdot \log z + \log \sqrt{2\pi} - z + \sum_{n=1}^{\infty} C_n / z^{2n-1}$ の部分和によって計算する。

4. $|y| \leq 1.0$ で (1), (3) 以外の場合は, 漸化式 $\Gamma(z) = \Gamma(z+1)/z$ を必要なだけ反復して 1. の場合に帰着させる。

5. $|y| > 1.0$ で $x^2 + y^2 \leq 16(32)$ ならば, 漸化式 $\Gamma(z+1) = z\Gamma(z)$ を必要なだけ反復して 3. の場合に帰着させる。

(4) 備考

1. CDGAMA の代わりに DCGAMA と書いてもよい。

DIGAM / DDIGAM

Digamma Function

ディガンマ関数

作成	二宮 市三	1981年4月
形式	関数	言語; FORTRAN サイズ; 37, 47行

(1) 概要

DIGAM (DDIGAM) は単(倍)精度実数 x に対してディガンマ関数 $\psi(x) = \Gamma'(x)/\Gamma(x)$ を単(倍)精度で計算する。

(2) 使用法

1. DIGAM (X), DDIGAM (D)

$X(D)$ は単(倍)精度実数型の任意の式である。DDIGAM は倍精度の宣言を必要とする。

2. 引数の範囲

$$X \geq -2^{18}, D \geq -2^{50}$$

ただし、0 及び負の整数を除く。

3. エラー処理

範囲外の引数が与えられた場合は、エラーとしてメッセージを印刷し、関数値を0として計算を続行する。(FNERST p. XII参照)

(3) 計算法

1. $x < 0.5$ ならば、反転公式 $\psi(1-x) = \psi(x) + \pi \cot \pi x$ により、 $x \geq 0.5$ の場合に変換する。

2. $0.5 \leq x \leq 16$ の場合は必要に応じて漸化式 $\psi(x+1) = \psi(x) + \frac{1}{x}$ により、引数を $0.5 \leq x \leq 1.5$ の範囲に変換し、最良有理近似式 R を用いて、 $\psi(x) = R(t) + t/(1+t) - \gamma$, $t = x-1$ と計算する。

3. $x > 16$ ならば、 $\psi(x) - \log(x) + \frac{1}{2x}$ の、 $\frac{1}{x^2}$ に関する最良近似多項式を利用する。

参考文献

- 1) Handbook of Mathematical Functions, Dover, N.Y., p.253 (1970).

EI / DEI
SI / DSI
CI / DCI

Exponential, Sine and Cosine Integrals

指数，正弦及び余弦積分

作成	二宮 市三	1977年4月
形式	関数	言語 ; FORTRAN サイズ ; 33, 52, 47, 81, 47, 80行

(1) 概要

$\left. \begin{array}{l} \text{EI (DEI)} \\ \text{SI (DSI)} \\ \text{CI (DCI)} \end{array} \right\}$ は，単（倍）精度実数 x に対して $\left\{ \begin{array}{l} E_1(x) \\ S_i(x) \\ C_i(x) \end{array} \right\}$ を単（倍）精度で計算する。

ただし，

$$E_1(x) = \int_x^\infty \frac{e^{-t}}{t} dt$$

$$S_i(x) = \int_0^x \frac{\sin t}{t} dt$$

$$C_i(x) = - \int_x^\infty \frac{\cos t}{t} dt$$

(2) 使用法

1. EI (X), SI (X), CI (X)

DEI (D), DSI (D), DCI (D)

X (D) は単（倍）精度実数型の任意の式である。倍精度の関数名は倍精度の宣言を要する。

2. 引数の範囲

指数積分については $0 < X, 0 < D$

正弦積分については制限なし

余弦積分については $0 < X, 0 < D$

3. エラー処理

範囲外の引数が与えられた場合はエラーとしてメッセージを印刷し，関数値を0として計算を続行する。(FNERST p. xii参照)

(3) 計算法

1. EI (DEI)

(1) $x \leq 0$ ならばエラーとする。

(2) $0 < x < 1$ ならば $E_1(x) = f(x) - \log x$ とし， $f(x)$ を近似多項式（有理関数）で計算する。

(3) $1 \leq x \leq 180.218$ ならば $E_1(x) = e^{-x} \cdot g(x)$ とし， $g(x)$ を近似有理関数で計算する。

(4) $x > 180.218$ ならば $E_1(x) = 0$ とする。

2. SI (DSI)

(1) $|x| < 4$ ならば $S_i(x)$ を近似多項式によって計算する。

(2) $4 \leq |x| \leq 8.23 \cdot 10^5$ ($4 \leq |x| \leq 3.53 \cdot 10^{15}$)ならば $y = |x|$, $S_i(x) = \text{sign } x \cdot S_i(y)$, $S_i(y) = \frac{\pi}{2} + \left(P\left(\frac{4}{y}\right) \cos y - Q\left(\frac{4}{y}\right) \sin y \right) \frac{1}{y}$ とし, P, Q は近似多項式(有理関数)によって計算する。

(3) $|x| > 8.23 \cdot 10^5$ ($|x| > 3.53 \cdot 10^{15}$) ならば $S_i(y) = \frac{\pi}{2}$ とする。

3. CI (DCI)

(1) $x \leq 0$ ならばエラーとする。

(2) $0 < x < 4$ ならば $C_i(x) = \log x + f(x)$ とし, $f(x)$ を近似多項式によって計算する。

(3) $4 \leq x \leq 8.23 \cdot 10^5$ ($4 \leq x \leq 3.53 \cdot 10^{15}$) ならば $C_i(x) = - \left(Q\left(\frac{4}{x}\right) \cos x + P\left(\frac{4}{x}\right) \sin x \right) \frac{1}{x}$ とし, P, Q は近似多項式(有理関数)によって計算する。

(4) $x > 8.23 \cdot 10^5$ ($x > 3.53 \cdot 10^{15}$) ならば $C_i(x) = 0$ とする。

(4) 備 考

1. 指数積分には, 本ルーチンと異なる定義もあるので注意されたい。

参 考 文 献

- 1) Y.L. Luke and J. Wimp; "Jacobi polynomial Expansions of a Generalized Hypergeometric Function over a Semi-Infinite Ray", *Math. Comp.*, Vol.17 (1963).
- 2) 二宮市三; "漸近級数域での諸関数の計算法", 数理解析研究所講究録 172 (1973).

FRESS / DFRESS
FRESC / DFRESC

Fresnel Sine and Cosine Integrals
 フレネル正弦及び余弦積分

作成	二宮 市三	1977年4月
形式	関数	言語 ; FORTRAN サイズ ; 44, 16, 44, 77行

(1) 概要

FRESS (DERESS) } は, 単 (倍) 精度実数 x に対して $\left\{ \begin{matrix} S(x) \\ C(x) \end{matrix} \right\}$ を単 (倍) 精度で計算する。ただし,
 FRESC (DFRESC)

$$S(x) = \frac{1}{\sqrt{2\pi}} \int_0^x \frac{\sin t}{\sqrt{t}} dt$$

$$C(x) = \frac{1}{\sqrt{2\pi}} \int_0^x \frac{\cos t}{\sqrt{t}} dt$$

(2) 使用法

1. FRESS (X), FRESC (X)

DFRESS (D), DFRESC (D)

X (D) は単 (倍) 精度実数型の任意の式である。倍精度の関数名は倍精度の宣言を要する。

2. 引数の範囲

$$0 \leq X, \quad 0 \leq D$$

3. エラー処理

範囲外の引数が与えられた場合は, エラーとしてメッセージを印刷し, 関数値を0として計算を続行する。(FNERST p. xii参照)

(3) 計算法

1. FRESS (DFRESS)

(1) $0 \leq x < 4$ ならば $S(x) = \sqrt{x} f(x)$ とし, $f(x)$ を近似多項式によって計算する。

(2) $4 \leq x < 8.23 \cdot 10^5$ ($4 \leq x < 3.53 \cdot 10^{15}$) ならば $S(x) = \frac{1}{2} + \frac{1}{\sqrt{x}} \left(\sin x \cdot P\left(\frac{4}{x}\right) - \cos x \cdot Q\left(\frac{4}{x}\right) \right)$

とし, P, Q を近似多項式 (有理関数) によって計算する。

(3) $x < 0$ ならばエラーとする。

(4) $x > 8.23 \cdot 10^5$ ($x > 3.53 \cdot 10^{15}$) ならば $S(x) = \frac{1}{2}$ とする。

2. FRESC (DFRESC)

(1) $0 \leq x < 4$ ならば $C(x) = \sqrt{x} f(x)$ とし, $f(x)$ を近似多項式によって計算する。

(2) $4 \leq x < 8.23 \cdot 10^5$ ($4 \leq x < 3.53 \cdot 10^{15}$) ならば $C(x) = \frac{1}{2} + \frac{1}{\sqrt{x}} \left(\cos x \cdot P\left(\frac{4}{x}\right) + \sin x \cdot Q\left(\frac{4}{x}\right) \right)$

とし, P, Q を近似多項式 (有理関数) によって計算する。

(3) $x < 0$ ならばエラーとする。

(4) $x > 8.23 \cdot 10^5 (x > 3.53 \cdot 10^{15})$ ならば $C(x) = \frac{1}{2}$ とする。

COMPLEX
GAMMA-DEMO

(4) 備考

1. フレネル積分には、本ルーチンと異なる定義もあるので注意されたい。

参考文献

- 1) J. Boersma; "Computation of Fresnel Integrals", *Math. Comp.*, Vol.17 (1960).
- 2) 二宮市三; "漸近級数域での諸関数の計算法"; 数理解析研究所講究録 172 (1973)。

CND / DCND

CNDC / DCNDC

The Cumulative Normal Distribution Function and its Complement

累積正規分布関数とその余関数

作 成	二宮 市三	1977年4月 改訂	1981年4月
形 式	関数	言語; FORTRAN	サイズ; 5, 6, 5, 6行

(1) 概 要

CND (DCND) } は単 (倍) 精度実数 x に対して $\left\{ \begin{array}{l} \Phi(x) \\ \Psi(x) \end{array} \right\}$ を単 (倍) 精度で計算する。
CNDC (DCNDC) }

ただし,

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-t^2/2} dt$$

$$\Psi(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt = \frac{1}{2} - \Phi(x)$$

(2) 使用法

1. CND (X) DCND (D)

CNDC (X) DCNDC (D)

X (D) は単 (倍) 精度実数型の任意の式である。DCND, DCNDC は倍精度宣言を必要とする。

2. 引数の範囲

引数に対する制限はない。

(3) 計算法

1. CND (DCND)

$\Phi(x) = \text{erf}(x/\sqrt{2})/2$ の関係により, 基本外部関数 ERF (DERF) を使って計算する。

2. CNDC (DCNDC)

$\Psi(x) = \text{erfc}(x/\sqrt{2})/2$ の関係により, 基本外部関数 ERFC (DERFC) を使って計算する。

ACND / DACND
ACNDC / DACNDC

The Inverses of the Cumulative Normal Distribution Function and its Complement
累積正規分布関数とその余関数の逆関数

作成	二宮 市三	1977年4月	改訂	1981年4月
形式	関数	言語 ;	FORTRAN	サイズ 9, 10, 9, 10行

(1) 概要

ACND (DACND)
 ACNDC (DACNDC) } は単 (倍) 精度実数 x に対して $\left\{ \begin{matrix} \Phi^{-1}(x) \\ \Psi^{-1}(x) \end{matrix} \right\}$ を単 (倍) 精度で計算する。

ただし,

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-t^2/2} dt$$

$$\Psi(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt = \frac{1}{2} - \Phi(x)$$

であり, $\Phi^{-1}(x)$, $\Psi^{-1}(x)$ はそれぞれ $\Phi(x)$, $\Psi(x)$ の逆関数である。

(2) 使用法

1. ACND (X), DACND (D)

ACNDC (X), DACNDC (D)

X (D) は単 (倍) 精度実数型の任意の式である。DACND, DACNDC は倍精度宣言を必要とする。

2. 引数の範囲

(1) ACND, DACND

$$-\frac{1}{2} < X < \frac{1}{2}, \quad -\frac{1}{2} < D < \frac{1}{2}$$

(2) ACNDC, DACNDC

$$0 < X < 1, \quad 0 < D < 1$$

3. エラー処理

範囲外の引数を与えられた場合は, エラーとしてメッセージを印刷し, 関数値を0として計算を続行する。(FNERST p. XII参照)

(3) 計算法

1. ACND (DACND)

$\Phi^{-1}(x) = \sqrt{2} \operatorname{erf}^{-1}(2x)$ の関係により, AERF (DAERF) を引用して計算する。

2. ACNDC (DACNDC)

$\Psi^{-1}(x) = \sqrt{2} \operatorname{erfc}^{-1}(2x)$ の関係により, AERFC (DAERFC) を引用して計算する。

AERF / DAERF
AERFC / DAERFC

The Inverse of the Error Function and Its Complement

誤差関数及びその余関数の逆関数

作成	二宮 市三	1977年4月
形式	関数	言語; FORTRAN サイズ; 21, 31, 37, 63行

(1) 概要

AERF (DAERF)
AERFC (DAERFC) } は, 単(倍)精度実数 x に対して $\left\{ \begin{matrix} \operatorname{erf}^{-1} x \\ \operatorname{erfc}^{-1} x \end{matrix} \right\}$ を単(倍)精度で計算する。

ただし,

$$\operatorname{erf} x = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

$$\operatorname{erfc} x = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt$$

$$\operatorname{erf} x + \operatorname{erfc} x = 1$$

であり, $\operatorname{erf}^{-1} x$, $\operatorname{erfc}^{-1} x$ は, それぞれ, $\operatorname{erf} x$, $\operatorname{erfc} x$ の逆関数である。

(2) 使用法

1. AERF (X), AERFC (X)

DAERF (D), DAERFC (D)

X (D) は単(倍)精度実数型の任意の式である。DAERF, DAERFC は倍精度の宣言を要する。

2. 引数の範囲

逆誤差関数については, $-1 < X < 1$, $-1 < D < 1$

逆余誤差関数については, $0 < X < 2$, $0 < D < 2$

3. エラー処理

範囲外の引数が与えられた場合はエラーとしてメッセージを印刷し, 関数値を0として計算を続行する。(FNERST p. xii参照)

(3) 計算法

1. AERF (DAERF)

(1) $|x| \geq 1$ ならばエラーとする。

(2) $|x| \leq 0.75$ ならば最良有理近似式 R_1 によって $\operatorname{erf}^{-1} x = xR_1(t)$ と計算する。ただし, $t = (0.75 + x) \cdot (0.75 - x)$ 。

(3) $|x| > 0.75$ ならば AERFC (DAERFC) を引用して $\operatorname{erf}^{-1} x = \operatorname{erfc}^{-1}(1-x)$ とする。

2. AERFC (DAERFC)

(1) $x \leq 0$ 又は $x \geq 2$ ならばエラーとする。

- (2) $0.25 \leq x \leq 1.75$, すなわち $|1-x| \leq 0.75$ ならば, AERF (DAERF) を引用して $\operatorname{erfc}^{-1} x = \operatorname{erf}^{-1}(1-x)$ とする。
- (3) $0.75 < |1-x| \leq 0.9375$ ならば, 最良有理近似式 R_2 によって $\operatorname{erfc}^{-1} x = \operatorname{sign}(1-x) \cdot (1-u) \cdot R_2(t)$ と計算する。ただし, $u = \min(2-x, x)$; $t = (u-0.0625) \cdot (1.9375-u)$
- (4) $|1-x| > 0.9375$ ならば, 最良有理近似式 R_3 によって $\operatorname{erfc}^{-1} x = t \cdot R_3(t)$ と計算する。ただし, $t = \sqrt{-\log(\min(2-x, x))}$

参 考 文 献

- 1) A.J. Strecok; "On the Calculation of the Inverse of the Error Function," *Math. Comp.*, Vol.22 (1968).

PN / DPN

PNM / DPNM

Legendre and Adjoint Legendre Polynomial

ルジャンドル多項式及びルジャンドル陪多項式

作成	二宮 市三, 桑野 甯世 1977年4月
形式	関数 言語; FORTRAN サイズ; 18, 19, 31, 32行

(1) 概要

PN (DPN) は、整数 n と単 (倍) 精度実数 x に対して $P_n(x)$ を単 (倍) 精度で計算する。

PNM (DPNM) は、整数 n, m 及び単 (倍) 精度実数 x に対して $P_n^m(x)$ を単 (倍) 精度で計算する。ただし、 $P_n(x)$ はルジャンドル多項式、 $P_n^m(x)$ はルジャンドル陪多項式である。

(2) 使用法

1. PN (N, X), PNM (N, M, X)

DPN (N, D), DPNM (N, M, D)

N, M は整数型, X (D) は単 (倍) 精度実数型の任意の式である。DPN, DPNM は倍精度の宣言を要する。

2. 引数の範囲

$$0 \leq N, 0 \leq M \leq N.$$

ルジャンドル陪多項式については $|X| \leq 1, |D| \leq 1$ 。

3. エラー処理

範囲外の引数が与えられた場合は、エラーとしてメッセージを印刷し、関数値を 0 として計算を続行する。(FNERST p. xii 参照)

(3) 計算法

1. PN (DPN)

(1) $n < 0$ ならばエラーとする。

(2) $n = 0$ ならば $P_n(x) = 1$ とする。

(3) $n = 1$ ならば $P_n(x) = x$ とする。

(4) $n \geq 2$ ならば $P_0(x) = 1, P_1(x) = x$ より出発して、漸化式

$$P_k(x) = \frac{2k-1}{k} x P_{k-1}(x) - \frac{k-1}{k} P_{k-2}(x)$$

を $2 \sim n$ なる k について順次適用する。

2. PNM (DPNM)

(1) $n \geq 0, 0 \leq m \leq n$ でなければエラーとする。

(2) $m = 0$ ならば $P_n^m(x) = P_n(x)$ と計算する。

(3) $m > 0$ ならば $P_n^m(x) = (1-x^2)^{\frac{m}{2}} F_n^m(x)$ と計算する。

- (4) $n = m$ ならば $F_n^m(x) = 1 \cdot 3 \cdots (2n-1)$ とする。
 (5) $n = m+1$ ならば $F_n^m(x) = 1 \cdot 3 \cdots (2n-1) \cdot x$ とする。
 (6) $n \geq m+2$ ならば $F_n^m(x) = 1 \cdot 3 \cdots (2m-1)$, $F_{m+1}^m(x) = 1 \cdot 3 \cdots (2m+1) \cdot x$ より出発し, 漸化式

$$F_n^m(x) = \frac{2k-1}{k-m} x F_{k-1}^m(x) - \frac{k+m-1}{k-m} F_{k-2}^m(x)$$

を $m+2 \sim n$ なる k について順次適用する。

(4) 備考

本ルーチンのルジャンドル陪多項式の定義は

$$P_n^m(x) = (1-x^2)^{\frac{m}{2}} \frac{d^m P_n(x)}{dx^m}$$

である。この定義と異なる定義もあるので注意されたい。

QNOME / DQNOME

The Nome of Elliptic \wp Functions

楕円 \wp 関数のノーム

作成	二宮 市三	1981年4月
形式	関数	言語; FORTRAN サイズ; 23, 32行

(1) 概要

QNOME(DQNOME) は単(倍)精度実数 x に対して楕円 \wp 関数のノーム $q(x)$ を単(倍)精度で計算する。ただし, x は楕円関数のパラメータ $m = k^2$ を表すものとする。

(2) 使用法

1. QNOME(X), DQNOME(D)

X(D) は, 単(倍)精度実数型の任意の式である。DQNOME は倍精度の宣言を必要とする。

2. 引数の範囲

$$0 \leq X \leq 1, \quad 0 \leq D \leq 1$$

3. エラー処理

範囲外の引数が与えられた場合は, エラーとしてメッセージを印刷し, 関数値を0として計算を続行する。(FNERST p. xii参照)

(3) 計算法

1. $0 \leq x \leq 0.5$ ならば最良近似有理式によって $q(x)$ を計算する。

2. $0.5 < x \leq 1$ ならば関係式 $\log q(x) \log q(1-x) = \pi^2$ を用いて, 補ノーム $q(1-x)$ の計算に変換する。

参考文献

- 1) Handbook of Mathematical Functions, Dover, N.Y., p.591 (1970).

