FUJITSU

FUJITSU Software

FUJITSU
SSL II 拡張機能使用手引書 II

J2UL-1905-01Z0(01)
2015年 9月

ま　え　が　き

本マニュアルは，科学用サブルーチンライブラリ SSL II(Scientific Subroutine Library II)の拡張機能 II の使用方法について記述しています.

SSL II の拡張機能の第 2 巻であり，大規模な科学技術計算をスーパーコンピュータで高速に処理する上で有効なアルゴリズム・機能を第 1 巻の増補版として提供しています.

本書の構成は次のとおりです.

第 I 部　概説
SSL II の拡張機能 II の概要および利用する上での注意事項を示します.

第 II 部　サブルーチンの使用方法
個々のサブルーチンの使用方法を述べます．各サブルーチンはアルファベット順に編集されています.

SSL II の拡張機能 II は，最新の技術を維持するために，改良並びに新規追加がなされることがあります．また，改良若しくは新規追加したサブルーチンが，既存サブルーチンの機能を包含し，性能的にまさる場合には，一定の猶予期間をおいて，既存サブルーチンを削除することがありますので，あらかじめご承知おきください.

SSL II の全般的な規約及び標準機能ならびに拡張機能（第 1 巻）については，次のマニュアルを参照してください.
“富士通 SSL II 使用手引書”
“FUJITSU SSL II 拡張機能使用手引書”
なお，SSL II の拡張機能 II はオーストラリア国立大学（The Australian National University：ANU）と富士通株式会社の共同開発によるものです.
ANU での開発は，Mike Osborne 教授と Richard Brent 教授により指導され，オーストラリア国立大学スーパーコンピュータセンター長の Bob Gingold 博士が幹事を勤められました．下記の ANU の研究者の方々が SSL II の拡張機能 II の設計と実現に従事されました．富士通株式会社はこれらの人々の御協力に対して感謝致します.

Professor Richard Peirce Brent

Dr Andrew James Cleary

Dr Murray Leslie Dow

Mr Christopher Robert Dun

Dr Lutz Grosz

Dr David Lawrence Harrar II

Dr Markus Hegland

Ms Judith Helen Jenkinson

Dr Margaret Helen Kahn

Mr Jeoffrey Keating

Dr Zbigniew Leyk

Mr Gavin John Mercer

Mr David John Miron

Mr Ole Møller Nielsen

Professor Michael Robert Osborne

Dr Peter Frederick Price

Dr Stephen Gwyn Roberts

Dr David Barry Singleton

Dr David Edward Stewart

本書は旧マニュアル(前版)を元に作成しました．記載内容が旧マニュアルと異なる箇所には，目次に＊印を付けてあります．

輸出管理規制について

本ドキュメントを輸出または第三者へ提供する場合は，お客様が居住する国および米国輸出管理関連法規等の規制をご確認のうえ，必要な手続きをおとりください．

出版年月および版数

版数	マニュアルコード
2015 年　9 月 第 11 版	J2UL-1905-01Z0(01)
2014 年 10 月 第 10 版	J2UL-1905-01Z0(00)
2013 年　6 月 第 9 版	―
2011 年　3 月 第 8 版	―
2008 年　2 月 第 7 版 改訂	―
2003 年 12 月 第 7 版	―
2002 年 12 月 第 6 版	―
2001 年　9 月 第 5 版	―
1999 年　9 月 第 4 版	―
1997 年 12 月 第 3 版	―
1997 年　6 月 第 2 版	―
1995 年 11 月 初版	―

変更履歴

変更内容	変更箇所	版数
Neumann型前処理を利用する場合の使用上の注意を追加	VCGD, DVCGD, VCGE, DVCGE	第 9 版
体裁の見直し	表紙, まえがき	第 10 版
作業域 W についての使用上の注意を追加	VCFM1, DVCFM1, VCFT3, DVCFT3	第 11 版
誤記訂正	VCPF1, DVCPF1	

お　願　い

・本書を無断で他に転載しないようお願いします.

・本書は予告なしに変更されることがあります.

目　　次

注）　＊については“まえがき”を参照.