CELI1 / DCELI1
CELI2 / DCELI2

Complete Elliptic Integrals of the First and the Second Kind
 第1種及び第2種完全楕円積分

作成	二宮 市三	1977年4月
形式	関数	言語 ; FORTRAN サイズ ; 36, 61, 40, 64行

(1) 概要

CELI1 (DCELI1) } は, 単(倍)精度実数 x に対して $\left\{ \begin{matrix} K(x) \\ E(x) \end{matrix} \right\}$ を単(倍)精度で計算する。
 CELI2 (DCELI2) }

ただし,

$$K(x) = \int_0^{\pi/2} \frac{d\theta}{\sqrt{1-x \sin^2 \theta}}$$

$$E(x) = \int_0^{\pi/2} \sqrt{1-x \sin^2 \theta} d\theta$$

すなわち, x は通常 k^2 と書かれる値である。

(2) 使用法

1. CELI1 (X), CELI2 (X)

DCELI1 (D), DCELI2 (X)

X (D) は単(倍)精度実数型の任意の式である。DCELI1, DCELI2 は倍精度の宣言を要する。

2. 引数の範囲

第1種完全楕円積分については, $0 \leq X < 1, 0 \leq D < 1$ 。

第2種完全楕円積分については, $0 \leq X \leq 1, 0 \leq D \leq 1$ 。

3. エラー処理

範囲外の引数が与えられた場合は, エラーとしてメッセージを印刷し, 関数値を0として計算を続行する。(FNERST p. xii参照)

(3) 計算法

1. CELI1 (DCELI1)

(1) $x < 0$ 又は $x \geq 1$ ならばエラーとする。

(2) $0 \leq x < \frac{1}{2}$ ならば $y = 1 - x$ の近似有理関数によって計算する。

(3) $\frac{1}{2} \leq x < 1$ ならば $y = 1 - x, K(x) = P(y) - Q(y) \log y$ とし, P, Q を近似有理式によって

計算する。

2. CELI2 (DCELI2)

- (1) $x < 0$ 又は $x > 1$ ならばエラーとする。
- (2) $x = 1$ ならば $E(x) = 1$ とする。
- (3) $0 \leq x < \frac{1}{2}$ ならば $y = 1 - x$ の近似有理関数によって計算する。
- (4) $\frac{1}{2} \leq x < 1$ ならば $y = 1 - x$, $E(x) = P(y) - Q(y) \log y$ とし, P, Q を近似有理式によって計算する。

(4) 備考

引数 x には, 母数 k ではなくて k^2 を入れることに注意されたい。

参考文献

- 1) Handbook of Mathematical Functions, Dover, N.Y., p.587 (1970).

ICEILS / D

Incomplete Elliptic Integrals of the First and Second Kind

第1種及び第2種不完全楕円積分

作成	二宮 市三 1982年2月
形式	サブルーチン 言語 ; FORTRAN サイズ ; 74, 79行

(1) 概要

上限 x と母数 m が与えられたとき, 第1種及び第2種不完全楕円積分

$$F(x, m) = \int_0^x \frac{dt}{\sqrt{(1-t^2)(1-mt^2)}} = \int_0^\varphi \frac{d\theta}{\sqrt{1-m\sin^2\theta}}$$

$$E(x, m) = \int_0^x \sqrt{\frac{1-mt^2}{1-t^2}} dt = \int_0^\varphi \sqrt{1-m\sin^2\theta} d\theta$$

を計算する。ただし,

$$0 \leq x \leq 1, \quad 0 \leq \varphi \leq \pi/2, \quad 0 \leq m \leq 1, \quad (x = \sin \varphi)$$

(2) 使用法

CALL ICEILS/D (X, AM, F, E, CF, CE, IND)

引数	型と種類*	属性	内容
X	実数型	入力	積分の上限。 $0 \leq x \leq 1$
AM	実数型	入力	母数 $m (= k^2)$ 。 $0 \leq m \leq 1$
F	実数型	出力	第1種不完全楕円積分 $F(x, m)$ 。
E	実数型	出力	第2種不完全楕円積分 $E(x, m)$ 。
CF	実数型	出力	第1種完全楕円積分 $K(m)$ 。
CE	実数型	出力	第2種完全楕円積分 $E(m)$ 。
IND	整数型	入出力	入力: IND = 0: F と CF のみを計算する。 IND \neq 0: F, E, CF, CE を計算する。 出力: IND = 0: 正常終了。 IND = 10000: $x = 0, x = 1, m = 0, m = 1$ のいずれかであった。結果は正常である。 IND = 20000: 20回反復しても算術幾何平均法が収束しなかった。 IND = 30000: $x < 0, x > 1, m < 0, m > 1$ のいずれかであった。計算を中断する。

* ICEILD の場合には実数型をすべて倍精度実数型とする。

(3) 計算法

ガウスの算術幾何平均法を用いる。

1. $a_0 = 1, b_0 = \sqrt{1-m}, c_0 = \sqrt{m}$ から出発して, 数列 $(a_i, b_i, c_i), i = 1, 2, \dots$ を次のように生成する。

$$\begin{aligned} a_{i+1} &= (a_i + b_i) / 2, \\ b_{i+1} &= \sqrt{a_i b_i}, \\ c_{i+1} &= (a_i - b_i) / 2, \end{aligned}$$

十分な精度で $c_N \cong 0$ となったとき停止する。このとき、完全楕円積分は

$$\begin{aligned} K(m) &= \pi / (2 a_N), \\ E(m) &= K(m) \left\{ 1 - \frac{1}{2} (c_0^2 + 2 c_1^2 + 2^2 c_2^2 + \dots + 2^N c_N^2) \right\} \end{aligned}$$

で与えられる。

2. $\varphi_0 = \sin^{-1} x$ から出発して、数列 $\varphi_1, \varphi_2, \dots, \varphi_N$ を次のように生成する。

$$\tan(\varphi_{i+1} - \varphi_i) = (b_i / a_i) \tan \varphi_i, \quad i = 0, 1, \dots, N-1$$

ただし、 $\varphi_{n+1} > \varphi_n$,

$$|\varphi_{n+1} - 2\varphi_n| < \pi_0.$$

このとき、不完全楕円積分は

$$\begin{aligned} F(x, m) &= \varphi_N / (2^N a_N) \\ E(x, m) &= (E(m) / K(m)) \cdot F(x, m) + c_1 \sin \varphi_1 + \dots + c_N \sin \varphi_N \end{aligned}$$

として与えられる。

3. $m = 1$ のときは

$$\begin{aligned} CF &= 0, \quad (K(m) = \infty) \\ CE &= 1 \\ F &= \frac{1}{2} \log \frac{1+x}{1-x} = \log \left(\tan \left(\frac{\varphi}{2} + \frac{\pi}{4} \right) \right) \\ E &= x \end{aligned}$$

とする。

(4) 備考

1. 本サブルーチンは標準型の第1種及び第2種の不完全楕円積分を計算するためのもので、副産物として第1種及び第2種の完全楕円積分をも出力する。しかしながら完全楕円積分だけを計算する場合には、そのための専用関数ルーチン CELI1(DCELI1), CELI2(DCELI2)を用いる方が合理的である。
2. 本ルーチンでは、直前のコールのときの AM の値を記憶していて、続いて同じ AM が入力されると、計算法の前段 1. の部分を省略するので、なるべく AM の値を固定しておいて、X だけを変えてコールするようにするのが合理的である。
3. P を t に関する 3 次又は 4 次の多項式、 R を任意の有理式とすると、一般の楕円積分

$$\int_a^b R(\sqrt{P}) dt$$

は適当な変数変換によって、第1種 $F(x, m)$ 、第2種 $E(x, m)$ 及び第3種 $\Pi(n; x, m)$ の三つの標準型に帰着させることができる。ただし、

$$\Pi(n; x, m) = \int_0^x \frac{dt}{(1-nt^2)\sqrt{(1-t^2)(1-m^2t^2)}}$$

参考文献

- 1) Handbook of Mathematical Functions, Dover, N.Y., pp.589-626 (1970).

JACELS / D

Jacobian Elliptic Functions sn , cn , dn

ヤコビの楕円関数 sn , cn , dn

作成	二宮 市三 1977年4月
形式	サブルーチン 言語; FORTRAN サイズ; 54, 55行

(1) 概要

JACELS (JACELD) は、単(倍)精度実数 u , k^2 に対してヤコビ楕円関数 $sn(u, k^2)$, $cn(u, k^2)$, $dn(u, k^2)$ 及び第1種完全楕円積分 $K(k^2)$ を単(倍)精度で計算するサブルーチン副プログラムである。

(2) 使用法

CALL JACELS/D (U, AK, SN, CN, DN, QP, ILL)

引数	型と種類*	属性	内容
U	実数型	入力	変数 u の値を表す。 $ U \leq QP$
AK	実数型	入力	母数の二乗 k^2 の値を表す。 $0 \leq AK \leq 1$
SN	実数型	出力	sn 関数の値が入る。
CN	実数型	出力	cn 関数の値が入る。
DN	実数型	出力	dn 関数の値が入る。
QP	実数型	出力	K の値が入る。
ILL	整数型	出力	ILL = 0: U, AK についての制限が満たされ、正常に計算が行われたとき。 ILL = 30000: $AK < 0$ 又は $AK > 1$ のとき。 ILL = 1: $ U > QP$ となったとき。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 計算法

ザルツァーの算術幾何平均法¹⁾による。

(4) 備考

AK には母数 k ではなくて、 k^2 を入れることに注意されたい。

参考文献

1) H.E. Salzer; "Quick Calculation of Jacobian Elliptic Functions", *CACM*, Vol.5, p.399, (1962).

BJ0 / DJ0

BJ1 / DJ1

BY0 / DY0

BY1 / DY1

Bessel Functions of the Order 0 and 1

0 次及び 1 次のベッセル関数

作 成	二宮 市三	1977 年 4 月
形 式	関数	言語 ; FORTRAN サイズ ; 40, 78, 39, 77, 48, 91, 47, 90 行

(1) 概 要

$$\left. \begin{array}{l} \text{BJ0 (DJ0)} \\ \text{BJ1 (DJ1)} \\ \text{BY0 (DY0)} \\ \text{BY1 (DY0)} \end{array} \right\} \text{は, 単(倍)精度実数 } x \text{ に対して } \left\{ \begin{array}{l} J_0(x) \\ J_1(x) \\ Y_0(x) \\ Y_1(x) \end{array} \right\} \text{を単(倍)精度で計算する。}$$

(2) 使用法

- BJ0 (X), BJ1 (X), BY0 (X), BY1 (X)
DJ0 (D), DJ1 (D), DY0 (D), DY1 (D)

X(D) は単(倍)精度実数型の任意の式である。倍精度の関数名は倍精度の宣言を必要とする。

2. 引数の範囲

第 1 種ベッセル関数については, $0 \leq X \leq 8.23 \cdot 10^5$, $0 \leq D \leq 3.53 \cdot 10^{15}$

第 2 種ベッセル関数については, $0 < X \leq 8.23 \cdot 10^5$, $0 < D \leq 3.53 \cdot 10^{15}$

3. エラー処理

範囲外の引数が与えられた場合は, エラーとしてメッセージを印刷し, 関数値を 0 として計算を続ける。(FNERST p. xii 参照)

(3) 計算法

- $0 < x < 2$ ならば近似多項式 $P_0, P_1, Q_0, Q_1, R_0, R_1$ により,

$$J_0(x) = P_0(x^2), J_1(x) = xP_1(x^2)$$

$$Y_0(x) = Q_0(x^2) \log_2 \frac{x}{2} + R_0(x^2), Y_1(x) = xQ_1(x^2) \log_2 \frac{x}{2} + \frac{1}{x}R_1(x^2)$$

と計算する。

- $x > 2$ ならば近似有理式 S_0, S_1, T_0, T_1 により,

$$J_0(x) = \frac{1}{\sqrt{x}} \left(S_0 \left(\frac{4}{x^2} \right) \cos \left(x - \frac{\pi}{4} \right) - T_0 \left(\frac{4}{x^2} \right) \sin \left(x - \frac{\pi}{4} \right) \right)$$

$$Y_0(x) = \frac{1}{\sqrt{x}} \left(S_0 \left(\frac{4}{x^2} \right) \sin \left(x - \frac{\pi}{4} \right) + T_0 \left(\frac{4}{x^2} \right) \cos \left(x - \frac{\pi}{4} \right) \right)$$

$$J_1(x) = \frac{1}{\sqrt{x}} \left(T_1\left(\frac{4}{x^2}\right) \cos\left(x - \frac{\pi}{4}\right) + S_1\left(\frac{4}{x^2}\right) \sin\left(x - \frac{\pi}{4}\right) \right)$$

$$Y_1(x) = \frac{1}{\sqrt{x}} \left(T_1\left(\frac{4}{x^2}\right) \sin\left(x - \frac{\pi}{4}\right) - S_1\left(\frac{4}{x^2}\right) \cos\left(x - \frac{\pi}{4}\right) \right)$$

と計算する。

(4) 備考

第2種ベッセル関数 $N_0(x)$, $N_1(x)$ は、それぞれ、 $Y_0(x)$, $Y_1(x)$ と同じものである。

参考文献

- 1) A.J.M. Hitchcock; "Polynomial Approximations to Bessel Functions of Order Zero and One and to Related Functions", *MTAC*, Vol.11 (1957).
- 2) J.F. Hart; "Computer Approximations", J. Wiley (1968).

BI0 / DI0
BI1 / DI1
BK0 / DK0
BK1 / DK1

Modified Bessel Functions of the Order 0 and 1

0次及び1次の変形ベッセル関数

作成	二宮 市三	1977年4月
形式	関数	言語 ; FORTRAN サイズ ; 35, 63, 34, 61, 39, 60, 39, 60行

(1) 概要

BI0 (DI0) }
 BI1 (DI1) } は, 単(倍)精度実数 x に対して $\left\{ \begin{array}{l} I_0(x) \\ I_1(x) \\ K_0(x) \\ K_1(x) \end{array} \right\}$ を単(倍)精度で計算する。
 BK0 (DK0) }
 BK1 (DK1) }

(2) 使用法

1. BI0 (X), BI1 (X), BK0 (X), BK1 (X)

DI0 (D), DI1 (D), DK0 (D), DK1 (D)

X (D) は単(倍)精度実数型の任意の式である。

2. 引数の範囲

BI0 (X), BI1 (X), DI0 (X), DI1 (X) に対しては, $0 \leq X < 174.673$

BK0 (X), BK1 (X), DK0 (X), DK1 (X) に対しては, $0 < X < 180.218$

3. エラー処理

範囲外の引数が与えられた場合はエラーとし, エラー・メッセージを印刷し, 関数値を0として計算を続行する。(FNERST p. xii参照)

(3) 計算法

1. BI0, BI1 (DI0, DI1)

(1) $x < 0$ 又は $x \geq 174.673$ ならばエラーとする。

(2) $0 \leq x < 8$ ならば近似多項式によって計算する。

(3) $8 \leq x < 174.673$ ならば $I_n(x) = \frac{e^x}{\sqrt{x}} f\left(\frac{8}{x}\right)$ とし, f を近似多項式(有理関数)によって計算する。

2. BK0, BK1 (DK0, DK1)

(1) $x \leq 0$ ならばエラーとする。

(2) $0 < x < 2$ ならば $K_n(x) = f(x) \log x + g(x)$ とし, $f(x), g(x)$ を近似多項式によって計算する。

(3) $2 \leq x \leq 180.218$ ならば $K_n(x) = \frac{e^{-x}}{\sqrt{x}} h\left(\frac{2}{x}\right)$ とし, h を近似多項式 (有理関数) によって計算する。

(4) $x > 180.218$ ならば $K_n(x) = 0$ とする。

参 考 文 献

- 1) Y.L. Luke and J. Wimp; "Jacobi Polynomial Expansions of a Generalized Hypergeometric Functions over a Semi-Infinite Ray", *Math. Comp.*, Vol.17 (1963).
- 2) 二宮市三; "漸近級数域での諸関数の計算法", 数理解析研究所講究録 172 (1973).

SJ0 / DSJ0
 SJ1 / DSJ1
 SY0 / DSY0
 SY1 / DSY1

Spherical Bessel Functions of the Order 0 and 1
 0 次及び 1 次の球ベッセル関数

作成	二宮 市三	1977 年 4 月
形式	関数	言語 ; FORTRAN サイズ ; 18, 22, 18, 22, 19, 25, 19, 25行

(1) 概要

$\left. \begin{array}{l} \text{SJ0 (DSJ0)} \\ \text{SJ1 (DSJ1)} \\ \text{SY0 (DSY0)} \\ \text{SY1 (DSY1)} \end{array} \right\}$ は、単(倍)精度実数 x に対して $\left\{ \begin{array}{l} j_0(x) \\ j_1(x) \\ y_0(x) \\ y_1(x) \end{array} \right\}$ を単(倍)精度で計算する。

ただし、

$$j_n(x) = \sqrt{\frac{\pi}{2x}} J_{n+1/2}(x), \quad y_n(x) = \sqrt{\frac{\pi}{2x}} Y_{n+1/2}(x)$$

(2) 使用法

1. SJ0 (X), SJ1 (X), SY0 (X), SY1 (X),
 DSJ0 (D), DSJ1 (D), DSY0 (D), DSY1 (D)
 X (D) は単(倍)精度実数型の任意の式である。倍精度の関数名は倍精度の宣言を要する。

2. 引数の範囲

第1種球ベッセル関数については、 $|X| \leq 8.23 \cdot 10^5$, $|D| \leq 3.53 \cdot 10^{15}$

第2種球ベッセル関数については、 $0 < |X| \leq 8.23 \cdot 10^5$, $0 < |D| \leq 3.53 \cdot 10^{15}$

3. エラー処理

範囲外の引数が与えられた場合は、エラーとしてメッセージを印刷し、関数値を0として計算を続行する。(FNERST p. xii参照)

(3) 計算法

1. SJ0 (DSJ0)

(1) $|x| \geq 8.23 \cdot 10^5$ ($|x| \geq 3.53 \cdot 10^{15}$) ならばエラーとする。

(2) $|x| < 1$ ならば近似多項式によって $j_0(x)$ を計算する。

(3) $|x| \geq 1$ ならば $j_0(x) = \frac{\sin x}{x}$ と計算する。

2. SJ1 (DSJ1)

(1) $|x| \geq 8.23 \cdot 10^5$ ($|x| \geq 3.53 \cdot 10^{15}$) ならばエラーとする。

(2) $|x| < 1$ ならば近似多項式によって $j_1(x)$ を計算する。

(3) $|x| \geq 1$ ならば $j_1(x) = \frac{1}{x} \left(\frac{\sin x}{x} - \cos x \right)$ と計算する。

3. SY 0 (DSY 0)

(1) $x = 0$ 又は $|x| \geq 8.23 \cdot 10^5$ ($|x| \geq 3.53 \cdot 10^{15}$) ならばエラーとする。

(2) $|x| < 1$ ならば近似多項式によって $y_0(x)$ を計算する。

(3) $|x| \geq 1$ ならば $y_0(x) = -\frac{\cos x}{x}$ と計算する。

4. SY 1 (DSY 1)

(1) $x = 0$ 又は $|x| \geq 8.23 \cdot 10^5$ ($|x| \geq 3.53 \cdot 10^{15}$) ならばエラーとする。

(2) $|x| < 1$ ならば近似多項式によって $y_1(x)$ を計算する。

(3) $|x| \geq 1$ ならば $y_1(x) = -\frac{1}{x^2} (\cos x + x \sin x)$ と計算する。

(4) 備考

1. 球ベッセル関数は三角関数によって簡単に定義されるが、定義式どおり計算すると、原点付近で桁落ちのため精度の低下を招きやすい。本節の関数を使えばそのような心配はなく、また、計算時間も少なくてすむ。

2. $y_n(x)$ の代わりに $n_n(x)$ という記号が使われることもある。

S10 / DS10
 S11 / DS11
 SK0 / DSK0
 SK1 / DSK1

Modified Spherical Bessel Functions of the Order 0 and 1
 0 次及び 1 次の変形球ベッセル関数

作 成	二宮 市三	1977 年 4 月
形 式	関数	言語 ; FORTRAN サイズ ; 18, 22, 18, 22, 23, 31, 24, 31行

(1) 概 要

$$\left. \begin{array}{l} \text{S10 (DS10)} \\ \text{S11 (DS11)} \\ \text{SK0 (DSK0)} \\ \text{SK1 (DSK1)} \end{array} \right\} \text{は, 単 (倍) 精度実数 } x \text{ に対して } \left\{ \begin{array}{l} i_0(x) \\ i_1(x) \\ k_0(x) \\ k_1(x) \end{array} \right\} \text{を単 (倍) 精度で計算する。}$$

ただし,

$$i_n(x) = \sqrt{\frac{\pi}{2x}} I_{n+1/2}(x), \quad k_n(x) = \sqrt{\frac{\pi}{2x}} K_{n+1/2}(x)$$

(2) 使用法

1. S10 (X), S11 (X), SK0 (X), SK1 (X)

DS10 (D), DS11 (D), DSK0 (D), DSK1 (D)

X (D) は単 (倍) 精度実数型の任意の式である。倍精度の関数名は倍精度の宣言を要する。

2. 引数の範囲

第 1 種変形球ベッセル関数については, $|x| < 174.673$, $|D| < 174.673$

第 2 種変形球ベッセル関数については, $0 < X$, $0 < D$

3. エラー処理

範囲外の引数が与えられた場合は, エラーとしてメッセージを印刷し, 関数値を 0 として計算を続行する。(FNERST p. XII 参照)

(3) 計算法

1. S10, DS10

(1) $|x| \geq 174.673$ ならばエラーとする。

(2) $|x| < 1$ ならば近似多項式によって $i_0(x)$ を計算する。

(3) $|x| \geq 1$ ならば $i_0(x) = \frac{\sinh x}{x}$ と計算する。

2. S11, DS11

(1) $|x| \geq 174.673$ ならばエラーとする。

(2) $|x| < 1$ ならば近似多項式によって $i_1(x)$ を計算する。

(3) $|x| \geq 1$ ならば $i_1(x) = \frac{1}{x} \left(\cosh x - \frac{\sinh x}{x} \right)$ と計算する。

3. SK0, DSK0

(1) $x \leq 0$ ならばエラーとする。

(2) $x > 180.218$ ならば $k_0(x) = 0$ とする。

(3) $0 < x < 1$ ならば近似有理式によって $k_0(x)$ を計算する。

(4) $1 \leq x \leq 180.218$ ならば $k_0(x) = \frac{\pi}{2} \frac{e^{-x}}{x}$ と計算する。

4. SK1, DSK1

(1) $x \leq 0$ ならばエラーとする。

(2) $x > 180.218$ ならば $k_1(x) = 0$ とする。

(3) $0 < x < 1$ ならば近似有理式によって $k_1(x)$ を計算する。

(4) $1 \leq x \leq 180.218$ ならば $k_1(x) = \frac{\pi}{2} \left(1 + \frac{1}{x} \right) \frac{e^{-x}}{x}$ と計算する。

(4) 備考

変形球ベッセル関数は指数関数や双曲線関数によって簡単に定義されるが、定義式どおり計算すると、原点付近で桁落ちのため精度の低下を招きやすい。本節の関数プログラムを使えばそのような心配はなく、また、計算時間も少なくてすむ。

BJ010 / DJ010
 BJ011 / DJ011
 BY010 / DY010
 BY011 / DY011

Integrals of Bessel Functions

ベッセル関数の積分

作成	二宮 市三	1978年1月
形式	関数	言語 ; FORTRAN サイズ ; 46, 78, 43, 79, 55, 93, 52, 94行

(1) 概要

$$\left. \begin{array}{l} \text{BJ010 (DJ010)} \\ \text{BJ011 (DJ011)} \\ \text{BY010 (DY010)} \\ \text{BY011 (DY011)} \end{array} \right\} \text{は, 単(倍)精度実数 } x \text{ に対して}$$

$$\left\{ \begin{array}{l} \int_0^x J_0(t) dt \\ \int_x^\infty \frac{J_0(t)}{t} dt \\ \int_0^x Y_0(t) dt \\ \int_x^\infty \frac{Y_0(t)}{t} dt \end{array} \right\} \text{を単(倍)精度で計算する。}$$

(2) 使用法

1. BJ010 (X), BJ011 (X), BY010 (X), BY011 (X)
 DJ010 (D), DJ011 (D), DY010 (D), DY011 (D)

X(D) は単(倍)精度実数型の任意の式である。倍精度の関数名は倍精度の宣言を要する。

2. 引数の範囲

BJ010, BY010については, $0 \leq X \leq 8.23 \cdot 10^5$

DJ010, DY010については, $0 \leq D \leq 3.53 \cdot 10^{15}$

BJ011, BY011については, $0 < X \leq 8.23 \cdot 10^5$

DJ011, DY011については, $0 < D \leq 3.53 \cdot 10^{15}$

3. エラー処理

範囲外の引数が与えられた場合は, エラーとしてメッセージを印刷し, 関数値を0として計算を続行する。(FNERST p. xii参照)

(3) 計算法

1. $0 < x \leq 4$ ならば, 近似多項式 $P_0, Q_0, R_0, P_1, Q_1, R_1$ によって, 次のように計算する。

$$\int_0^x J_0(t) dt = xP_0(x^2)$$

$$\int_0^x Y_0(t) dt = xQ_0(x^2) + xR_0(x^2) \log \frac{x}{2}$$

$$\int_x^\infty \frac{J_0(t)}{t} dt = P_1(x^2) - \log \frac{x}{2}$$

$$\int_x^\infty \frac{Y_0(t)}{t} dt = \left(Q_1(x^2) - \frac{1}{\pi} \log \frac{x}{2} \right) \log \frac{x}{2} + R_1(x^2)$$

2. $4 < x < 8.23 \cdot 10^5$ ならば、近似有理式 S_0, T_0, S_1, T_1 によって、次のように計算する。

$$\int_0^x J_0(t) dt = \frac{1}{\sqrt{x}} \left(S_0\left(\frac{4}{x}\right) \cos\left(x - \frac{\pi}{4}\right) - T_0\left(\frac{4}{x}\right) \sin\left(x - \frac{\pi}{4}\right) \right) + 1$$

$$\int_0^x Y_0(t) dt = \frac{1}{\sqrt{x}} \left(S_0\left(\frac{4}{x}\right) \sin\left(x - \frac{\pi}{4}\right) + T_0\left(\frac{4}{x}\right) \cos\left(x - \frac{\pi}{4}\right) \right)$$

$$\int_x^\infty \frac{J_0(t)}{t} dt = x^{-\frac{3}{2}} \left(S_1\left(\frac{4}{x}\right) \cos\left(x - \frac{\pi}{4}\right) - T_1\left(\frac{4}{x}\right) \sin\left(x - \frac{\pi}{4}\right) \right)$$

$$\int_x^\infty \frac{Y_0(t)}{t} dt = x^{-\frac{3}{2}} \left(S_1\left(\frac{4}{x}\right) \sin\left(x - \frac{\pi}{4}\right) + T_1\left(\frac{4}{x}\right) \cos\left(x - \frac{\pi}{4}\right) \right)$$

BI010 / DI010
BI011 / DI011
BK010 / DK010
BK011 / DK011

Integrals of Modified Bessel Functions
 変形ベッセル関数の積分

作成	二宮 市三	1978年8月
形式	関数	言語 ; FORTRAN サイズ ; 39, 67, 40, 71, 44, 66, 42, 64行

(1) 概要

BI010 (DI010)
 BI011 (DI011)
 BK010 (DK010)
 BK011 (DK011)

は、単(倍)精度実数 x に対して

$$\left\{ \begin{array}{l} \int_0^x I_0(t) dt \\ \int_0^x \frac{I_0(t)-1}{t} dt \\ \int_0^x K_0(t) dt \\ \int_x^\infty \frac{K_0(t)}{t} dt \end{array} \right\} \text{を単(倍)精度で計算}$$

する。

(2) 使用法

- BI010 (X), BI011 (X), BK010 (X), BK011 (X)
 DI010 (D), DI011 (D), DK010 (D), DK011 (D)
 X(D) は単(倍)精度実数型の任意の式である。倍精度の関数名は倍精度の宣言を要する。
- 引数の範囲
 BI010, BI011 に対しては, $0 \leq X \leq 174.673$
 DI010, DI011 に対しては, $0 \leq D \leq 174.673$
 BK010, BK011 に対しては, $0 \leq X$
 DK010, DK011 に対しては, $0 \leq D$

3. エラー処理

範囲外の引数が与えられた場合は、エラーとしてメッセージを印刷し、関数を0として計算を続行する。(FNERST p. xii 参照)

(3) 計算法

- BI010, DI010, BI011, DI011 の場合
 (1) $x < 0$, $x > 174.673$ ならば、エラーとする。
 (2) $0 \leq x \leq 16$ ならば、近似多項式 P_0, P_1 によって、次のように計算する。

$$\int_0^x I_0(t) dt = xP_0(x^2)$$

$$\int_0^x \frac{I_0(t)-1}{t} dt = P_1(x^2)$$

- (3) $16 < x \leq 174.673$ ならば, 近似有理式 R_0, R_1 によって, 次のように計算する。

$$\int_0^x I_0(t) dt = \frac{e^x}{\sqrt{x}} R_0\left(\frac{x-16}{x}\right)$$

$$\int_0^x \frac{I_0(t)-1}{t} dt = \frac{e^x}{x^{3/2}} R_1\left(\frac{x-16}{x}\right) - \log \frac{x}{16}$$

2. BK010, DK010の場合

- (1) $x < 0$ ならば, エラーとする。
 (2) $0 \leq x \leq 2$ ならば, 近似多項式 P_0, Q_0 によって, 次のように計算する。

$$\int_0^x K_0(t) dt = xP_0(x^2) + xQ_0(x^2) \log \frac{x}{2}$$

- (3) $2 < x \leq 180.218$ ならば, 近似有理式 R_0 によって, 次のように計算する。

$$\int_0^x K_0(t) dt = \frac{\pi}{2} - \frac{e^{-x}}{\sqrt{x}} R_0\left(\frac{2}{x}\right)$$

- (4) $x > 180.218$ ならば, $\int_0^x K_0(t) dt = \frac{\pi}{2}$ とする。

3. BK011, DK011の場合

- (1) $x \leq 0$ ならば, エラーとする。
 (2) $0 < x \leq 2$ ならば, 近似多項式 P_1, Q_1 によって, 次のように計算する。

$$\int_x^\infty \frac{K_0(t)}{t} dt = \left(P_1(x^2) + \frac{1}{2} \log \frac{x}{2} \right) \cdot \log \frac{x}{2} + Q_1(x^2)$$

- (3) $2 < x \leq 180.218$ ならば, 近似有理式 R_1 によって, 次のように計算する。

$$\int_x^\infty \frac{K_0(t)}{t} dt = \frac{e^{-x}}{x^{3/2}} R_1\left(\frac{2}{x}\right)$$

- (4) $x \geq 180.218$ ならば, $\int_x^\infty \frac{K_0(t)}{t} dt = 0$ とする。

BJN / DJN

BYN / DYN

Bessel Functions of Integral Orders

整数次のベッセル関数

作 成	二宮 市三	1981年9月
形 式	関数	言語; FORTRAN サイズ; 91, 91, 46, 47行

(1) 概 要

BJN (DJN) } は、整数 n と単 (倍) 精度実数 x に対して $\left\{ \begin{matrix} J_n(x) \\ Y_n(x) \end{matrix} \right\}$ を単 (倍) 精度で計算する。
BYN (DYN)

(2) 使用法

1. BJN (X), BYN (X)

DJN (D), DYN (D)

X (D) は単 (倍) 精度実数型の任意の式である。倍精度の関数名は倍精度の宣言を要する。

2. 引数の範囲

第1種ベッセル関数 $J_n(x)$ については、

$$N \geq 0, 0 \leq X \leq 8.23 \cdot 10^5 (0 \leq D \leq 3.53 \cdot 10^{15})$$

ただし、 $X > 200$, $N > 1.384\sqrt{X}$ ($D > 200$, $N > 1.384\sqrt{D}$) を除く。

第2種ベッセル関数 $Y_n(x)$ については、

$$N \geq 0, 0 < X \leq 8.23 \cdot 10^5 (0 < D \leq 3.53 \cdot 10^{15})$$

ただし、関数値がオーバーフローとなる領域を除く。

3. エラー処理

範囲外の引数が与えられた場合には、エラーとしてエラー・メッセージを印刷し、関数値を0として計算を続行する。(FNERST p. xii参照)

(3) 計算法

1. BJN (DJN)

(1) $n < 0$, $x < 0$ 又は $x > 8.23 \cdot 10^5 (3.53 \cdot 10^{15})$ ならばエラーとする。

(2) $(x/2)^2 \leq n+1$ ならば、テーラー級数

$$J_n(x) = \sum_{k=0}^{\infty} \frac{(-1)^k (x/2)^{n+2k}}{k! \cdot (n+k)!}$$

を計算する。

(3) $x \geq 10(18)$ かつ $x \geq 0.55 n^2$ ならば、漸近展開

$$J_n(x) = \sqrt{2/\pi x} \{ P(n, x) \cos \phi - Q(n, x) \sin \phi \}$$

を利用する。ただし、

$$\phi = x - (n/2 + 1/4) \pi,$$

$$P(n, x) = 1 - \frac{(4n^2-1)(4n^2-9)}{2!(8x)^2} + \dots\dots\dots,$$

$$Q(n, x) = \frac{4n^2-1}{8x} - \frac{(4n^2-1)(4n^2-9)(4n^2-25)}{3!(8x)^3} + \dots\dots\dots$$

(4) (2), (3)以外の領域で $x \leq 200$ ならば, 漸化式

$$J_{k-1}(x) = 2kJ_k(x)/x - J_{k+1}(x)$$

を利用する。

(5) 上記以外の領域はエラーとする。

2. BYN(DYN)

(1) $n < 0$, $x \leq 0$ 又は $x > 8.23 \cdot 10^5 (3.53 \cdot 10^{15})$ ならばエラーとする。

(2) $n = 0$ ならば BY 0 (DY 0) を引用して $Y_0(x)$ を計算する。

(3) $n = 1$ ならば BY 1 (DY 1) を引用して $Y_1(x)$ を計算する。

(4) $n \geq 2$ ならば $Y_0(x)$, $Y_1(x)$ から出発して, 漸化式

$$Y_{k+1}(x) = 2kY_k(x)/x - Y_{k-1}(x)$$

を反復適用して $Y_n(x)$ を求める。

(4) 備 考

1. 0次及び1次のベッセル関数を計算するには, 本節の関数よりも, それぞれ専門の関数を用いた方が計算時間が少なく精度もよい。例えば, BJN(0, X) とするよりは, BJ 0 (X) の方がよい。
2. 第2種ベッセル関数 $N_n(x)$ は, $Y_n(x)$ と同じものである。

参 考 文 献

- 1) 二宮市三; “漸化式による Bessel 関数の計算”, 電子計算機のための数値計算法 II, 培風館 (1966)。

BJF / DJF

Bessel Functions of the First Kind of Fractional Orders

非整数次の第1種ベッセル関数

作成	二宮 市三	1981年9月
形式	関数	言語; FORTRAN サイズ; 100, 100行

(1) 概要

BJF (DJF) は単(倍)精度実数 u, x に対して $J_u(x)$ を単(倍)精度で計算する。

(2) 使用法

1. BJF (U, X), DJF (W, D)

$U(W), X(D)$ は単(倍)精度実数型の任意の式である。DJF は倍精度の宣言を要する。

2. 引数の範囲

$$U \geq 0 (W \geq 0), 0 \leq X \leq 8.23 \cdot 10^5 (0 \leq D \leq 3.53 \cdot 10^{15})$$

ただし, $X > 200, U > 1.384\sqrt{X} (D > 200, W > 1.384\sqrt{D})$ を除く。

3. エラー処理

範囲外の引数が与えられた場合は, エラーとしてメッセージを印刷し, 関数値を0として計算を続行する。(FNERST p. xii参照)

(3) 計算法

1. BJF (DJF)

(1) $u < 0, x < 0$ 又は $x > 8.23 \cdot 10^5 (3.53 \cdot 10^{15})$ ならばエラーとする。

(2) $(x/2)^2 \leq u+1$ ならば, テーラー級数

$$J_u(x) = \sum_{k=0}^{\infty} \frac{(-1)^k (x/2)^{u+2k}}{k! \Gamma(u+k+1)}$$

を計算する。

(3) $x \geq 10(18)$ かつ $x \geq 0.55 u^2$ ならば, 漸近展開

$$J_u(x) = \sqrt{2/\pi x} \{P(u, x) \cos \phi - Q(u, x) \sin \phi\}$$

を利用する。ただし,

$$\phi = x - (u/2 + 1/4)\pi,$$

$$P(u, x) = 1 - \frac{(4u^2-1)(4u^2-9)}{2!(8x)^2} + \dots,$$

$$Q(u, x) = \frac{4u^2-1}{8x} - \frac{(4u^2-1)(4u^2-9)(4u^2-25)}{3!(8x)^3} + \dots$$

(4) (2), (3)以外の領域で $x \leq 200$ ならば, 漸化式

$$J_{\nu-1}(x) = 2\nu J_{\nu}(x)/x - J_{\nu+1}(x)$$

を利用する。

(5) 上記以外の領域はエラーとする。

(4) 備考

BJF, DJF はともに計算量が多く、時間がかかるので、これらの関数プログラム以外の方法で計算できる関数には適用しない方がよい。例えば、 $J_0(x)$ は BJ0(X), BJN(0, X), BJF(0.0, X) のどれでも計算できるが、BJ0(X) が最も速く精度もよい。一般的にいて、0次と1次のベッセル関数は BJ0, BJ1(DJ0, DJ1) で計算すべきである。また、半奇数次のベッセル関数は、球ベッセル関数を経由して計算する方が有利である。例えば、 $J_{5/2}(x)$ に対しては、BJF(2.5, X) と計算するよりは、SJN(2, X) に $\sqrt{\frac{2x}{\pi}}$ を乗ずる方が合理的である。

参考文献

- 1) 二宮市三; “漸化式による Bessel 関数の計算”, 電子計算機のための数値計算法 II, 培風館(1966)。

SJN / DSJN
SYN / DSYN

Spherical Bessel Functions of Integral Orders
整数次の球ベッセル関数

作 成	二宮 市三	1981年9月
形 式	関数	言語 ; FORTRAN サイズ ; 93, 98, 46, 48行

(1) 概 要

SJN (DSJN) } は、整数 n と単 (倍) 精度実数 x に対して $\begin{Bmatrix} j_n(x) \\ y_n(x) \end{Bmatrix}$ を単 (倍) 精度で計算する。
 SYN(DSYN)

ただし、

$$j_n(x) = \sqrt{\frac{\pi}{2x}} J_{n+1/2}(x), \quad y_n(x) = \sqrt{\frac{\pi}{2x}} Y_{n+1/2}(x)$$

(2) 使用法

1. SJN (N, X), SYN (N, X)

DSJN (N, D), DSYN (N, X)

N は整数型, X (D) は単 (倍) 精度実数型の任意の式である。倍精度の関数名は倍精度の宣言を要する。

2. 引数の範囲

第1種球ベッセル関数 $j_n(x)$ については、

$$N \geq 0, \quad 0 \leq X \leq 8.23 \cdot 10^5 \quad (0 \leq D \leq 3.53 \cdot 10^{15})$$

ただし、 $X > 200$, $N > 1.384\sqrt{X}$ ($D > 200$, $N \geq 1.384\sqrt{D}$) を除く。

第2種球ベッセル関数 $y_n(x)$ については、

$$N \geq 0, \quad 0 < X \leq 8.23 \cdot 10^5 \quad (0 < X \leq 3.53 \cdot 10^{15})$$

ただし、関数値がオーバーフローとなる領域を除く。

3. エラー処理

範囲外の引数が与えられた場合は、エラーとしてメッセージを印刷し、関数値を0として計算を続行する。(FNERST p. xii 参照)

(3) 計算法

1. SJN (DSJN)

(1) $n < 0$, $x < 0$ 又は $x > 8.23 \cdot 10^5$ ($3.53 \cdot 10^{15}$) ならばエラーとする。

(2) $(x/2)^2 \leq n+3/2$ ならばテーラー級数

$$j_n(x) = \frac{x^n}{1 \cdot 3 \cdots (2n+1)} \left\{ 1 - \frac{x^2/2}{1!(2n+3)} + \frac{(x^2/2)^2}{2!(2n+3)(2n+5)} - \cdots \right\}$$

を計算する。

(3) $x \geq 10(18)$ かつ $n + \frac{1}{2} \leq 1.384\sqrt{x}$ ならば、展開式

$$j_n(x) = \{P(n, x) \sin \phi + Q(n, x) \cos \phi\} / x$$

を利用する。ただし、

$$\phi = x - n\pi/2,$$

$$P(n, x) = 1 - \frac{(\mu-1)(\mu-9)}{2!(8x)^2} + \dots,$$

$$Q(n, x) = \frac{\mu-1}{x} - \frac{(\mu-1)(\mu-9)(\mu-25)}{3!(8x)^3} + \dots,$$

$$\mu = (2n+1)^2$$

(4) (2), (3)以外の領域で $x \leq 200$ ならば、漸化式

$$j_{k-1}(x) = (2k+1)j_k(x)/x - j_{k+1}(x)$$

を利用する。

(5) 上記以外の領域はエラーとする。

2. SYN (DSYN)

(1) $n < 0$, $x \leq 0$ 又は $x > 8.23 \cdot 10^5$ ($x > 3.53 \cdot 10^{15}$) ならばエラーとする。

(2) $n = 0$ ならば SY0 (DSY0) をコールして $y_0(x)$ を計算する。

(3) $n = 1$ ならば SY1 (DSY1) をコールして $y_1(x)$ を計算する。

(4) $n \geq 2$ ならば $y_0(x)$, $y_1(x)$ から出発して、漸化式

$$y_{k+1}(x) = \frac{(2k+1)}{x} y_k(x) - y_{k-1}(x)$$

を順次適用して $y_n(x)$ を求める。

(4) 備考

第2種の球ベッセル関数 $n_n(x)$ は、 $y_n(x)$ と同じものである。

参考文献

- 1) 二宮市三; “漸化式による Bessel 関数の計算”, 電子計算機のための数値計算法 II, 培風館 (1966).