SSL II 拡張機能 II 一覧表

線型計算

サブルーチン名	項　　目	ページ
VLSPX	正値対称行列の連立 1 次方程式 （ブロック化されたコレスキー分解法）	124
VSPLL	正値対称行列の LL^T 分解 （ブロック化されたコレスキー分解法）	190
VSPLX	LL^T 分解された正値対称行列の連立 1 次方程式	193
VLSBX	正値対称バンド行列の連立 1 次方程式 （変形コレスキー分解）	120
VBLDL	正値対称バンド行列の LDL^T（変形コレスキー分解）	44
VBLDX	LDL^T 分解された正値対称バンド行列の連立 1 次方程式	48
VLBX	実バンド行列の連立 1 次方程式（ガウスの消去法）	113
VBLU	実バンド行列の LU 分解（ガウスの消去法）	51
VBLUX	LU 分解された実バンド行列の連立 1 次方程式	56
VLDIV	LDL^T 分解された正値対称行列の逆行列	118
VLTQR	実 3 重対角行列の連立 1 次方程式（QR 分解）	127
VBCSD	非対称または不定値のスパース実行列の連立 1 次方程式 （BICGSTAB(*l*)法，対角形式格納法）	36
VBCSE	非対称または不定値のスパース実行列の連立 1 次方程式 （BICGSTAB(*l*)法，ELLPACK 形式格納法）	40
VCGD	正値対称スパース行列の連立 1 次方程式 （前処理付き CG 法，対角形式格納法）	74
VCGE	正値対称スパース行列の連立 1 次方程式 （前処理付き CG 法，ELLPACK 形式格納法）	80
VCRD	非対称または不定値のスパース実行列の連立 1 次方程式 （MGCR 法，対角形式格納法）	96
VCRE	非対称または不定値のスパース実行列の連立 1 次方程式 （MGCR 法，ELLPACK 形式格納法）	100

VQMRD	非対称または不定値のスパース実行列の連立 1 次方程式（QMR 法，対角形式格納法）	166
VQMRE	非対称または不定値のスパース実行列の連立 1 次方程式（QMR 法，ELLPACK 形式格納法）	170
VTFQD	非対称または不定値のスパース実行列の連立 1 次方程式（TFQMR 法，対角形式格納法）	206
VTFQE	非対称または不定値のスパース実行列の連立 1 次方程式（TFQMR 法，ELLPACK 形式格納法）	210
VMBV	実バンド行列と実ベクトルの積	130
VMVSD	スパース実行列と実ベクトルの積（対角形式格納法）	160
VMVSE	スパース実行列と実ベクトルの積(ELLPACK 形式格納法）	163

固有値・固有ベクトル

サブルーチン名	項　　目	ページ
VHEVP	エルミート行列の固有値・固有ベクトル	103
VLAND	実対称スパース行列の固有値・固有ベクトル（Lanczos 法, 対角形式格納法）	108
VSEVP	実対称行列の固有値・固有ベクトル	185
VTDEV	実 3 重対角行列の固有値・固有ベクトル	200

変換

サブルーチン名	項　　目	ページ
VCFM1	1 次元離散型複素フーリエ変換（2, 3, 5 および 7 の混合基底）	66
VCFT3	1 次元離散型複素フーリエ変換（2 基底, 一定間隔を持つデータ）	70
VCPF1	1 次元素因子離散型複素フーリエ変換	86
VCPF3	3 次元素因子離散型複素フーリエ変換	90
VMCF2	1 次元，多重，多次元離散型複素フーリエ変換（複素配列格納，混合基底）	133
VMCFT	1 次元，多重，多次元離散型複素フーリエ変換（実虚別配列，混合基底）	137
VMRF2	1 次元，多重，多次元離散型実フーリエ変換（混合基底）	145
VMRFT	多重，多次元離散型実フーリエ変換（2，3 および 5 の混合基底）	151
VRPF3	3 次元素因子離散型実フーリエ変換	180

VSRFT	1 次元，多重離散型実フーリエ変換 （2，3 および 5 の混合基底）	196
VMCST	離散型 cosine 変換	142
VMSNT	離散型 sine 変換	157
VCCVF	複素データの離散畳み込み および離散相関	60
VRCVF	実データの離散畳み込み および離散相関	174
VWFLT	ウェーブレットフィルターの生成	213
V1DWT	1 次元ウェーブレット変換	216
V2DWT	2 次元ウェーブレット変換	220

乱数

サブルーチン名	項　　目	ページ
DVRAN3	正規乱数の生成（倍精度）	23
DVRAN4	正規乱数の生成（倍精度，Wallace 法）	27
DVRAU4	一様乱数［0，1）の生成（倍精度）	31