BIN / DIN
BKN / DKN

Modified Bessel Functions of Integral Orders

整数次の変形ベッセル関数

作成	二宮 市二	1981年9月
形式	関数	言語 ; FORTRAN サイズ ; 123, 183, 49, 50行

(1) 概要

BIN (DIN)
BKN(DKN) } は、整数 n と単 (倍) 精度実数 x に対して $\left\{ \begin{matrix} I_n(x) \\ K_n(x) \end{matrix} \right\}$ を単 (倍) 精度で計算する。

(2) 使用法

1. BIN (N, X), BKN (N, X)

DIN (N, D), DKN (N, D)

N は整数型, X (D) は単 (倍) 精度実数型の任意の式である。倍精度の関数名は倍精度の宣言を要する。

2. 引数の範囲

第1種変形ベッセル関数 $I_n(x)$ については、

$$0 \leq N, 0 \leq X (0 \leq D)$$

ただし、関数値がオーバーフローとなる領域を除く。

第2種変形ベッセル関数 $K_n(x)$ については、

$$0 \leq N, 0 < X (0 < D)$$

ただし、関数値がオーバーフローとなる領域を除く。

3. エラー処理

範囲外の引数が与えられた場合は、エラーとしてメッセージを印刷し、関数値を0として計算を続行する。(FNERST p. xii参照)

(3) 計算法

1. BIN (DIN)

(1) $x < 0$ 又は $n < 0$ ならば、エラーとする。

(2) $(x/2)^2 \leq n+1$ ならばテーラー級数

$$I_n(x) = \sum_{k=0}^{\infty} \frac{(x/2)^{n+2k}}{k!(n+k)!}$$

を計算する。

(3) $x \geq 10(18)$ かつ $x \geq 0.55 n^2$ ならば、漸近展開

$$I_n(x) = \frac{e^x}{\sqrt{2\pi x}} \left\{ 1 - \frac{4n^2-1}{8x} + \frac{(4n^2-1)(4n^2-9)}{2!(8x)^2} - \frac{(4n^2-1)(4n^2-9)(4n^2-25)}{3!(8x)^3} + \dots \right\}$$

を利用する。

(4) (2), (3)以外の領域で $n \geq 10(35)$ ならば、一様漸近展開

$$I_n(x) = \frac{1}{\sqrt{2\pi n}} \frac{e^{n\eta}}{(1+\xi^2)^{1/4}} \left\{ 1 + \sum_{k=1}^{\infty} n^{-k} u_k(t) \right\}$$

を利用する。ただし、

$$\xi = x/n, \quad t = 1/\sqrt{1+\xi^2}, \quad \eta = \sqrt{1+\xi^2} + \log[\xi/(1+\sqrt{1+\xi^2})]$$

であり、 $u_k(t)$ は t の k 次多項式である。

(5) 上記以外の領域では、漸化式

$$I_{k-1}(x) = 2kI_k(x)/x + I_{k+1}(x)$$

を利用する。

2. BKN (DKN)

(1) $n < 0$ 又は $x \leq 0$ ならばエラーとする。

(2) $x > 180.218$ ならば $K_n(x) = 0$ とする。

(3) $n = 0$ ならば BK0 (DK0) をコールして $K_0(x)$ を計算する。

(4) $n = 1$ ならば BK1 (DK1) をコールして $K_1(x)$ を計算する。

(5) $n \geq 2$ ならば $K_0(x)$, $K_1(x)$ から出発して、漸化式

$$K_{k+1}(x) = 2kK_k(x) \frac{1}{x} + K_{k-1}(x)$$

を順次適用して、 $K_n(x)$ を求める。

(4) 備考

0次及び1次の変形ベッセル関数を計算するには、本節の関数よりは、それぞれ専門の関数を用いた方が計算時間が少なく精度もよい。例えば、DIN(0, X)とするよりは BIO(X)の方がよい。

参考文献

- 1) 二宮市三; “漸化式による Bessel 関数の計算”, 電子計算機のための数値計算法 II, 培風館(1966)。
- 2) D.E. Amos et al.; “CDC 6600 Subroutines IBESS and JBESS for Bessel Functions $I_\nu(x)$ and $J_\nu(x)$, $x \geq 0$, $\nu \geq 0$ ”, ACM Trans. on Math. Software, Vol.3, No.1, pp.76-92 (1977)。

BIF / DIF

Modified Bessel Functions of the First Kind of Fractional Orders

非整数次の第1種変形ベッセル関数

作成	二宮 市三	1981年9月
形式	関数	言語; FORTRAN サイズ; 129, 188行

(1) 概要

BIF (DIF) は、単 (倍) 精度実数 u と x とに対して $I_u(x)$ を単 (倍) 精度で計算する。

(2) 使用法

1. BIF (U, X), DIF (W, D)

U, X (W, D) は単 (倍) 精度実数型の任意の式である。DIF は倍精度の宣言を要する。

2. 引数の範囲

$$0 \leq U (0 \leq W), \quad 0 \leq X, \quad (0 \leq D)$$

ただし、関数値がオーバーフローとなる領域を除く。

3. エラー処理

範囲外の引数が与えられた場合は、エラーとしてメッセージを印刷し、関数値を0として計算を続行する。(FNERST p. XII参照)

(3) 計算法

1. BIF (DIF)

(1) $x < 0$ 又は $u < 0$ ならばエラーとする。

(2) $(x/2)^2 \leq u+1$ ならばテーラー級数

$$I_u(x) = \sum_{k=0}^{\infty} \frac{(x/2)^{u+2k}}{k! \Gamma(u+k+1)}$$

を計算する。

(3) $x \geq 10(18)$ かつ $x \geq 0.55 u^2$ ならば、漸近展開

$$I_u(x) = \frac{e^x}{\sqrt{2\pi x}} \left\{ 1 - \frac{4u^2-1}{8x} + \frac{(4u^2-1)(4u^2-9)}{2!(8x)^2} - \frac{(4u^2-1)(4u^2-9)(4u^2-25)}{3!(8x)^3} + \dots \right\}$$

を利用する。

(4) (2), (3)以外の領域で $u \geq 10(35)$ ならば、一様漸近展開

$$I_u(x) = \frac{1}{\sqrt{2\pi u}} \frac{e^{u\eta}}{(1+\xi^2)^{1/4}} \left\{ 1 + \sum_{k=1}^{\infty} u^{-k} u_k(t) \right\}$$

を利用する。ただし、

$$\xi = x/u, \quad t = 1/\sqrt{1+\xi^2}, \quad \eta = \sqrt{1+\xi^2} + \log\{\xi/(1+\sqrt{1+\xi^2})\}$$

であり、 $u_k(t)$ は t の k 次多項式である。

(5) 上記以外の領域では、漸化式

$$I_{\nu-1}(x) = 2\nu I_{\nu}(x)/x + I_{\nu+1}(x)$$

を利用する。

(4) 備考

BIF, DIF は、ともに計算量が多く時間がかかるので、これらの関数プログラム以外の方法で計算できる関数には適用しない方がよい。例えば、 $I_0(x)$ は BIO(X), BIN(0, X), BIF(0.0, X) のどれでも計算できるが、BIO(X) が最も速く精度もよい。一般的に言って、0次と1次の変形ベッセル関数は BIO, BI1(DI0, DI1) で計算し、その他の整数次の関数は BIN(DIN) で計算すべきである。また、半奇数次のベッセル関数は、球ベッセル関数を経由して計算する方が有利である。例えば、 $I_{5/2}(x)$ に対しては、BIF(2.5, X) と計算するよりは、SIK(2, X) を求め、これに $\sqrt{\frac{2x}{\pi}}$ を乗ずる方が合理的である。

参考文献

- 1) 二宮市三; “漸化式による Bessel 関数の計算”, 電子計算機のための数値計算法Ⅱ, 培風館(1966)。
- 2) D.E. Amos, S.L. Daniel and M.K. Weston; “CDC 6600 Subroutines IBESS and JBESS for Bessel Functions $I_{\nu}(x)$ and $J_{\nu}(x)$, $x \geq 0, \nu \geq 0$ ”, ACM Trans. on Math. Software, Vol.3, No.1, pp.76-92 (1977).

SIK / DSIK

SKN / DSKN

Modified Spherical Bessel Functions of Integral Orders

整数次の変形球ベッセル関数

作成	二宮 市三	1981年9月
形式	関数	言語 ; FORTRAN サイズ ; 125, 188, 49, 50行

(1) 概要

SIK (DSIK)
SKN (DSKN) } は、整数 n と単 (倍) 精度実数 x に対して $\left\{ \begin{matrix} i_n(x) \\ k_n(x) \end{matrix} \right\}$ を単 (倍) 精度で計算する。

ただし、

$$i_n(x) = \sqrt{\frac{\pi}{2x}} I_{n+1/2}(x), \quad k_n(x) = \sqrt{\frac{\pi}{2x}} K_{n+1/2}(x)$$

(2) 使用法

1. SIK (N, X), SKN (N, X)

DSIK (N, D), DSKN (N, D)

N は整数型, X (D) は単 (倍) 精度実数型の任意の式である。倍精度の関数名は倍精度の宣言を要する。

2. 引数の範囲

第1種変型ベッセル関数 $i_n(x)$ については、

$$N \geq 0, X \geq 0 (D \geq 0)$$

ただし、関数値がオーバーフローとなる領域を除く。

第2種変型ベッセル関数については、

$$N \geq 0, X > 0 (D > 0)$$

ただし、関数値がオーバーフローとなる領域を除く。

3. エラー処理

範囲外の引数が与えられた場合は、エラーとしてメッセージを印刷し、関数値を0として計算を続行する。(FNERST p. xii参照)

(3) 計算法

1. SIK (DSIK)

(1) $x < 0$ 又は $n < 0$ ならばエラーとする。

(2) $(x/2)^2 \leq n+3/2$ ならばテーラー級数

$$i_n(x) = \frac{x^n}{1 \cdot 3 \cdots (2n+1)} \left\{ 1 + \frac{x^2/2}{1!(2n+3)} + \frac{(x^2/2)^2}{2!(2n+3)(2n+5)} + \cdots \right\}$$

を計算する。

(3) $x \geq 10(18)$ かつ $x \geq 0.55 \left(n + \frac{1}{2}\right)^2$ ならば, 漸近展開

$$i_n(x) = \frac{e^x}{2x} \left\{ 1 - \frac{\mu-1}{8x} + \frac{(\mu-1)(\mu-9)}{2!(8x)^2} - \frac{(\mu-1)(\mu-9)(\mu-25)}{3!(8x)^3} + \dots \right\}$$

を利用する。ただし, $\mu = (2n+1)^2$

(4) (2), (3)以外の領域で $n + \frac{1}{2} \geq 10(35)$ ならば, 一様漸近展開

$$i_n(x) = \frac{1}{2\sqrt{nx}} \frac{e^{(n+\frac{1}{2})\eta}}{(1+\xi^2)^{1/4}} \left\{ 1 + \sum_{k=1}^{\infty} \left(n + \frac{1}{2}\right)^{-k} u_k(t) \right\}$$

を利用する。ただし,

$$\xi = x/(n+1/2), \quad t = 1/\sqrt{1+\xi^2}, \quad \eta = \sqrt{1+\xi^2} + \log\{\xi/(1+\sqrt{1+\xi^2})\}$$

であり, $u_k(t)$ は t の k 次多項式である。

(5) 上記以外の領域では, 漸化式

$$i_{k-1}(x) = (2k+1)i_k(x)/x + i_{k+1}(x)$$

を利用する。

2. SKN (DSKN)

(1) $n < 0$ 又は $x \leq 0$ ならばエラーとする。

(2) $x > 180.218$ ならば $k_n(x) = 0$ とする。

(3) $n = 0$ ならば SK0 (DSK0) をコールして $k_0(x)$ を計算する。

(4) $n = 1$ ならば SK1 (DSK1) をコールして $k_1(x)$ を計算する。

(5) $n \geq 2$ ならば $k_0(x), k_1(x)$ から出発して, 漸化式

$$k_{l+1}(x) = \frac{(2l+1)}{x} k_l(x) + k_{l+1}(x)$$

を順次適用して, $k_n(x)$ を求める。

参 考 文 献

- 1) 二宮市三; “漸化式による Bessel 関数の計算”, 電子計算機のための数値計算法 II, 培風館 (1966).
- 2) D.E. Amos et al; “CDC 6600 Subroutines IBESS and JBESS for Bessel Functions $I_\nu(x)$ and $J_\nu(x), x \geq 0, \nu \geq 0$ ”, ACM Trans. on Math. Software, Vol.3, No.1, pp.76-92 (1977).

BESJNC / B

BESINC / B

Bessel Functions with Complex Argument

複素変数のベッセル関数

作成	吉田 年雄	1973年6月
形式	関数	言語 ; FORTRAN サイズ ; 160, 161, 148, 149行

(1) 概要

BESJNC (BESJNB) は、複素変数 z の n (整数) 次の第1種ベッセル関数を単 (倍) 精度で求める関数副プログラムである。

BESINC (BESINB) は、複素変数 z の n (整数) 次の第1種変形ベッセル関数を単 (倍) 精度で求める関数副プログラムである。

(2) 使用法

1. BESJNC (N, Z), BESJNB (N, Z)

BESINC (N, Z), BESINB (N, Z)

関数名及び引数 Z に対して、COMPLEX * 8 BESJNC, Z, 又は、COMPLEX * 16 BESJNB, Z などの宣言を必要とする。

2. 引数の範囲

(1) $J_n(z)$ に対しては、

$$|\text{Imag}(z)| \leq 174.673$$

上記以外るとき、VALUE OF BESJNC IS NOT OBTAINED N = Z = (,) というエラー・メッセージを出し、結果を (0.0,0.0) として戻る。

(2) $I_n(z)$ に対しては

$$|\text{Rea1}(z)| \leq 174.673$$

上記以外るとき、 $J_n(z)$ の場合と同様の処理をする。

(3) 計算法

漸化式を用いる方法により計算を行う。詳細は参考文献を参照。

(4) 備考

本計算は漸化式を用いる方法であるので、 $|z|$ が大きいほど計算時間を要する。したがって、 $|z| > 100$ の場合には、漸近展開式を作成して併用すれば能率的である。

参考文献

- 1) 吉田年雄; “複素変数のベッセル関数副プログラムについて— $I_n(z)$ 及び $J_n(z)$ —”, 名古屋大学大型計算機センターニュース, Vol. 5, No.3, pp. 179—185 (1974)。
- 2) 吉田年雄他; “漸化式を用いる複素変数のベッセル関数 $I_n(z)$ の数値計算”, 情報処理, Vol.14, No.1, pp. 23—29 (1973)。

AI / DAI
 AIP / DAIP
 BI / DBI
 BIP / DBIP

Airy Functions and Their Derivatives

エアリ関数とその導関数

作成	二宮 市三	1981年4月
形式	関数	言語 ; FORTRAN サイズ ; 74, 131, 72, 130, 90, 160, 93, 159行

(1) 概要

$\left\{ \begin{array}{l} \text{AI(DAI)} \\ \text{AIP(DAIP)} \\ \text{BI(DBI)} \\ \text{BIP(DBIP)} \end{array} \right\}$ は単(倍)精度実数 x に対して, エアリ関数 $\left\{ \begin{array}{l} A_i(x) \\ A'_i(x) \\ B_i(x) \\ B'_i(x) \end{array} \right\}$ を単(倍)精度で計算する。

(2) 使用法

- AI(X), DAI(D)
 AIP(X), DAIP(D)
 BI(X), DBI(D)
 BIP(X), DBIP(D)

X(D) は単(倍)精度実数型の任意の式である。DAI, DAIP, DBI, DBIP は倍精度の宣言を必要とする。

2. 引数の範囲

(1) AI, DAI, AIP, DAIP に対しては,

$$X \geq -1.15130 \cdot 10^4 \left(\frac{2}{3} |X|^{3/2} \leq 2^{18} \pi \right)$$

$$D \geq -3.04201 \cdot 10^{10} \left(\frac{2}{3} |D|^{3/2} \leq 2^{50} \pi \right)$$

(2) BI, DBI, BIP, DBIP に対しては,

$$\begin{array}{l}
 -1.15130 \cdot 10^4 \leq X \leq 40.946 \left(\begin{array}{l} X < 0 \text{ のとき } \frac{2}{3} |X|^{3/2} \leq 2^{18} \pi \\ X > 0 \text{ のとき } \frac{2}{3} |X|^{3/2} \leq 252 \log 2 \end{array} \right) \\
 -3.04201 \cdot 10^{10} \leq D \leq 40.946 \left(\begin{array}{l} D < 0 \text{ のとき } \frac{2}{3} |D|^{3/2} \leq 2^{18} \pi \\ D > 0 \text{ のとき } \frac{2}{3} |D|^{3/2} \leq 252 \log 2 \end{array} \right)
 \end{array}$$

3. エラー処理

範囲外の引数が与えられた場合は、エラーとしてメッセージを印刷し、関数値を0として計算を続行する。(FNERST p. xii 参照)

(3) 計算法

1. AI, DAI

(1) $x < -9^{1/3}$ のとき、最良近似有理式 C_1, S_1 によって

$$A_i(x) = |x|^{-1/4} (C_1(2/z) \cos z + S_1(2/z) \sin z) \text{ と計算する。ただし、} z = \frac{2}{3} |x|^{3/2}.$$

(2) $-9^{1/3} \leq x < 0$ のとき、最良近似多項式 P_1, Q_1 によって

$$A_i(x) = P_1(x^3) + |x| Q_1(x^3) \text{ と計算する。}$$

(3) $0 \leq x \leq 9^{1/3}$ のとき、最良近似有理式 R_1 によって

$$A_i(x) = R_1(x) \text{ と計算する。}$$

(4) $9^{1/3} < x < 41.808$ ($2 < \frac{2}{3} x^{3/2} < 260 \log 2$) のとき、最良近似有理式 E_1 によって

$$A_i(x) = e^{-z} |x|^{-1/4} E_1(2/z) \text{ と計算する。ただし、} z = \frac{2}{3} x^{3/2}.$$

(5) $x \geq 41.808$ のとき、 $A_i(x) = 0$ とする。

2. AIP, DAIP

(1) $x < -9^{1/3}$ のとき、最良近似有理式 C_2, S_2 によって

$$A'_i(x) = |x|^{1/4} (C_2(2/z) \cos z + S_2(2/z) \sin z) \text{ と計算する。ただし、} z = \frac{2}{3} |x|^{3/2}.$$

(2) $-9^{1/3} \leq x < 0$ のとき、最良近似多項式 P_2, Q_2 によって

$$A'_i(x) = x^2 P_2(x^3) + Q_2(x^3) \text{ と計算する。}$$

(3) $0 \leq x \leq 9^{1/3}$ のとき、最良近似有理式 R_2 によって

$$A'_i(x) = R_2(x) \text{ と計算する。}$$

(4) $9^{1/3} < x < 41.808$ のとき、最良近似有理式 E_2 によって

$$A'_i(x) = x^{1/4} e^{-z} E_2(2/z) \text{ と計算する。ただし、} z = \frac{2}{3} x^{3/2}.$$

(5) $x \geq 41.808$ のとき、 $A'_i(x) = 0$ とする。

3. BI, DBI

(1) $x < -9^{1/3}$ のとき、最良近似有理式 C_3, S_3 によって

$$B_i(x) = |x|^{-1/2} (C_3(2/z) \cos z + S_3(2/z) \sin z) \text{ と計算する。ただし、} z = \frac{2}{3} |x|^{3/2}.$$

(2) $-9^{1/3} \leq x < 0$ のとき、最良近似多項式 P_3, Q_3 によって

$$B_i(x) = P_3(x^3) + x Q_3(x^3) \text{ と計算する。}$$

(3) $0 \leq x \leq 144^{1/3}$ のとき、最良近似多項式 A_1, B_1 によって

$$B_i(x) = A_1(x^3) + x B_1(x^3) \text{ と計算する。}$$

(4) $144^{1/3} < x$ のとき、最良近似有理式 E_3 によって

$$B_i(x) = x^{-1/2} e^z E_3(1-8/z) \text{ と計算する。ただし、} z = \frac{2}{3} x^{3/2}.$$

4. BIP, DBIP

(1) $x < -9^{1/3}$ のとき、最良近似有理式 C_4, S_4 によって

- $B'_i(x) = |x|^{1/4}(C_4(2/z)\cos z + S_4(2/z)\sin z)$ と計算する。ただし、 $z = \frac{2}{3}|x|^{3/2}$ 。
- (2) $-9^{1/3} \leq x < 0$ のとき、最良近似多項式 P_4, Q_4 によって
 $B'_i(x) = x^2 P_4(x^3) + Q_4(x^3)$ と計算する。
- (3) $0 \leq x \leq 144^{1/3}$ のとき、最良近似多項式 A_2, B_2 によって
 $B'_i(x) = x^2 A_2(x^3) + B_2(x^3)$ と計算する。
- (4) $144^{1/3} < x$ のとき、最良近似有理式 E_4 によって
 $B'_i(x) = x^{1/4} e^z E_4(1-8/z)$ と計算する。ただし、 $z = \frac{2}{3}x^{3/2}$ 。

参 考 文 献

- 1) Handbook of Mathematical Functions, Dover, N.Y., p.446 (1970).

LIPS / D

Linear Programing by CRISS-CROSS Method

クリス・クロス法を用いた線形計画問題解法ルーチン

作成	赤塚 保雄 1980年11月
形式	サブルーチン 言語; FORTRAN サイズ; 2684(2770) バイト

(1) 概要

最適制御の最小化及び最大化問題をシンプレックス法を応用した CRISS-CROSS 法によって解く。LIPS は単精度用、LIPD は倍精度用ルーチンである。

Minimize
 (目的関数) $z = a_{1,1} + a_{1,2}x_1 + a_{1,3}x_2 + \dots + a_{1,n}x_{n-1}$

Subject to (条件式) $a_{2,1} + a_{2,2}x_1 + a_{2,3}x_2 + \dots + a_{2,n}x_{n-1} \geq 0$

 $a_{m,1} + a_{m,2}x_1 + a_{m,3}x_2 + \dots + a_{m,n}x_{n-1} \geq 0$

and (前定条件) $x_i (i = 1, \dots, n-1) \geq 0$

(2) 使用法

CALL LIPS/ D (A, L, M, N, IW, Y, EPS, ICON)

引数	型と種類*	属性	内容
A	実数型 2次元配列	入出力	目的関数及び条件式の係数行列を入力する。
L	整数型	入力	A の配列宣言における第1添字の値。L ≥ M
M	整数型	入力	A の行数 (条件式の数 + 1 (目的関数))。
N	整数型	入力	A の列数 (方程式の元数 + 1)。
IW	整数型 1次元配列	入出力	M + N 個の要素を持つ1次元配列名。作業領域として用いられる。
Y	実数型 1次元配列	出力	M + N - 1 個の要素を持つ1次元配列。Y ₁ : 目的関数の最小値 (最大値)。Y _{2~n} : 最適解。Y _{n+1~n+m-1} : Slack 値。
EPS	実数型	入力	計算途中で EPS より小さい値を0とみなす。
ICON	整数型	入出力	入力。 ICON = 0: 最適解が得られるまで計算を行う。 ICON ≠ 0: 掃出し計算を1回行うごとに復帰する。 出力。 ICON = 30000: 解が求まらない。 ICON = 1000: ICON ≠ 0 で CALL されたとき。 ICON = 0: 正常に解が求まった。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

(3) 備考

1. 最大化問題を解くには、目的関数 z の各係数の符号を逆にして与えればよい。ただし、 Y の値は絶対値がとられる。
2. 等号条件の問題を解くには、符号を逆にした条件式を追加するか、又は変数と条件式を追加することによって不等号条件にすればよい。

参考文献

- 1) N.K. Kwak; "Mathematical Programming with Business Applications," McGraw-Hill Book Company (1973).
- 2) T. Hu; "Integer Programming And Network Flows," ADDISON-WESLEY PUBLISHING COMPANY, (1970).

SIMPLX / SIMPLD

Linear Programming by Simplex Method

シンプレックス法を用いた線形計画法

作成	加藤 三千代 1975年1月
形式	サブルーチン 言語 ; FORTRAN サイズ ; 138, 139行

(1) 概要

線形計画法の一種であるシンプレックス法により最適制御の maximize 及び minimize 問題を解く。
制限条件

$$a_{i1}x_1 + \dots + a_{in_0}x_{n_0} \leq s_j \quad (i = 1, \dots, m_1) \quad (1)$$

$$a_{j1}x_1 + \dots + a_{jn_0}x_{n_0} \geq s_j \quad (j = 1, \dots, m_2) \quad (2)$$

$$a_{k1}x_1 + \dots + a_{kn_0}x_{n_0} = s_k \quad (k = 1, \dots, m_3) \quad (3)$$

及び

$$x_l \geq 0 \quad (l = 1, \dots, n_0) \quad (4)$$

のもとで

$$z = c_1x_1 + \dots + c_nx_n \quad (5)$$

を最大にするような x_1, \dots, x_{n_0} を求める。SIMPLX は単精度用, SIMPLD は倍精度用サブルーチンである。

(2) 使用法

CALL SIMPLX (A, KA, M1, M2, M3, N0, S, C, INDEX, OF, X, EPS, ILL)

引数	型と種類*	属性	内容
A	実数型 2次元配列	入出力	条件式の係数行列を始める N0 列にセットする。M = M1 + M2 + M3 行と N0 + M + M2 列の大きさを必要とする。
KA	整数型	入力	A の第1添字。KA ≥ M1 + M2 + M3
M1	整数型	入力	(1) 式の standard な条件式の数を M1 とする。M1 ≥ 0
M2	整数型	入力	(2) 式の逆向きの不等式の数を M2 とする。M2 ≥ 0
M3	整数型	入力	(3) 式の等式の数を M3 とする。M3 ≥ 0
N0	整数型	入力	変数の数を N0 とする。N0 ≥ 2
S	実数型 1次元配列	入力	S(M) なる1次元配列名。条件式 (1), (2), (3) の右辺をセットしておく。ただし, M = M1 + M2 + M3。
C	実数型 1次元配列	入力	C(N) なる1次元配列名。評価関数 (5) の係数をセットしておく。ただし, N = N0 + M + M2。
INDEX	整数型	入力	maximize のとき 1 を, minimize のとき -1 をセットする。
X	実数型 1次元配列	出力	X(N) なる1次元配列名。X(1), ..., X(N0) には最適解が, X(N0 + 1), ..., X(N) には slack 変数** の値が入る。
OF	実数型	出力	目的関数の最大値 (最小値) が入る。

引数	型と種類*	属性	内容
EPS	実数型	入力	計算途中で EPS より小さい値を 0 とみなす。EPS > 0
ILL	整数型	出力	入力引数が条件を満足しないとき、30000。 INDEX = 1 の場合: $z_j - c_j < 0$ で、それに対応する基底ベクトルの要素がすべて負であり、 θ^* が負のとき、1001。 INDEX = -1 の場合: $z_j - c_j > 0$ で、それに対応する基底ベクトルの要素がすべて負のとき、1002。

* 倍精度用のサブルーチンの場合は、実数型をすべて倍精度実数型とする。

** (3) の計算法を参照のこと。

(3) 計算法

ここでは、standard な条件式のみの場合 ($m = m_1$) を考える。スラック変数 $\lambda_1 \geq 0, \dots, \lambda_m \geq 0$ を導入すると、(1)式は

$$a_{i1}x_1 + \dots + a_{in_0}x_{n_0} + \lambda_i = s_i \quad (i = 1, \dots, m) \quad (6)$$

となる。 $\lambda_1, \dots, \lambda_m$ を $x_{n_0+1}, \dots, x_{n_0+m}$ とすると(6), (4), (5)式は

$$\sum_{j=0}^{n_0} a_{ij}x_j + \sum_{j=1}^m x_{n_0+j} = s_i \quad (i = 1, \dots, m) \quad (7)$$

$$x_j \geq 0 \quad (j = 1, \dots, n_0+m) \quad (8)$$

$$z = \sum_{j=1}^{n_0+m} c_j x_j \quad (j = 1, \dots, n_0+m) \quad (9)$$

(7)を $AX = S$ とすると、 A は $m \times (n_0+m)$ のマトリックス、 X は (n_0+m) の列ベクトル、 S は m の列ベクトルである。

Step 1

マトリックス A のうち m 個の基底ベクトル $a_{n_0+1}, \dots, a_{n_0+m}$ を選ぶ。 $(a_{n_0+1}, \dots, a_{n_0+m})$ の rank は m であり、

$$x_{n_0+1}a_{n_0+1} + \dots + x_{n_0+m}a_{n_0+m} = S \quad (10)$$

を満足する X は実行可能基本解である。

Step 2

初期端点を

$$X = (0, \dots, 0, x_{n_0+1}, \dots, x_{n_0+m}) \quad (11)$$

とする。 $a_{n_0+1}, \dots, a_{n_0+m}$ は一次独立であり、 a_1, \dots, a_m は

$$a_j = x_{n_0+1,j}a_{n_0+1} + \dots + x_{n_0+m,j}a_{n_0+m} \quad (j = 1, \dots, m) \quad (12)$$

と表される。(9), (11)式より

$$z_0 = c_{n_0+1}x_{n_0+1} + \dots + c_{n_0+m}x_{n_0+m} \quad (13)$$

z_j を

$$z_j = x_{n_0+1,j}c_{n_0+1} + \dots + x_{n_0+m,j}c_{n_0+m} \quad (14)$$

と仮定すれば、適正な正の数 θ に対して(10)- $\theta \times$ (12), (13)- $\theta \times$ (14)は、

$$(x_{n_0+1} - \theta x_{n_0+1,j})a_{n_0+1} + \dots + (x_{n_0+m} - \theta x_{n_0+m,j})a_{n_0+m} + \theta a_j = S \quad (15)$$

$$(x_{n_0+1} - \theta x_{n_0+1,j})c_{n_0+1} + \dots + (x_{n_0+m} - \theta x_{n_0+m,j})c_{n_0+m} + \theta c_j = z_0 - \theta(z_j - c_j) \quad (16)$$

となる。(16)式の左辺で、 $(x_{n_0+i} = \theta x_{n_0+i,j}) (i = 1, \dots, m)$ のうち一つを 0 にし、残りを非負とする

θ をみつけると、左辺は新しい評価関数 z になる。すなわち、

$$z = z_0 - \theta(z_j - c_j) \quad (17)$$

となる。 $(z_j - c_j) \leq 0$ ($j = 1, \dots, m$) ならば、 $z \geq z_0$ となり、最大化問題は改善される。また、 $(z_j - c_j) \geq 0$ となったときが求める最適解である。最小化問題は、この逆である。

Step 3

$$\theta = \min \frac{x_{n_0+i}}{x_{n_0+i,j}}$$

と選び、(15)式に代入すると

$$x'_{n_0+1}a_{n_0+1} + \dots + x'_{n_0+i-1}a_{n_0+i-1} + x'_{n_0+i+1}a_{n_0+i+1} + \dots + x'_{n_0+m}a_{n_0+m} + x'_j a_j = S \quad (18)$$

$$\begin{cases} x'_{n_0+k} = x_{n_0+k} - \theta x_{n_0+k,j} = x_{n_0+k} - \frac{x_{n_0+i} \cdot x_{n_0+k,j}}{x_{n_0+i,j}} & (k = 1, 2, \dots, i-1, i+1, m) \\ x'_j = \theta = \frac{x_{n_0+i}}{x_{n_0+i,j}} \end{cases} \quad (19)$$

x'_{n_0+k}, x'_j は正となり、端点は $X' = (0, \dots, x'_j, \dots, x'_{n_0+1}, \dots, x_{n_0+i-1}, x_{n_0+i+1}, x_{n_0+m})$ へ移動する。

(4) 使用例

6 変数の最小化問題の例を示す。

```

0000010      DIMENSION A(100,200),B(100),C(200),X(200), W(1600)
0000020      10 READ(5,1000,END=30) NS
0000030      NW=NS*80
0000040      READ(5,1010) (W(I),I=1,NW)
0000050      1000 FORMAT(I10)
0000060      1010 FORMAT(80A1)
0000070      WRITE(6,1020) (W(I),I=1,NW)
0000080      1020 FORMAT(1H ,80A1)
0000090      READ(5,1040) M1,M2,M3,NO,INDEX
0000100      M=M1+M2+M3
0000110      N=NO+M+M2
0000120      DO 20 I=1,M
0000130      READ(5,1030) (A(I,I1),I1=1,NO)
0000140      20 CONTINUE
0000150      READ(5,1030) (B(I2),I2=1,M)
0000160      READ(5,1030) (C(I2),I2=1,N)
0000170      1030 FORMAT(8F10.0)
0000180      1040  FORMAT(5I10)
0000190      EPS=1.E-5
0000200      CALL SIMPLX(A,100,M1,M2,M3,NO,B,C,INDEX,OF,X, EPS, ILL)
0000210      WRITE(6,1050) ILL,OF
0000220      WRITE(6,1060) (X(I),I=1,NO)
0000230      WRITE(6,1070) (C(I),I=1,N)
0000240      GO TO 10
0000250      30 STOP
0000260      1050 FORMAT(1H0,' ILL=',I5,' OBJECT FUNCTION VALUE IS' E15.6)
0000270      1060 FORMAT(1H ,5E12.5)
0000280      1070 FORMAT(1H , 'ZJ-CJ'/5E12.5)
0000290      END

```

入力例

```

6
*** SUBJECT TO
X1+3X2-X3+2X5>=7.0
-2X2+4X3+X4 >=12
-4X2+3X3+8X5+X6>=10
AND X(1<=I<=6)>=0
MINIMIZE F(X)=X2-3X3+2X5

```

	3		6	-1	
1.0	3.0	-1.0	0.0	2.0	0.0
0.0	-2.0	4.0	1.0	0.0	0.0
0.0	-4.0	3.0	0.0	8.0	1.0
7.0	12.0	10.0			
0.0	1.0	-3.0	0.0	2.0	0.0

出力例

```

*** SUBJECT TO
X1+3X2-X3+2X5>=7.0
-2X2+4X3+X4 >=12
-4X2+3X3+8X5+X6>=10
AND X(1<=I<=6)>=0
MINIMIZE F(X)=X2-3X3+2X5
ILL= 0 OBJECT FUNCTION VALUE IS -0.110000E+02
0.0 0.40000E+01 0.50000E+01 0.0 0.0
0.0
ZJ-CJ
0.20000E+00 0.0 0.0 -0.80000E+00-0.24000E+0
ZJ-CJ
0.0 -0.20000E+00-0.80000E+00 0.0

```

(5) 備考

変数の数に比べて条件式が多い時双対問題にするとよい。

例 SUBJECT to

$$3u_2 \geq 1$$

$$2u_1 + 4u_2 \geq 4$$

$$2u_1 + u_2 \geq 3$$

$$3u_1 \geq 3$$

$$4u_1 + 2u_2 \geq 4$$

$$u_1 + 2u_2 \geq 1$$

and $u_i (i = 1, 2) \geq 0$ の下で minimize $g = 2u_1 + 2u_2$ は

$$2x_2 + 2x_3 + 3x_4 + 4x_5 + x_6 \leq 2$$

$$3x_1 + 4x_2 + x_3 + 2x_5 + 2x_6 \leq 2$$

and $x_i (i = 1, 2, \dots, 6) \geq 0$ の下で maximize $f = x_1 + 4x_2 + 3x_3 + 3x_4 + 4x_5 + x_6$ とおきかえることができる。

$$u_1 = 4/3, u_2 = 1/3$$

$$x_2 = 1/3, x_3 = 2/3$$

$$z = g(u) = f(x) = 10/3$$

M-TYPE

Multiple Precision Arithmetic with Error Estimation

(M-TYPE; Machine Language Like Arithmetic)

誤差評価の可能な多重精度演算 (M型)

作 成	大中 幸三郎, 安井 裕 1973 年, 改訂 大中 幸三郎 1979 年
形 式	サブルーチン 言語; FORTRAN サイズ; 1,130 行
利用者の義務	プログラム名と作者名を明記する。

(1) 概 要

数値計算によって得られた解の確度, 言い換えれば誤差の評価は, 数値計算上の重要な事柄である。解析的な評価も重要ではあるが, どのような計算過程についても誤差の振舞いを知ろうとするときには, 計算過程の個々について何らかの形で直接的に誤差の影響をつかんだ演算を行わねばならない。したがって, 我々は誤差評価を可能とする機能として, 丸めの制御, 誤差項の評価, interval arithmetic 等を行う機能をもつ一連の多重精度演算パッケージ (M, S, E, I 型) を作成した。

M型はS, E, I型の基本となる演算であり, もっとも自由度の高い記述のできる方式である。この型の誤差評価機能としては, 丸めの制御と桁落ちのインディケータを持っている。

(2) 数の表現

M型の多重精度の数は整数型一次元配列を用いて表わされる。その例として π を10進25桁について示すと下図のようになる。

配列名

MX

(1)	1
(2)	0
(3)	3
(4)	1415
(5)	9265
(6)	3589
(7)	7932
(8)	3846
(9)	2643

符号部 $MX(1) = \begin{cases} 1 & \text{正数} \\ 0 & \text{第2要素以下の内容にかかわらずゼロ} \\ -1 & \text{負数} \end{cases}$

指数部
 整数部
 小数部 } 仮数部

$$MX = 3.1415 \dots 2643(10^4)^0$$

上図は仮数部の配列一要素に10進で4桁 (一般には10進で n 桁, 又は 10^n 進1桁と考えてもよい)はいるときの図である。そして, 指数部は 10^4 毎 (一般には 10^n 毎)の指数を表わしている。したがってMX(2) = 1ならばMX = 31415.26...を表わすことになる。また, 配列の長さは上図では9であるが, 一般にはプリセットパラメータで与えられる。

(3) プリセットパラメータと宣言文

利用者が与えるプリセットパラメータは NN, JTE, JDE, JDM, JR, KM, NW の 7 個 (いずれも整数型) であって, その役割と条件は次の通りである。

NN : 仮数部の配列一要素にはいる10進の桁数を示す n 。 $1 \leq NN \leq 4$ 。

JTE : 指数部の上限。 $| \text{指数部} | < JTE$ 。 $3 \leq 3 \times NN < JDE < JTE < 2^{30}$ 。

ただし, JTE に負の値を与えるとデフォルト値 10^8 とみなされる。

JDE : 倍精度実数型の指数部の上限。 $10^{-JDE} < |x| < 10^{JDE}$ となる x が倍精度実数型で表現できること。 $3 \leq 3 \times NN < JDE < JTE < 2^{30}$ 。

ただし, JDE に負の値を与えるとデフォルト値 75 とみなされる。

JDM : 倍精度実数型の仮数部の有効桁数。 $2 \leq 2 \times NN < JDM \leq 15$ 。

ただし, JDM に負の値を与えるとデフォルト値 15 とみなされる。

JR : 丸め処理を定めるパラメータ。 $0 \leq JR \leq 10$ 。

ただし, JR に負の値を与えるとデフォルト値 5 とみなされる。

JR = 0 切り上げ → 小数点以下 1 位で切り上げて 10 となったとすれば, 真の切り

1 0 捨 1 入 上げは $9 < x \leq 10$ であるが, JR = 0 のときは $9 \leq x < 10$ となり, 多小異なる。

• •

• •

5 4 捨 5 入

• •

• •

9 8 捨 9 入

10 切り捨て

KM : M 型の多重精度の数を表わす整数型一次元配列の長さ。 $7 \leq KM < 2^{30}$ 。

NW : WRITE 文の装置番号。エラーの印字のときに必要である。

これらのプリセットパラメータから計算される諸定数などの情報の受け渡しにはコモン領域を用いている。したがって, 利用者は主プログラムで次の宣言をする必要がある。

DOUBLE PRECISION DIR

COMMON/BMSEI1/DIR, K, KK, KKK, KMAX, KMAX1, KMAX2, KMAX3, KMAXW,
KMAXW2, IR1, KETA, IND, IERROR, NW

ただし, コモン領域中の DIR 以外はすべて整数型である。また, M 型で使用されるものは IR1, KETA, IERROR, NW のみであり, 他は S, E, I 型の多重精度演算と混用するときに意味をもつ。

(4) 使用法

引 数	型 と 種 類	属 性	内 容
MACC1	整 数 型 1 次 元 配 列	入 出 力	M 型の数を表わす配列。ただし, サブルーチン CVLNG では出力のみ。大きさ KM。

引数	型と種類	属性	内容
MACC2	整数型 1次元配列	入力	M型の数を表わす配列。大きさ KM。
MACCW	整数型 1次元配列	作業領域	大きさ KKM。ただし、 $KKM \equiv 2 \times KM - 2$ 。
INT	整数型	入力	$ INT < 10^{NN}$ 。
DX	倍精度実数型	入力又は出力	サブルーチン CVSHO では出力。 サブルーチン CVLNG では入力。
KKM	整数型	入出力	$KKM \equiv 2 \times KM - 2$ 。 サブルーチン START では出力。 サブルーチン MULT1, DIVI1 では入力。
NN, JTE JDE, JDM JR, KM	整数型	入力	プリセットパラメータ。(3)参照。

1. M型の多重精度演算ルーチンを使用するに際し必須のもの

CALL START(NN, JTE, JDE, JDM, JR, KM, KKM)

(3)に述べたプリセットパラメータから計算される諸定数の計算と条件の判定、並びにコモン領域中の KETA (桁落ちのインディケータ) と IERROR (エラーインディケータ) のゼロクリアを行う。プリセットパラメータを定めた後、他のサブルーチンを呼ぶ前に必ずこのルーチンを呼ぶこと。