第 I 部　概説

第 1 章　SSL II 拡張機能 II の概要

大規模な科学技術計算で使われる，以下のアルゴリズムを提供します．

(1) 倍精度乱数（一様／正規）

大規模シミュレーションを考慮して，短くても 10^{52} 以上の長周期で統計的特性のよい乱数を提供しています．正規乱数は Polar 法の他に，さらに高速な Wallace 法も提供しています．ただし，統計的特性が厳しく問われる場合は比較的，前者のほうが良好です．

(2) スパース行列の連立 1 次方程式（正値対称行列／非対称または不定値の実行列）

反復法を用いて，スパース行列の連立 1 次方程式を解きます．メモリの使用を抑え，大規模問題を，高速に解くことを可能にします．データの格納方法については，“第 3 章　データの格納方法”を参照してください．

正値対称行列には，CG 法（Conjugate Gradient method）を提供しています．CG 法では前処理として Neumann 級数の 1 次近似と修正不完全コレスキー分解による 2 種類を指定することができます．楕円型の偏微分方程式を離散化して得られた連立 1 次方程式に対しては，修正不完全コレスキー分解による前処理が有効です．

非対称または不定値の実行列に関しては，堅固で高速な MGCR 法（Modified Generalized Conjugate Residuals method）を提供しています．

非対称または不定値の実行列に関しては，更に高速な QMR 法（Quasi Minimal Residual method）および TFQMR 法（Transpose-Free Quasi-Minimal Residual method）および BICGSTAB(*l*)法（Bi-Conjugate Gradient Stabilized (*l*) method）を提供しています．これらの解法を使う上での指針に関しては，“第 I 部　概説　第 4 章　連立 1 次方程式の反復解法と収束性”を参照してください．

(3) スパース実行列と実ベクトルの積

(4) 実 3 重対角行列の連立 1 次方程式

大規模な実三重対角行列の連立 1 次方程式を QR 分解により高速に解く方法を提供しています．

(5) バンド行列の連立 1 次方程式（正値対称／実行列）

ベクトル計算機での性能を重視したデータ格納法およびアルゴリズムを採用したものを提供しています．直接法は堅固ではありますが，メモリーをバンド幅分使用するため，次数の大きいスパースな構造を持つバンド行列には向きません．この場合は，スパース

向きの格納法を利用した上記反復法をご利用ください.

(6) 固有値問題

大規模な実対称スパース行列の最大または／および最小固有値のいくつかと対応する固有ベクトルを求めるために Lanczos 法を提供しています.

実 3 重対角行列の固有値・固有ベクトル，実対称行列の固有値・固有ベクトルおよびエルミート行列の固有値・固有ベクトルを高速に求める解法を提供しています.

(7) フーリエ変換

ベクトル計算機で高性能な，多重フーリエ変換および多次元フーリエ変換を行う機能（混合基底，複素数／実数）を提供します．この機能は，1 次元フーリエ変換も高速です．ならびに 3 次元素因子フーリエ変換（複素数／実数）を提供します.

また，スカラー計算機で高性能な，多重フーリエ変換および多次元フーリエ変換を行う機能（複素数／実数）を提供します.

(8) 畳み込み，相関

信号処理分野においてよく現れる，畳み込み および相関を行う機能（複素数／実数）を提供します.

(9) ウェーブレット変換

高性能な 1 次元および 2 次元ウェーブレット変換を提供しています.

第 2 章　SSL II 拡張機能 II の一般規約

各機能について共通な一般規約について記述します.

2.1 サブルーチンの精度

単精度および倍精度のルーチンを提供します．ただし，乱数ルーチンは倍精度版だけです.

2.2 サブルーチン名

単精度のルーチンは V から始まり，倍精度のルーチンの先頭は DV で始まります．スレーブルーチンは，単精度は U で，倍精度は DU で始まります.

2.3 パラメタ

(1) パラメタの並びの順

パラメタの並びの順序は SSL II 標準機能と同じく，原則的に次の形式に従っています.
（入出力パラメタの並び，入力パラメタの並び，出力パラメタの並び，ICON）

(2) パラメタの型

先頭が I, J, K, L, M, N で始まるものは整数型です．Z で始まるものは複素数型です．その他の文字で始まるものは, 特別な指定がなければ, 単精度用のルーチンは単精度実数型を, 倍精度用のルーチンは倍精度実数型を表します.