2. M型の多重精度の二数の四則演算

CALL ADD(MACC1, MACC2, KM)

$MACC1 \leftarrow MACC1 + MACC2$

桁落ちのインディケータ KETA (コモン領域) がセットされる。このインディケータは 10^n 進法で桁落ちした桁数を示す。M型で KETA がセットされるのはこのルーチンと次に示すサブルーチン SUB だけである。このルーチンは内部で SUB を呼んでいる。

CALL SUB(MACC1, MACC2, KM)

$MACC1 \leftarrow MACC1 - MACC2$

このルーチンは内部でサブルーチン ADD を呼んでいる。また、KETA については ADD と同一である。

CALL MULT1(MACC1, MACC2, MACCW, KM, KKM)

$MACC1 \leftarrow MACC1 \times MACC2$

CALL DIVI1(MACC1, MACC2, MACCW, KM, KKM)

$MACC1 \leftarrow MACC1 / MACC2$

3. M型の多重精度の数と整数の乗除算

CALL MULT2(MACC1, INT, KM)

$MACC1 \leftarrow MACC1 \times INT$

CALL DIVI2(MACC1, INT, KM)

MACC1 ← MACC1/INT

4. M型の多重精度の数と倍精度実数の変換

CALL CVSHO (MACC2, DX, KM)

DX ← MACC2

CALL CVLNG (DX, MACC1, KM)

MACC1 ← DX

5. スレーブサブルーチン

CALL ERRMS (NS1, NS2)

エラーが発生したときの印字ルーチンであり、エラーの発生したサブルーチン名とエラーインディケータを印字し、実行を停止する。利用者は直接にこのルーチンを呼ぶことはない。2., 3., 4.のサブルーチンから呼ばれる。引数の説明は省略。

(5) エラーインディケータ

コモン領域中のエラーインディケータ IERROR の値とその意味を示す。

0 : 正常処理

負 : 正常処理。E型の多重精度演算と混用するか、利用者がIERRORの値を負に再定義しないかぎり、負とならない。

10 : オーバーフロー (ADD, SUB, MULT1, DIVI1, MULT2, DIVI2)。

11 : アンダーフロー (ADD, SUB, MULT1, DIVI1, MULT2, DIVI2)。

20 : 不定 (DIVI1)。

21 : 不能 (DIVI1)。

30 : 不定 (DIVI2)。

31 : 不能 (DIVI2)。

40 : $|INT| \geq 10^{NN}$ 。

50 : オーバーフロー (CVSHO)。

51 : アンダーフロー (CVSHO)。

90 : プリセットパラメータエラー。ただしチェック項目は次の通り。

$3 \leq 3 \times NN < JDE < JTE, \quad 2 \times NN < JDM, \quad 0 \leq JR \leq 10, \quad 7 \leq KM。$

(6) 使用例

```
//MTYPE JOB
// EXEC FORTHCG,PARM.FORT='S',PRVLIB='CENT.MPRECIS'
//FORT.SYSIN DD *
C *** SPECIFICATION STATEMENT ***
  DIMENSION MACC1(13),MACC2(13),MACCW(24)
  DOUBLE PRECISION DIR,X,Z
  COMMON /BMSEI1/DIR,K,KK,KKK,KMAX,KMAX1,KMAX2,KMAX3,KMAXW,KMAXW2,
  1 IR1,KETA,IND,IERROR,NW
C *** PARAMETER SET ***
  NW=6
  NR=5
  NN=4
  JTE=10**8
  JDE=75
  JDM=15
  JR=5
  KM=13
```

```

CALL START(NN,JTE,JDE,JDM,JR,KM,KKM)
C *** TEST OF ADD ***
IERROR=0
READ(NR,9000) (MACC1(I),I=1,KM)
READ(NR,9000) (MACC2(I),I=1,KM)
WRITE(NW,2000) (MACC1(I),I=1,KM)
WRITE(NW,2100) (MACC2(I),I=1,KM)
CALL ADD(MACC1,MACC2,KM)
IF(IERROR.NE.0) GO TO 100
WRITE(NW,2000) (MACC1(I),I=1,KM)
WRITE(NW,2100) (MACC2(I),I=1,KM)
WRITE(NW,2200) KETA
100 WRITE(NW,2400)
2000 FORMAT(6HOMACC1,I6,I12,11I6)
2100 FORMAT(6H MACC2,I6,I12,11I6)
2200 FORMAT(5H KETA,I7)
2400 FORMAT(1H ,100(1H-))
9000 FORMAT(I2,I13,11I4)
STOP
END
//GO.SYSIN DD *
-1      1000043219876543219876543219876543219876543219876
-1      1000067891234567891234567891234567891234567891234
//

```

<出力例>

MACC1	-1	10000	4321	9876	5432	1987	6543	2198	7654	3219	8765
MACC2	-1	10000	6789	1234	5678	9123	4567	8912	3456	7891	2345
MACC1	-1	10001	1	1111	1111	1111	1111	1111	1111	1111	1111
MACC2	-1	10000	6789	1234	5678	9123	4567	8912	3456	7891	2345
KETA	0										

(7) プリセットパラメータの変更

プリセットパラメータの値を変更する場合は原則としてサブルーチン START を呼ぶ必要があるが、JR, KM, NW の変更の場合には、各パラメータが(3)に述べた条件を満足するかぎり、START を呼ぶ必要はない。各々の変更手順を次に示す。

JR : コモン領域中の $IR1(= JR \times 10^{NN-1})$ を変更する。ただし、JR のデフォルト機能は使用できない。

KM : KM と KKM ($= 2 \times KM - 2$) を変更する。演算毎に KM と KKM を変化させて、実行時間の短縮を行うことができる。

NW : コモン領域中の NW を変更する。

(8) 備考

1. コモン領域中の KETA と IERROR の値を利用者が再定義しても実行に影響はないが、IERROR が正のときはゼロにリセットされる。ただし、E型の多重精度演算と混用する場合の IERROR の再定義には別の意味がある。(E型参照)。
2. コモン領域の名前は(3)で述べた BMSEI1の他に BMSEI2を用いている。したがって、BMSEI2 は予約語であり、利用者は使用できない。
3. S, E, I 型の多重精度演算と混用する場合は、それぞれに必要なプリセットパラメータを定めた後、各型のスタートルーチンを呼べばよい。
4. 引数 MACC1, MACC2が(2)に述べた表現を満足するか否かのチェックは行っていない。

参考文献

- 1) 大中幸三郎, 安井裕; “誤差評価の可能な多重精度演算”, 情報処理, Vol.15, pp.109~117 (1974)。

S-TYPE

Multiple Precision Arithmetic with Error Estimation

(S-TYPE; Symbolic Language Like Arithmetic)

誤差評価の可能な多重精度演算 (S型)

作成	大中 幸三郎, 安井 裕 1973年, 改訂 大中 幸三郎 1979年
形式	サブルーチン 言語; FORTRAN サイズ; 1,403行
利用者の義務	プログラム名と作者名を明記する。

(1) 概要

数値計算によって得られた解の確度, 言い換えれば誤差の評価は, 数値計算上の重要な事柄である。解析的な評価も重要ではあるが, どのような計算過程についても誤差の振舞いを知ろうとするときには, 計算過程の個々について何らかの形で直接的に誤差の影響をつかんだ演算を行わねばならない。したがって, 我々は誤差評価を可能とする機能として, 丸めの制御, 誤差項の評価, interval arithmetic 等を行う機能をもつ一連の多重精度演算パッケージ (M, S, E, I型) を作成した。

S型は引数として, 2個のオペランドと演算結果のみをもち, M型よりも容易に利用できるばかりでなく, 数の表現に必要な語数を節約している。この型の誤差評価機能はM型と全く同一であり, 丸めの制御と桁落ちのインディケータである。

(2) 数の表現

S型の多重精度の数はM型の場合と同様に整数型一次元配列を用いて表わされる。M型との相異は小数部一要素に10進2n桁 (M型はn桁) を入れたことである。したがって, M型と同じくπを10進25桁について示すと下図のようになる。

配列名

L X			
(1)	1	符号部	LX(1) = $\begin{cases} 1 & \text{正数} \\ 0 & \text{第2要素以下の内容にかかわらずゼロ} \\ -1 & \text{負数} \end{cases}$
(2)	0	指数部	
(3)	3	整数部	
(4)	14159265	} 小数部	} 仮数部
(5)	35897932		
(6)	38462643		
			LX = 3.1415 ... 2643(10 ⁴) ⁰

上図は $n = 4$ の場合の図である。配列の第一要素 (符号部), 第二要素 (指数部), 第三要素 (整数部) の定義はM型と同一である。したがって, 指数部は10²ⁿ ごとではなく, 10ⁿ ごとの指数を表わし, 整数部には10進n桁 (10ⁿ進1桁), 小数部には10進2n桁 (10ⁿ進2桁) はいることに注意する必要がある。

(3) プリセットパラメータと宣言文

利用者が与えるプリセットパラメータは NN, JTE, JDE, JDM, JR, K, NW の7個 (いずれも整数型) であるが, Kをのぞく6個はM型と同一である。

NN : 仮数部の配列一要素にはいる10進の桁数を示す n 。 $1 \leq NN \leq 4$ 。

JTE : 指数部の上限。 $| \text{指数部} | < JTE$ 。 $3 \leq 3 \times NN < JDE < JTE < 2^{30}$ 。

ただし、JTE に負の値を与えるとデフォルト値10⁸とみなされる。

JDE : 倍精度実数型の指数部の上限。 $10^{-JDE} < |x| < 10^{JDE}$ となる x が倍精度実数型で表現できること。 $3 \leq 3 \times NN < JDE < JTE < 2^{30}$ 。

ただし、JDE に負の値を与えるとデフォルト値75とみなされる。

JDM : 倍精度実数型の仮数部の有効桁数。 $2 \leq 2 \times NN < JDM \leq 15$ 。

ただし、JDM に負の値を与えるとデフォルト値15とみなされる。

JR : 丸め処理を定めるパラメータ。 $0 \leq JR \leq 10$ 。

ただし、JR に負の値を与えるとデフォルト値5とみなされる。

JR = 0 切り上げ → 小数点以下1位で切り上げて10となったとすれば、真の切り

1 0捨1入 上げは $9 < x \leq 10$ であるが、JR = 0 のときは $9 \leq x < 10$ となり、多少異なる。

・ ・

5 4捨5入

・ ・
・ ・

9 8捨9入

10 切り捨て

K : S型の多重精度の数を表わす整数型一次元配列の長さ。 $5 \leq K \leq 103$ 。

NW : WRITE 文の装置番号。エラーの印字のときに必要である。

これらのプリセットパラメータから計算される諸定数などの情報の受渡しにはコモン領域を用いている。したがって、利用者は主プログラムでM型の場合と全く同一の次の宣言をする必要がある。

DOUBLE PRECISION DIR

COMMON/BMSEI1/DIR,K, KK, KKK, KMAX, KMAX1, KMAX2, KMAX3, KMAXW,
KMAXW2, IR1, KETA, IND, IERROR, NW

ただし、コモン領域中の DIR 以外はすべて整数型である。また、S型で使用されるものは K, KK, KKK, IR1, KETA, IERROR, NW のみであり、他はE型の多重精度演算と混用するときに意味をもつ。

(4) 使用法

引 数	型 と 種 類	属 性	内 容
LX, LY	整 数 型 1 次 元 配 列	入 力	S型の数を表わす配列。大きさK。
LZ	整 数 型 1 次 元 配 列	出 力	S型の数を表わす配列。演算結果が入る。大きさK。
INT	整 数 型	入 力	$ INT < 10^{NN}$ 。

引 数	型 と 種 類	属 性	内 容
DX	倍精度実数型	入力又は出力	サブルーチン CVSHOS では出力。 サブルーチン CVLNGS では入力。
NN, JTE, JDE, JDM, JR	整数型	入力	プリセットパラメータ。(3)参照。

1. S型の多重精度演算ルーチンを使用するに際し必須のもの

CALL STARTS(NN, JTE, JDE, JDM, JR)

(3) に述べたプリセットパラメータから計算される諸定数の計算と条件の判定、並びにコモン領域中の KETA (桁落ちのインディケータ) と IERROR (エラーインディケータ) のゼロクリアを行う。プリセットパラメータを定めた後、他のサブルーチンを呼ぶ前に必ずこのルーチンを呼ぶこと。

2. S型の多重精度の二数の四則演算

CALL ADDS(LX, LY, LZ)

$LZ \leftarrow LX + LY$

桁落ちのインディケータ KETA (コモン領域) がセットされる。このインディケータは 10^n 進法で桁落ちした桁数を示す。S型で KETA がセットされるのはこのルーチンと次に示すサブルーチン SUB だけである。このルーチンは内部で SUB を呼んでいる。

CALL SUBS(LX, LY, LZ)

$LZ \leftarrow LX - LY$

このルーチンは内部でサブルーチン ADD を呼んでいる。また、KETA については ADD と同一である。

CALL MULT1S(LX, LY, LZ)

$LZ \leftarrow LX \times LY$

CALL DIV1S(LX, LY, LZ)

$LZ \leftarrow LX / LY$

3. S型の多重精度の数と整数の乗除算

CALL MULT2S(LX, INT, LZ)

$LZ \leftarrow LX \times INT$

CALL DIVI2S(LX, INT, LZ)

$LZ \leftarrow LX / INT$

4. S型の多重精度の数と倍精度実数の変換

CALL CVSHOS(LX, DX)

$DX \leftarrow LX$

CALL CVLNGS(DX, LZ)

5. スレーブサブルーチン

CALL ERRMS(NS1, NS2)

エラーが発生したときの印字ルーチンであり、エラーの発生したサブルーチン名とエラーイン

ディケータを印字し、実行を停止する。利用者は直接にこのルーチンと呼ぶことはない。2, 3, 4のサブルーチンから呼ばれる。M型のサブルーチン ERRMS と全く同一である。引数の説明は省略。

(5) エラーインディケータ

コモン領域中のエラーインディケータ IERROR の値とその意味を示す。

0 : 正常処理。

負 : 正常処理。E型の多重精度演算と混用するか、利用者が IERROR の値を負に再定義しないかぎり、負とならない。

10 : オーバーフロー (ADDS, SUBS, MULT1S, DIVI1S, MULT2S, DIVI2S)。

11 : アンダーフロー (ADDS, SUBS, MULT1S, DIVI1S, MULT2S, DIVI2S)。

20 : 不定 (DIVI1S)。

21 : 不能 (DIVI1S)。

30 : 不定 (DIVI2S)。

31 : 不能 (DIVI2S)。

40 : $|INT| \geq 10^{NN}$ 。

50 : オーバーフロー (CVSHOS)。

51 : アンダーフロー (CVSHOS)。

90 : プリセットパラメータエラー。ただしチェック項目は次の通り。

$3 \leq 3 \times NN < JDE < JTE, 2 \times NN < JDM, 0 \leq JR \leq 10, 5 \leq K$ 。

(6) 使用例

```
//STYPE JOB
// EXEC FORTHCG,PARM.FORT='S',PRVLIB='CENT.MPRECIS'
//FORT.SYSIN DD *
C *** SPECIFICATION STATEMENT ***
  DIMENSION LX(8),LY(8),LZ(8)
  DOUBLE PRECISION DIR,X,Z
  COMMON /BMSI1/DIR,K,KK,KKK,KMAX,KMAX1,KMAX2,KMAX3,KMAXW,KMAXW2,
  1      IR1,KETA,IND,IERROR,NW
C *** PARAMETER SET ***
  NW=6
  NR=5
  NN=4
  JTE=10**8
  JDE=75
  JDM=15
  JR=5
  K=8
  CALL STARTS(NN,JTE,JDE,JDM,JR)
C *** TEST OF ADDS ***
  IERROR=0
  READ(NR,9000) (LX(I),I=1,K)
  READ(NR,9000) (LY(I),I=1,K)
  WRITE(NW,2000) (LX(I),I=1,K)
  WRITE(NW,2100) (LY(I),I=1,K)
  CALL ADDS(LX,LY,LZ)
  IF(IERROR.NE.0) GO TO 100
  WRITE(NW,2000) (LX(I),I=1,K)
  WRITE(NW,2100) (LY(I),I=1,K)
  WRITE(NW,2800) (LZ(I),I=1,K)
  WRITE(NW,2200) KETA
2000 FORMAT(3HOLX,I6,7I12)
```

```

2100 FORMAT(3H LY,I6,7I12)
2200 FORMAT(5H KETA,I7)
2800 FORMAT(3H LZ,I6,7I12)
9000 FORMAT(I2,I13,I5,5I8)
100 CONTINUE
STOP
END
//GO.SYSIN DD *
-1      10000 4321 987654321 987654321 987654321 987654321
-1      10000 6789 123456789 123456789 123456789 123456789
//

```

<出力例>

```

LX  -1      10000      4321      9876543      21098765      43210987      65432109
LY  -1      10000      6789      1234567      89012345      67890123      45678901

LX  -1      10000      4321      9876543      21098765      43210987      65432109
LY  -1      10000      6789      1234567      89012345      67890123      45678901
LZ  -1      10001              1      11101111      11111011      11111110      11111111
KETA      0

```

(7) プリセットパラメータの変更

プリセットパラメータの値を変更する場合は原則としてサブルーチン STARTS を呼ぶ必要があるが、JR, K, NW の変更の場合には、各パラメータが(3)に述べた条件を満足するかぎり、STARTS を呼ぶ必要はない。JR と NW の変更手順はM型の多重精度演算の場合と同一である。

JR : コモン領域中の IR1 ($\equiv JR \times 10^{NN-1}$) を変更する。ただし、JR のデフォルト機能は使用できない。

K : コモン領域中の K, KK ($\equiv 2 \times K - 3$), KKK ($\equiv 4 \times K - 8$) を変更する。演算毎に K, KK, KKK を変化させて、実行時間の短縮を行うことができる。

NW : コモン領域中の NW を変更する。

(8) 備考

1. コモン領域中の KETA と IERROR の値を利用者が再定義しても実行に影響はないが、IERROR が正のときはゼロにリセットされる。ただし、E型の多重精度演算と混用する場合の IERROR の再定義には別の意味がある (E型参照)。
2. コモン領域の名前は(3)で述べた BMSEI1の他に BMSEI2と BWRKS を用いている。したがって、BMSEI2と BWRKS は予約語であり、利用者は使用できない。
3. M, E, I 型の多重精度演算と混用する場合は、それぞれに必要なプリセットパラメータを定めた後、各型のスタートルーチンと呼ばばよい。
4. 引数 LX, LY が(2)に述べた表現を満足するか否かのチェックは行っていない。
5. 演算ルーチンと呼ぶときの引数の配列名は各々異なる名でなくてもよい。

```

例 CALL ADDS(LX, LY, LX)          LX ← LX+LY
    CALL MULT1S(LX, LX, LX)       LX ← LX×LX

```

参考文献

- 1) 大中幸三郎, 安井裕; “誤差評価の可能な多重精度演算”, 情報処理, Vol.15, pp.109~117 (1974)。

内容の定義からわかるように、ゼロに対して誤差は認めず、ゼロには絶対的なゼロしかないものとして取扱う。指数部、整数部、小数部の定義はM型の場合と同一である。除算における誤差項の計算のために相対誤差を1に近くすることはできない。したがって、 $b \geq 4$ の条件が必要である。

(3) プリセットパラメータと宣言文

利用者が与えるプリセットパラメータは NN, JTE, JDE, JDM, JR, KWAX, IND, KW の8個(いずれも整数型)であるが、KMAX と IND をのぞく6個はM型と同一である。

NN : 仮数部の配列一要素にはいる10進の桁数を示す n 。 $1 \leq NN \leq 4$ 。

JTE : 指数部の上限。 $| \text{指数部} | < JTE$ 。 $3 \leq 3 \times NN < JDE < JTE < 2^{30}$ 。

ただし、JTE に負の値を与えるとデフォルト値 10^8 とみなされる。

JDE : 倍精度実数型の指数部の上限。 $10^{-JDE} < |x| < 10^{JDE}$ となる x が倍精度実数型で表現できること。 $3 \leq 3 \times NN < JDE < JTE < 2^{30}$ 。

ただし、JDE に負の値を与えるとデフォルト値75とみなされる。

JDM : 倍精度実数型の仮数部の有効桁数。 $2 \leq 2 \times NN < JDM \leq 15$ 。

ただし、JDM に負の値を与えるとデフォルト値15とみなされる。

JR : 丸め処理を定めるパラメータ。 $0 \leq JR \leq 10$ 。

ただし、JR に負の値を与えるとデフォルト値5とみなされる。

JR = 0 切り上げ → 小数点以下1位で切り上げて10となったとすれば、真の

1 0 捨1入 切り上げは $9 < x \leq 10$ であるが、JR = 0 のときは $9 \leq x < 10$ となり、多小異なる。

・ ・

・ ・

5 4 捨5入

・ ・

・ ・

9 8 捨9入

10 切り捨て

KMAX : 仮数部の長さの最大値を与える。 $5 \leq KMAX \leq 201$ 。

IND : 誤差の伝播や桁落ちによって、仮数部の長さが4より小さくなったときの処理を指示する。このとき、コモン領域中のエラーインディケータ IERROR は、仮数部の長さが4より小さくなる毎に1つつ引かれる(初期値はゼロ)。すなわち、IERROR の絶対値が発生回数を示すことになる。

IND = 0 仮数部の長さを4とし、誤差項を $10^{2^n} - 1$ として演算を続行する。

ただし、演算結果がゼロのときは誤差項もゼロとする。

1 警告のメッセージを印字する。その後の処理は IND = 0 と同一。

2 警告のメッセージとそのときの引数の値を印字する。その後の処理は IND = 0 と同一。

3 エラーメッセージとそのときの引数の値を印字し、演算を停止する。

NW : WRITE 文の装置番号, エラーの印字のときに必要である。

これらのプリセットパラメータから計算される諸定数などの情報の受渡しにはコモン領域を用いている。したがって, 利用者は主プログラムでM, S型の場合と全く同一の次の宣言をする必要がある。

DOUBLE PRECISION DIR

COMMON/BMSEI1/DIR, K, KK, KKK, KMAX, KMAX1, KMAX2, KMAX3, KMAXW,
KMAXW2, IR1, KETA, IND, IERROR, NW

ただし, コモン領域中の DIR 以外はすべて整数型である。また, K, KK, KKK, KETA は E 型では使用されず, M, S, I 型の多重精度演算と混用するときに意味をもつ。

(4) 使用法

引 数	型 と 種 類	属 性	内 容
MACC1 MACC2	整 数 型 1 次 元 配 列	入 力	E型の数を表わす配列。大きさ KMAX3。 ただし $KMAX3 \equiv KMAX+3$ 。
MACC3	整 数 型 1 次 元 配 列	出 力	E型の数を表わす配列。演算結果が入る。大きさKMAX3。 ただし $KMAX3 \equiv KMAX+3$ 。
INT	整 数 型	入 力	$ INT < 10^{NN}$ 。
NN,JTE JDE, JDM JR	整 数 型	入 力	プリセットパラメータ。(3)参照。

1. E型の多重精度演算ルーチンを使用するに際し必須のもの。

CALL STARTE(NN, JTE, JDE, JDM, JR)

(3)に述べたプリセットパラメータから計算される諸定数の計算と条件の判定, 並びにコモン領域中の IERROR (エラーインディケータ)のゼロクリアを行う。プリセットパラメータを定めた後, 他のサブルーチンを呼ぶ前に必ずこのルーチンを呼ぶこと。

2. E型の多重精度の二数の四則演算。

CALL ADDE(MACC1, MACC2, MACC3)

MACC3 ← MACC1+MACC2

このルーチンは内部でサブルーチン SUBE を呼んでいる。

CALL SUBE(MACC1, MACC2, MACC3)

MACC3 ← MACC1-MACC2

このルーチンは内部でサブルーチン ADDE を呼んでいる。

CALL MULT1E(MACC1, MACC2, MACC3)

MACC3 ← MACC1×MACC2

CALL DIV1E(MACC1, MACC2, MACC3)

MACC3 ← MACC1/MACC2

3. E型の多重精度の数と整数の乗除算

CALL MULT2E(MACC1, INT, MACC3)

MACC3 ← MACC1×INT

CALL DIVI2E (MACC1, INT, MACC3)

MACC3 ← MACC1/INT

4. スレーブサブルーチン

CALL ERRORE (NS1, NS2, MACC1, MACC2, INT, IOP)

エラー、警告が発生したときの印字ルーチンであり、それらの発生したサブルーチン名、エラーインディケータとそのときの引数の値を印字する。利用者は直接にこのルーチンを呼ぶことはない。2., 3.のルーチンから呼ばれる。引数の説明は省略。

(5) エラーインディケータ

コモン領域中のエラーインディケータ IERROR の値とその意味を示す。

0 : 正常処理

負 : 仮数部の長さが4より小さくなった回数にマイナスの符号をつけたものが入る。ただし、サブルーチンの引用の前後で値が変化しなければ、そのサブルーチンでは正常に処理されたことを示す。

10 : オーバーフロー。

11 : アンダーフロー。

20 : 不定 (DIVI1E)。

21 : 不能 (DIVI1E)。

30 : 不定 (DIVI2E)。

31 : 不能 (DIVI2E)。

40 : $|INT| > 10^{NN}$ 。

90 : プリセットパラメータエラー、ただし、チェック項目は次の通り。

$3 \leq 3 \times NN < JDE < JTE, 2 \times NN < JDM, 0 \leq JR \leq 10, 5 \leq KMAX \leq 201,$
 $0 \leq IND \leq 3。$

(6) 使用例

```
//ETYPE JOB
// EXEC FORTHCG, PARM.FORT='S', PRVLIB='CENT.MPRECIS'
//FORT.SYSIN DD *
C *** SPECIFICATION STATEMENT ***
  DIMENSION MACC1(14), MACC2(14), MACC3(14)
  DOUBLE PRECISION DIR
  COMMON /BMSEI1/DIR, K, KK, KKK, KMAX, KMAX1, KMAX2, KMAX3, KMAXW, KMAXW2,
  1 IR1, KETA, IND, IERROR, NW
C *** PARAMETER SET ***
  NW=6
  NR=5
  NN=4
  JTE=10**8
  JDE=75
  JDM=15
  JR=5
  KMAX=11
  IND=2
  CALL STARTE(NN, JTE, JDE, JDM, JR)
C *** TEST OF ADDE ***
  IERROR=0
  READ(NR, 9000) (MACC1(I), I=1, KMAX3)
  READ(NR, 9000) (MACC2(I), I=1, KMAX3)
```

```

K1=IABS(MACC1(1))+3
K2=IABS(MACC2(1))+3
WRITE(NW,2000) (MACC1(I),I=1,K1)
WRITE(NW,2100) (MACC2(I),I=1,K2)
CALL ADDE(MACC1,MACC2,MACC3)
WRITE(NW,2000) (MACC1(I),I=1,K1)
WRITE(NW,2100) (MACC2(I),I=1,K2)
K3=IABS(MACC3(1))+3
WRITE(NW,2200) (MACC3(I),I=1,K3)
STOP
2000 FORMAT(1H0,5HMACC1,I6,2I12,11I6)
2100 FORMAT(1H ,5HMACC2,I6,2I12,11I6)
2200 FORMAT(1H ,5HMACC3,I6,2I12,11I6)
9000 FORMAT(I3,I10,I12,11I4)
END
//GO.SYSIN DD *
-5      0      0 99999999999999999999
-5 10000000 2 9999 9999 9999 9999 9999
<出力例>
MACC1  -5      0      0 999 9999 9999 9999 9999
MACC2  -5 10000000 2 9999 9999 9999 9999 9999

MACC1  -5      0      0 999 9999 9999 9999 9999
MACC2  -5 10000000 2 9999 9999 9999 9999 9999
MACC3  -6 10000001 3   1   0   0 999 9999 9999

```

(7) プリセットパラメータの変更

プリセットパラメータの値を変更する場合は原則としてサブルーチン STARTE を呼ぶ必要があるが、JR, KMAX, IND, NW の変更の場合には、各パラメータが(3)に述べた条件を満足するかぎり、STARTE を呼ぶ必要はない。NW の変更手順は M, S 型の多重精度演算の場合と全く同一である。

JR : コモン領域中の $DIR(≡\max(JR, 10-JR)/10)$ と $IR1(≡JR \times 10^{NN-1})$ を変更する。
ただし、JR のデフォルト機能は使用できない。

KMAX: E型は誤差の伝播による仮数部の長さの変化を自動的に評価する機能をもっているから、KMAX を変更するよりも仮数部の長さを有効に用いる方が好ましい。どうしても KMAX を変更したいときには、コモン領域中の次のものを変更すればよい。ただし、KMAX は仮数部の長さの最大値であるから、仮数部の長さ \leq KMAX でなければならない。

```

KMAX
KMAX1 (KMAX1 ≡ KMAX+1)
KMAX2 (KMAX2 ≡ KMAX+2)
KMAX3 (KMAX3 ≡ KMAX+3)
KMAXW (KMAXW ≡ 2×KMAX)
KMAXW2 (KMAXW2 ≡ 2×KMAX+2)

```

IND : コモン領域中の IND を変更する。

NW : コモン領域中の NW を変更する。

(8) 備考

1. コモン領域中の IERROR の値が利用者が再定義したときには、その値が正であればゼロにリセットされ、非正であればその値が初期値となり、その後、仮数部の長さが4より小さくなるごとに1ずつ引かれる。すなわち、仮数部の長さが4より小さくなった回数の積算開始点を任意に設定で

きる。

2. コモン領域の名前は(3)で述べた BMSEI1の他に BMSEI2と BWRKE を用いている。したがって、BMSEI2と BWRKE は予約語であり、利用者は使用できない。
3. M, S, I 型の多重精度演算と混用する場合は、それぞれに必要なプリセットパラメータを定めた後、各型のスタートルーチンを呼ばばよい。
4. 引数 MACC1, MACC2が(2)に述べた表現を満足するか否かのチェックは行っていない。
5. 演算ルーチンを呼ぶときの配列名は各々異なる名でなくてもよい。

例 CALL SUBE(MACC1, MACC2, MACC1) MACC1 ← MACC1-MACC2
CALL DIVIE(MACC1, MACC2, MACC2) MACC2 ← MACC1/MACC2

参 考 文 献

- 1) 大中幸三郎, 安井裕; “誤差評価の可能な多重精度演算”, 情報処理, Vol.15, pp.109~117 (1974)。
- 2) K. Ohnaka, H. Yasui & T. Kubo; “An Automatic Error Estimation Method in Numerical Calculation,” Technol. Repts. Osaka Univ., Vol.25, pp.257-265 (1975).

I-TYPE

Multiple Precision Arithmetic with Error Estimation

(I-TYPE; Interval Arithmetic)

誤差評価の可能な多重精度演算 (I型)

作成	大中 幸三郎, 安井 裕 1973年, 改訂 大中 幸三郎 1979年
形式	サブルーチン 言語; FORTRAN サイズ; 1,207行
利用者の義務	プログラム名と作者名を明記する。

(1) 概要

数値計算によって得られた解の精度, 言い換えれば誤差の評価は, 数値計算上の重要な事柄である。解析的な評価も重要ではあるが, どのような計算過程についても誤差の振舞いを知らうとするときには, 計算過程の個々について何らかの形で直接的に誤差の影響をつかんだ演算を行わねばならない。したがって, 我々は誤差評価を可能とする機能として, 丸めの制御, 誤差項の評価, interval arithmetic 等を行う機能をもつ一連の多重精度演算パッケージ (M, S, E, I型) を作成した。

I型は interval arithmetic を行うパッケージであり, 誤差評価機能としては完全であると言ってもよい。しかしながら, 誤差を過大に評価する点と実行時間がM, S型の3~5倍程度必要とする点に注意を要する。

(2) 数の表現

多重精度の interval arithmetic を行うのがI型である。したがって, I型の多重精度の数は interval number となるから, S型の二数を用いて表わし (S型参照), 整数型二次元配列となる。その例として π を10進25桁について示すと下図のようになる。

配列名

IX	(1,)	(2,)	
(, 1)	1	1	} 仮数部
(, 2)	0	0	
(, 3)	3	3	
(, 4)	14159265	14159265	
(, 5)	35897932	35897932	
(, 6)	38462644	38462644	

$$IX \text{ の上限} \equiv IX(1,) \equiv 3.1415 \dots 2644 \times (10^4)^0$$

$$IX \text{ の下限} \equiv IX(2,) \equiv 3.1415 \dots 2642 \times (10^4)^0$$

IX(1,) に上限を, IX(2,) に下限を入れることに注意を要する。

(3) プリセットパラメータと宣言文

利用者が与えるプリセットパラメータは NN, JTE, JDE, JDM, K, NW の6個 (いずれも整数型) で

あるが、すべてS型と同一である。

NN : 仮数部の配列一要素にはいる10進の桁数を示す n 。 $1 \leq NN \leq 4$ 。

JTE : 指数部の上限。 $| \text{指数部} | < JTE$ 。 $3 \leq 3 \times NN < JDE < JTE < 2^{30}$ 。

ただし、JTE に負の値を与えるとデフォルト値 10^8 とみなされる。

JDE : 倍精度実数型の指数部の上限。 $10^{-JDE} < |x| < 10^{JDE}$ となるが x が倍精度実数型で表現できること。 $3 \leq 3 \times NN < JDE < JTE < 2^{30}$ 。

ただし、JDE に負の値を与えるとデフォルト値75とみなされる。

JDM : 倍精度実数型の仮数部の有効桁数。 $2 \leq 2 \times NN < JDM \leq 15$ 。

ただし、JDM に負の値を与えるとデフォルト値15とみなされる。

K : S型の多重精度の数を表わす整数型一次元配列の長さ。I型では二次元配列の大きさ $2 \times K$ の部分が多重精度の数となる。 $5 \leq K \leq 103$ 。

NW : WRITE 文の装置番号。エラーの印字のときに必要である。

これらのプリセットパラメータから計算される諸定数などの情報の受渡しにはコモン領域を用いている。したがって、利用者は主プログラムでM, S, E型の場合と全く同一の次の宣言をする必要がある。

DOUBLE PRECISION DIR

COMMON/BMSEI1/DIR, K, KK, KKK, KMAX, KMAX1, KMAX2, KMAX3, KMAXW,
KMAXW2, IR1, KETA, IND, IERROR, NW

ただし、コモン領域中の DIR 以外はすべて整数型である。また、I型で使用されるものはK, KK, KKK, IERROR, NWのみであり、他はM, S, E型の多重精度演算と混用するときに意味をもつ。

(4) 使用法

引 数	型 と 種 類	属 性	内 容									
LXI LYI	整 数 型 2 次 元 配 列	入 力	I型の数を表わす配列。ただし $LR \geq K$ 。 大きさ $2 \times LR$ 。									
LZI	整 数 型 2 次 元 配 列	出 力	I型の数を表わす配列。演算結果が入る。ただし $LR \geq K$ 。 大きさ $2 \times LR$ 。									
MACC1 MACC2	整 数 型 1 次 元 配 列	入 力	M型の数を表わす配列。大きさ KK。 ただし $KK \equiv 2 \times K - 3$ 。									
MACC3	整 数 型 1 次 元 配 列	出 力	M型の数を表わす配列。演算結果が入る。大きさ KK。 ただし $KK \equiv 2 \times K - 3$ 。									
MACC4 MACC5	整 数 型 1 次 元 配 列	作 業 領 域	大きさ KK。ただし $KK \equiv 2 \times K - 3$ 。									
MACCW	整 数 型 1 次 元 配 列	作 業 領 域	大きさ KKK。ただし $KKK \equiv 4 \times K - 8$ 。									
INT	整 数 型	入 力	$ INT < 10^{NN}$ 。									
IR 2	整 数 型	入 力	丸め処理を指定する。値は0又は1を入力する。 <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>IR 2 \ 演算結果</td> <td>負</td> <td>正, 0</td> </tr> <tr> <td>0</td> <td>切り上げ</td> <td>切り捨て</td> </tr> <tr> <td>1</td> <td>切り捨て</td> <td>切り上げ</td> </tr> </table>	IR 2 \ 演算結果	負	正, 0	0	切り上げ	切り捨て	1	切り捨て	切り上げ
IR 2 \ 演算結果	負	正, 0										
0	切り上げ	切り捨て										
1	切り捨て	切り上げ										

引 数	型 と 種 類	属 性	内 容
			ただし、この場合の切り上げはM, S, E型の JR = 0 の場合とは異なり、真の切り上げである。
I	整 数 型	入 力	I型↔M型の変換に用いられる。値は1又は2を入力する。
NN, JTE JDE, JDM	整 数 型	入 力	プリセットパラメータ。(3)参照。

1. I型の多重精度演算ルーチンを使用するに際し必須のもの

CALL STARTI(NN, JTE, JDE, JDM)

(3)に述べたプリセットパラメータから計算される諸定数の計算と条件の判定、並びにコモン領域中の IERROR (エラーインディケータ) のゼロクリアを行う。プリセットパラメータを定めた後、他のサブルーチンを呼ぶ前に必ずこのルーチンを呼ぶこと。

2. I型の多重精度の二数の四則演算

CALL ADDI(LXI, LYI, LZI)

LZI ← LXI+LYI

CALL SUBI(LXI, LYI, LZI)

LZI ← LXI-LYI

CALL MULT1I(LXI, LYI, LZI)

LZI ← LXI×LYI

CALL DIV1I(LXI, LYI, LZI)

LZI ← LXI/LYI

3. I型の多重精度の数と整数の乗除算

CALL MULT2I(LXI, INT, LZI)

LZI ← LXI×INT

CALL DIVI2I(LXI, INT, LZI)

LZI ← LXI/INT

4. スレーブサブルーチン (その1 ; I型 ↔ M型の変換)

2., 3.のサブルーチンの内部で呼ばれているものであり、5.に示すサブルーチンを用いるために必要である。利用者は必ずしも直接にこれらを利用する必要はない。

CALL EXPAN(LXI, MACC3, D)

MACC3 ← LXI (I,)

CALL COMP(MACC1, LZI, D)

LZI(I,) ← MACC1

5. スレーブサブルーチン (その2 ; M型の演算ルーチン)

2., 3.のサブルーチンの内部で呼ばれているものであり、M型の多重精度の数の表現に対してI型の演算を行うサブルーチンである。利用者は必ずしも直接にこれらを使用する必要はないが、有効に使用すれば2., 3.のものを用いるよりも実行時間の短縮をはかることができる。ただし、エラーが発生した場合は、コモン領域中の IERROR (エラーインディケータ) がセットされるだけで、エラーの表示は行わない。この時の演算結果は不定である。

CALL ADDIP (MACC1, MACC2, MACC3, MACC4, MACC5, IR2)

MACC3 ← MACC1 + MACC2

ただし、MACC1とMACC2は異符号であってはならない。

CALL SUBIP (MACC1, MACC2, MACC3, MACC4, MACC5, IR2)

MACC3 ← MACC1 - MACC2

ただし、MACC1とMACC2は異符号であってはならない。

CALL MU1IP (MACC1, MACC2, MACC3, MACCW, IR2)

MACC3 ← MACC1 × MACC2

CALL DI1IP (MACC1, MACC2, MACC3, MACC4, MACCW, IR2)

MACC3 ← MACC1 / MACC2

CALL MU2IP (MACC1, INT, MACC3, MACC4, IR2)

MACC3 ← MACC1 × INT

CALL DI2IP (MACC1, INT, MACC3, MACC4, IR2)

MACC3 ← MACC1 / INT

6. スレーブサブルーチン (その3 ; エラー処理ルーチン)

CALL ERRORI (NS1, NS2, LXI, LYI, INT, IOP)

エラーが発生したときの印字ルーチンであり、エラーの発生したサブルーチン名、エラーインディケータとそのときの引数の値を印字し、実行を停止する。利用者は直接にこのルーチンを呼ぶことはない。2., 3.のサブルーチンから呼ばれる。引数の説明は省略。

(5) エラーインディケータ

コモン領域中のエラーインディケータ IERROR の値とその意味を示す。

0 : 正常処理。

負 : 正常処理, E型の多重精度演算と混用するか, 利用者が IERROR の値を負に再定義しないかぎり, 負とならない。

10 : オーバーフロー。

20 : 不定 (DIVI1)。

21 : 不能 (DIVI1)。

30 : 不定 (DIVI2)。

31 : 不能 (DIVI2)。

40 : $|INT| \geq 10^{NN}$ 。

90 : プリセットパラメータエラー。ただし, チェック項目は次の通り。

$3 \leq 3 \times NN < JDE < JTE, 2 \times NN < JDM, 5 \leq KM.$

(6) 使用例

```
//ITYPE JOB
// EXEC FORTHCG, PARM.FORT='S', PRVLIB='CENT.MPRECIS'
//FORT.SYSIN DD *
C *** SPECIFICATION STATEMENT ***
  DIMENSION LXI(2,8), LYI(2,8), LZI(2,8), MACC1(13), MACC2(13),
  1 MACC3(13), MACC4(13), MACC5(13), MACCW(24)
  DOUBLE PRECISION DIR, DN, DIT, DIT2, DIT3
```


MMFUNC

Elementary Functions of Multiple Precision (M-TYPE)

多重精度の初等関数 (M型)

作成	大中 幸三郎, 安井 裕 1973年, 改訂 大中 幸三郎 1979年
形式	サブルーチン 言語; FORTRAN サイズ; 1,498行
利用者の義務	プログラム名と作者名を明記する。

(1) 概要

M型の多重精度演算を用い、平方根、正弦、余弦、逆正接、常用対数、10を底とする指数の値を求めるサブルーチンパッケージである。内部でM型のサブルーチンを用いているから、M型に必要なプリセットパラメータ、宣言文を利用者が定め、M型のサブルーチン START を呼んだ後であれば、初等関数の値を計算できる。また、M型とS型の表現の変換を利用者が行うことにより、S型の初等関数の計算にも使用できる。E、I型との混用は、誤差項と区間中の評価ができない点はあるが、混用すること自体は問題はないから、利用者の判断によって有効に使用することができる。

(2) 使用法

引数	型と種類	属性	内容
MPAI4 MLOG2 MLOG3 MLOGE	整数型 1次元配列	入力 又は 出力	サブルーチン PAILOG では出力。M型の数を表わす配列、 $\pi/4$, $\log_{10}2$, $\log_{10}3$, $\log_{10}e$ の値が入っている。他のサブルーチンでは入力。M型の数を表わす配列。初等関数の値を計算するのに必要な定数 $\pi/4$, $\log_{10}2$, $\log_{10}3$, $\log_{10}e$ の値を与える。大きさ KM。
MA	整数型 1次元配列 KM以上	入出力	M型の数を表わす配列。大きさ KM。関数の値を求める点を入力する。出力としては入力した点の関数値が入る。 $MA \leftarrow f(MA)$ 。
MB, MC MD, ME MF	整数型 1次元配列	作業領域	大きさ KM。
MW	整数型 1次元配列 KKM以上	作業領域	大きさ KKM。ただし $KKM \equiv 2 \times KM - 2$ 。
IOP	整数型	入力	サブルーチン PAILOG と ADSU1 で用いられるものである。意味については各ルーチンの説明の項で述べる。
KM	整数型	入力	M型のプリセットパラメータ。(M型参照)
KKM	整数型	入力	M型のサブルーチン START で計算される定数。 $KKM \equiv 2 \times KM - 2$ 。(M型参照)

1. $\pi/4$, $\log_{10}2$, $\log_{10}3$, $\log_{10}e$ の計算。

CALL PAILOG (MPAI4, MLOG2, MLOG3, MLOGE, MB, MW, IOP, KM, KKM)

2. に示すサブルーチンに必要な定数 $\pi/4$, $\log_{10}2$, $\log_{10}3$, $\log_{10}e$ を求めるサブルーチンである。こ

これらの値が所要の精度まで既知の場合は、このルーチンを用いる必要はない。IOP の役割は、

$$\text{IOP} \begin{cases} > 0 & \pi/4 \text{ のみ計算} \\ = 0 & \pi/4, \log_{10} 2, \log_{10} 3, \log_{10} e \text{ を計算} \\ < 0 & \log_{10} 2, \log_{10} 3, \log_{10} e \text{ を計算} \end{cases}$$

である。

なお、コモン領域中の KETA の値はゼロとなる。

2. 初等関数の計算

CALL MMSQRT(MA, MB, MC, MW, KM, KKM)

$MA \leftarrow MA^{1/2}$ ただし、 $MA \geq 0$ 。

コモン領域中の KETA の値はゼロとなる。

CALL MMSIN(MA, MPAI4, MB, MC, MW, KM, KKM)

$MA \leftarrow \sin(MA)$

コモン領域中の KETA の値については(4)参照。

CALL MMCOS(MA, MPAI4, MB, MC, MW, KM, KKM)

$MA \leftarrow \cos(MA)$

コモン領域中の KETA の値については(4)参照。

CALL MMATAN(MA, MPAI4, MB, MC, MD, ME, MF, MW, KM, KKM)

$MA \leftarrow \tan^{-1}(MA)$

このルーチンは内部でサブルーチン MMSQRT を呼んでいる。また、コモン領域中の KETA の値はゼロとなる。

CALL MMLG10(MA, MLOG2, MLOG3, MLOGE, MB, MC, MW, KM, KKM)

$MA \leftarrow \log_{10}(MA)$ ただし、 $MA > 0$ 。

コモン領域中の KETA の値については(4)参照。

CALL MMEP10(MA, MLOGE, MB, MW, KM, KKM)

$MA \leftarrow 10^{MA}$

コモン領域中の KETA の値については(4)参照。

3. スレーブサブルーチン

以下に示すサブルーチンは 2. のルーチンの内部から呼ばれているものであるが、使用条件を満足する場合には、直接、利用者が使用してもよい。

CALL MMSINP(MA, MB, MC, MW, KM, KKM)

$MA \leftarrow \sin(MA)$ ただし、 $-\pi/4 \leq MA \leq \pi/4$ 。

MA が使用条件を満足するか否かのチェックは行っていない。エラーが発生したときにはエラーメッセージを印字せず、MA(1)に 100 を入れて戻る。なお、KETA の値は不定である。

CALL MMCOSP(MA, MB, MC, MW, KM, KKM)

$MA \leftarrow \cos(MA)$ ただし、 $-\pi/4 \leq MA \leq \pi/4$ 。

MA が使用条件を満足するか否かのチェックは行っていない。エラーが発生したときにはエラーメッセージを印字せず、MA(1)に 100 を入れて戻る。なお、KETA の値は不定である。

CALL ADSU1(MA, IOP, KM)

$$MA \leftarrow \begin{cases} MA+1 & IOP > 0 \\ MA & IOP = 0 \\ MA-1 & IOP < 0 \end{cases}$$

IOP の値がゼロのときには何もせずに戻る。なお、KETA については M 型の多重精度演算のサブルーチン ADD, SUB と同一である。MA ± 1 の演算の場合は、M 型の ADD, SUB を用いるよりも、このルーチンを使用した方が実行時間と記憶容量の点で有利である。

(3) エラーメッセージ

エラーが発生した場合には、その発生したサブルーチン名とエラーの種類を文字で印字し、実行を停止する。(例外 MMSINP, MMCOSP)。コモン領域中のエラーインディケータ IERROR は使用しない。すなわち、初等関数ルーチンの内部で使用している M 型の演算ルーチンで変更されないかぎり、IERROR の値は保存される。

(4) 計算法

1. $\pi/4$, $\log_{10} 2$, $\log_{10} 3$, $\log_{10} e$

NN = 1 のとき

倍精度の基本外部関数 DATAN, DLOG10, DLOG を用いる。

NN ≥ 2 のとき

$$\pi/4 = \begin{cases} 4 \tan^{-1} \frac{1}{5} - \tan^{-1} \frac{1}{70} + \tan^{-1} \frac{1}{99}, & NN = 2, \\ 12 \tan^{-1} \frac{1}{18} + 8 \tan^{-1} \frac{1}{57} - 5 \tan^{-1} \frac{1}{239}, & NN \geq 3. \end{cases}$$

ただし、 \tan^{-1} の計算式は Taylor 展開による。

$$\log_{10} 2 = \log_e 2 / \log_e 10,$$

$$\log_{10} 3 = \log_e 3 / \log_e 10,$$

$$\log_{10} e = 1 / \log_e 10.$$

$$\text{ただし、} \log_e 2, \log_e 3, \log_e 10 \text{ の計算式は } \log_e \frac{x}{a} = 2 \sum_{n=0}^{\infty} \frac{1}{2n+1} \left(\frac{x-a}{x+a} \right)^{2n+1}$$

を用い、 $\log_e 2$ では $a = 1$ 、 $\log_e 3$ では $a = 2$ 、 $\log_e 10$ では $a = 8$ とする。

2. 平方根

Newton 法を用いる。ただし、初期値は倍精度の基本外部関数 DSQRT によって求める。

$$x_{n+1} = \frac{1}{2} \left(x_n + \frac{a}{x_n} \right), \quad \lim_{n \rightarrow \infty} x_n = \sqrt{a}$$

3. 正弦と余弦

対称性と周期性から、 $0 \leq x \leq \pi/4$ での正弦と余弦が求められれば、全区間での値が求まる。

$0 \leq x \leq \pi/4$ の区間の計算式は Taylor 展開を用いる。

$$\sin x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!},$$

$$\cos x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!}$$

コモン領域の KETA は与えられた引数から、 $0 \leq x \leq \pi/4$ に変換するときの modulo 演算において桁落ちした桁数を示す。

4. 逆正接 (主値)

次の関係式により、 $0 \leq x \leq \sqrt{2}-1$ での逆正接が求められれば、全区間での値が求まる。

$$\tan^{-1} x = -\tan^{-1}(-x),$$

$$\tan^{-1} x = \frac{\pi}{2} - \tan^{-1} \frac{1}{x},$$

$$\tan^{-1} x = \frac{\pi}{4} + \tan^{-1} \frac{1-x}{1+x}$$

$0 \leq x \leq \sqrt{2}-1$ に対し倍角公式

$$\tan^{-1} x = 2 \tan^{-1} \frac{\sqrt{1+x^2}-1}{x},$$

を高々 2 回用いると区間を $0 \leq x < 0.0985$ に縮小することができる。この区間に対して Taylor 展開式を用いる。

$$\tan^{-1} x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{2n+1}$$

5. 常用対数

$0.1 \leq x < 1$ での常用対数が求められれば、その値に整数を加減算することによって、 $0 < x$ での値が求まる。 $0.1 \leq x < 1$ の区間に対して定数 $\log_{10} 2$, $\log_{10} 3$ を使い、区間をさらに $0.4 \leq x < 0.5$ に縮小する。この区間での計算式は次の級数展開を用いる。

$$\log_{10} x = \log_{10} a + 2 \log_{10} e \sum_{n=0}^{\infty} \frac{1}{2n+1} \left(\frac{x-a}{x+a} \right)^{2n+1}, \quad \text{ただし, } a = 0.45$$

与えられた引数が 1 の近傍であれば、桁落ちが発生する。したがって、コモン領域の KETA は与えられた引数から 1 を減算したときに桁落ちした桁数を示す。

6. 10 を底とする指数

$-1 < x < 1$ での値が求められれば、その値に整数を乗除算することによって、全区間での値が求まる。 $-1 < x < 1$ に対して関係式

$$10^x = (e^{x \log_e 10/k})^k, \quad \text{ただし, } k = 1, 2, 4, 8, 16, 32, 64,$$

を用い、区間を $-0.036 < x \log_e 10/k < 0.036$ に縮小する。この区間での計算式は Taylor 展開を用いる。

$$e^y = \sum_{n=0}^{\infty} \frac{y^n}{n!}$$

コモン領域の KETA は与えられた引数の整数部の桁数を示す。正確には、引数の配列名を MA とすれば、次式となる。

$$KETA \leftarrow \begin{cases} 0 & MA(2) < 0 \\ MA(2)+1 & 0 \leq MA(2) \leq KM-4 \\ KM-2 & KM-3 \leq MA(2) \end{cases}$$

(5) 使用例

```

//MMFUNC JOB
// EXEC FORTHCG,PARM.FORT='S',PRVLIB='CENT.MPRECIS'
//FORT.SYSIN DD *
C *** SPECIFICATION STATEMENT ***
  DIMENSION MPAI4(13),MLOG2(13),MLOG3(13),MLOGE(13),
  1          MA(13),MB(13),MC(13),MD(13),ME(13),MF(13),MW(24)
  DOUBLE PRECISION DIR,DN,DIT,DIT2,DIT3
  COMMON /BMSEI1/DIR,K,KK,KKK,KMAX,KMAX1,KMAX2,KMAX3,KMAXW,KMAXW2,
  1          IR1,KETA,IND,IERROR,NW
C *** PARAMETER SET ***
  NW=6
  NR=5
  NN=4
  JTE=10**8
  JDE=75
  JDM=15
  JR=5
  KM=13
  CALL START(NN,JTE,JDE,JDM,JR,KM,KKM)
C *** TEST OF PAILOG ***
  IOP=0
  WRITE(NW,1100)
  CALL PAILOG(MPAI4,MLOG2,MLOG3,MLOGE,MB,MW,IOP,KM,KKM)
  WRITE(NW,2110) NN,KM
  WRITE(NW,2120) (MPAI4(I),I=1,KM)
  WRITE(NW,2130) (MLOG2(I),I=1,KM)
  WRITE(NW,2140) (MLOG3(I),I=1,KM)
  WRITE(NW,2150) (MLOGE(I),I=1,KM)
C *** TEST OF MMSQRT ***
  WRITE(NW,1200)
  READ(NR,9000) (MA(I),I=1,KM)
  WRITE(NW,8100) (MA(I),I=1,KM)
  CALL CLOCKM(ITIME)
  CALL MMSQRT(MA,MB,MC,MW,KM,KKM)
  CALL CLOCKM(JTIME)
  JTIME=JTIME-ITIME
  WRITE(NW,2210) (MA(I),I=1,KM),KETA,JTIME
  DO 21 I=1,KM
  MB(I)=MA(I)
  21 CONTINUE
  CALL MULT1(MA,MB,MW,KM,KKM)
  WRITE(NW,2220) (MA(I),I=1,KM)
  WRITE(NW,8200)
  STOP
1100 FORMAT(30H ***** TEST OF PAILOG ***** //)
1200 FORMAT(30H ***** TEST OF MMSQRT ***** //)
2110 FORMAT(4H N =,I2,10X,4HKM =,I3)
2120 FORMAT(16H PAI/4 ,I3,I12,1116)
2130 FORMAT(16H LOG10(2) ,I3,I12,1116)
2140 FORMAT(16H LOG10(3) ,I3,I12,1116)
2150 FORMAT(16H LOG10(E) ,I3,I12,1116)
2210 FORMAT(16H SQRT ,I3,I12,1116,
  1          6X,4HKETA,I3,6X,10HTIME(MSEC),I6 )
2220 FORMAT(16H SQRT**2 ,I3,I12,1116)
8100 FORMAT(16H ARGUMENT ,I3,I12,1116)
8200 FORMAT(1H ,I30(1H-))
9000 FORMAT(I2,I13,1114)
9100 FORMAT(I2,I13,1115,6X,I4)
  END
//GO.SYSIN DD *
  1          1 2 0 0 0 0 0 0 0 0 0
//

```

<出力例>

***** TEST OF PAILOG *****

```

N = 4          KM = 13
PAI/4         1          -1  7853  9816  3397  4483   961  5660  8458  1987
LOG10(2)      1          -1  3010  2999  5663  9811  9521  3738  8947  2449
LOG10(3)      1          -1  4771  2125  4719  6624  3729  5027  9032  5511
LOG10(E)      1          -1  4342  9448  1903  2518  2765  1128  9189  1660
***** TEST OF MMSQRT *****

```

```

ARGUMENT      1          1          2          0          0          0          0          0          0          0          0
SQRT           1          0          141  4213  5623  7309  5048  8016  8872  4209
SQRT**2        1          1          2          0          0          0          0          0          0          0

```

(6) KM の値に対する制限

(4) に述べたように平方根の場合を除いて、初等関数の計算は級数展開を用いている。級数展開の係数の計算は、実行時間の短縮のためにM型の多重精度演算ルーチン中の DIVI2 を使用している。したがって、DIVI2 の引数の制限条件(M型参照)から、計算する項数に限界がある。この限界はサブルーチン、引数の値によって異なるが、KM が $10^{NN}/NN$ のオーダーになると現われる可能性がある。NN の値が1 のときに最も制限がきびしいものは、JDM に小さな値を入れないうえ、MMSIN (MMSINP), MMCOS (MMCOSP) であり、NN の値が2 以上のときには PAILOG において $IOP \leq 0$ のときである。

上記の制限により、NN の値が1 のときには倍精度演算よりも有効桁数が短くなり、初等関数ルーチンは実用上、無意味となる。NN の値が2 以上のときは倍精度演算よりも有効桁数が長く、特に3 以上であれば実用上の制限はないに等しい。

この制限に違反した場合には、エラーメッセージを印字し、実行を停止する。利用者の対策としては、NN の値を可能なかぎり大きくして用いることである。特に NN が2 以上のときは、すでに述べたように PAILOG が最も制限がきびしい。他のものは $KM > 10^{NN}/NN$ で発生するのに対し、PAILOG では $KM \approx 10^{NN}/(2 \times NN)$ で発生する。したがって、PAILOG による定数計算だけを NN の値を大きくすることによって、救済できる場合もある。ただし、この場合にはM型の多重精度の数の表現(M型参照)の変換は利用者が行う必要がある。

(7) 備考

- (6) で述べた制限以外に KM には、 $KM \leq JTE$ の制限があるが、この制限は JTE の値を極端に小さくとらないかぎり、違反することはなく、実用上は無視してよい。なお、 $KM \leq JTE$ のチェックは行っていない。
- 初等関数ルーチンに限らず、M, S, E, I 型の多重精度演算ルーチンにおいても、NN の値を可能なかぎり大きく、KM の値を必要最小限にとる方が、実行時間と記憶容量の点で有利である。特に初等関数ルーチンでは、その効果は大きい。
- 初等関数ルーチンは丸め誤差と打ち切り誤差が影響するから、M, S, E, I 型の多重精度演算ルーチンよりも誤差は大きい。通常、その誤差は配列の最下位一要素以内であるが、もっと大きな誤差を含む場合もある。誤差の推定にはコモン領域中の KETA の値を用いることができる。
- 引数 MA がM型の多重精度の数の表現を満足するか否かのチェックは行っていない。
- 内部でM型の多重精度演算ルーチンを用いているから、M型の使用法で述べた事項はすべて適用

される。

参 考 文 献

- 1) 大中幸三郎, 安井裕; “誤差評価の可能な多重精度演算”, 情報処理, Vol.15, pp.109~117 (1974)。
- 2) D. Shanks & J.W. Wrench, Jr.: “Calculation of π to 100,000 Decimals, Math. Comp.,” Vol.16, pp.76-99 (1962)。
- 3) 森口繁一, 宇田川銑久, 一松信; 数学公式II, 岩波全書 229, 岩波, 東京, p.62, p.197 (1957)。

RANDOM / DRANDM

Generation of Uniform Random Number

一様乱数の生成

作 成	二宮 市三	1980 年
形 式	関数	言語 ; アセンブラ サイズ ; 25行

(1) 概 要

区間(0, 1)の擬似一様乱数を合同法により生成する。RANDOM は単精度用, DRANDM は倍精度用ルーチンであり, DRANDM は倍精度の宣言を必要とする。

(2) 使用法

$X = \text{RANDOM}(IX)$

引 数	型 と 種 類*	属 性	内 容
IX	整 数 型	入 力	初期値。非負の整数。一組の乱数列を生成する場合, 2回目以後は0でなければならない。

(3) 使用例

```
DIMENSION X(100)
X(1) = RANDOM(1)
DO 1 I = 2, 100
1 X(I) = RANDOM(0)
  ⋮
  ⋮
```

(4) 備 考

多量の乱数を発生させる場合は, 富士通 SSL II 中の RANU 2 の方が効率がよい。

BITLOGIC

Bitwise Logical Operations for 4-Byte Data

4 バイトデータ間のビットごとの論理演算

作成	二宮 市三	1980年8月
形式	関数	言語 ; アセンブラ サイズ ; 71行

(1) 概要

4 バイトデータ間のビットごとの論理演算を行う関数である。

IAND	論理積	INAND	論理積の否定
IOR	論理和	INOR	論理和の否定
IEOR	排他的論理和	IMPLY	含意
IEQV	対当	IDIF	論理差
INEQV	対当の否定 (IEOR に同じ)	ICOMPL	} 否定
		INOT	

(2) 使用法

4 バイト整数型関数。引数は4 バイト数ならば何でもよい。ICOMPL, INOT は1 変数関数, 他はすべて2 変数関数である。

K = IAND (X, Y)	K = INOR (X, Y)
K = IOR (X, Y)	K = IMPLY (X, Y)
K = IEOR (X, Y)	K = IDIF (X, Y)
K = IEQV (X, Y)	K = ICOMPL (X)
K = INEQV (X, Y)	K = INOT (X)
K = INAND (X, Y)	

念のためにこれらの関数の真理値表を示す。

X	Y	IAND	IOR	IEOR (INEQV)	IEQV	INAND	INOR	IMPLY	IDIF	ICOMPL(X) (INOT)
0	0	0	0	0	1	1	1	1	0	1
0	1	0	1	1	0	1	0	1	0	1
1	0	0	1	1	0	1	0	0	1	0
1	1	1	1	0	1	0	0	1	0	0

(3) 備考

論理演算子 .NOT., .AND., .OR. に対する論理要素は論理型変数 (定数, 配列の要素, 関係式) でなければならないが, この関数は4 バイトデータに対して, ビットごとの論理演算を行う。

ROUND / DROUND

Round-off of Real Numbers

実数の0捨1入

作成	二宮 市三	1981年4月
形式	関数	言語 ; アセンブラ サイズ ; 18, 18行

(1) 概要

ROUND(DROUND)は倍精度(4倍精度)の実数を単精度(倍精度)に0捨1入する。

(2) 使用法

ROUND(D), DROUND(Q)

D(Q)は任意の倍精度(4倍精度)の式である。DROUNDは倍精度の宣言を要する。