2.4 コンディションコード

実行後の状態を示すパラメタ ICON が用意されています.
コードは 0 から 39999 の値をとりますが，結果が保証されているか否かに応じて表 2.1 のように区分されています.

表 2.1　コンディションコードの区分

コード	意　　味	結果の保障	区分
0	正常に処理が終了している.	結果は保障される.	正常
1 ～ 9999	正常に処理が終了しているが，なんらかの補助的な情報を含んでいる.		
10000 ～ 19999	処理の過程で，内部的な制限を加えることにより，一応の処理は終了している.	制限付きで結果は保障される.	警告
20000 ～ 29999	処理の過程で異常を生じ，処理が打ち切られた.	結果は保障されない.	異常
30000 ～ 39999	入力パラメタにエラーがあったため，処理が打ち切られた.		

第 3 章　データの格納方法

SSL II 拡張機能 II ではバンド行列およびスパース行列の連立 1 次方程式の解法のために以下の格納方法を利用しています.

3.1 バンド行列

SSL II の標準的なバンド行列の格納方法を使用せずに, ベクトル計算機での性能上有利な格納方法を採用しています. バンド行列に関する各ルーチンの使用方法の記述を参照してください.

3.2 スパース行列

3.2.1 スパース行列の格納方法

スパース行列に対する各機能は ELLPACK 形式の格納方法および対角形式の格納方法のおのおのに対して提供しています.

ELLPACK 形式の格納方法は, 係数行列の各行ベクトルの非零要素のみを圧縮して格納する方法です.

また, 対角形式の格納方法は, 非零要素のある対角方向のベクトルを格納する方法です.

3.2.1.1 一般スパース行列の格納方法

a. 一般スパース行列の ELLPACK 形式の格納方法

ELLPACK 形式（“付録　参考文献一覧表”の[23],[34]参照）のスパース行列の格納方法では, 係数行列 ***A*** の行ベクトルの非零要素を圧縮して配列 ***COEF*** の対応する行ベクトルに格納します. さらに, ***COEF*** に格納された非零要素が何番目の列ベクトルにあるかを示す値を, 対応する ***ICOL*** の配列要素に格納します. ***COEF*** に係数行列 ***A*** の行ベクトルの非零要素を格納するとき, 必ずしも左詰めである必要はありません.

格納には, 2 つの配列 ***COEF***(*K*,*IWIDT*) および ***ICOL***(*K*,*IWIDT*) の ***COEF***(1:*N*,*), ***ICOL***(1:*N*,*) 部分を使います.

行列 ***A*** の行ベクトルに現れる非零要素の最大数を *nz*, 係数行列の次数を *n* とすると, *IWIDT* ≧*nz*, *K*≧*n* です.

係数行列***A***の行ベクトルにある非零要素の数が*IWIDT*より少ないときは, 余った配列***COEF***の行ベクトルの要素には零を設定し, 対応する ***ICOL*** の配列要素には, 何番目の行ベクトルかを示す値を設定（***COEF*** (*i*, *j*)＝0,***ICOL*** (*i*, *j*)＝*i* とします）してください.

例：　係数行列 A を $COEF$，$ICOL$ を使って格納します．

$$A = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 3 & 4 & 0 \\ 0 & 0 & 5 & 0 \\ 6 & 0 & 0 & 0 \end{bmatrix} \Rightarrow \quad COEF = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 0 \\ 6 & 0 \end{bmatrix}, \quad ICOL = \begin{bmatrix} 1 & 4 \\ 2 & 3 \\ 3 & 3 \\ 1 & 4 \end{bmatrix}$$

b. 一般スパース行列の対角形式の格納方法

対角形式（“付録参考文献一覧表”の[27],[31]参照）のスパース行列の格納方法は，非零要素の存在する対角方向のベクトルを配列 ***DIAG*** の列ベクトルとして格納します．

ある整数 k に対して，次の対角方向のベクトルを対角ベクトルと本マニュアルでは呼びます．特に，対角要素からなるベクトルを主対角ベクトルと呼びます．

$(a_{1,1+k}, a_{2,2+k}, ..., a_{n,n+k})$

ただし $i + k < 1$ または $i + k > n$ のとき $a_{i,i+k} = 0$ とします．

対角ベクトルを配列 ***DIAG*** に格納する順序に特に制限はありません．

また，***NOFST*** (i) には，***DIAG***$(*, i)$ に格納される対角ベクトルの主対角ベクトルからの距離を格納します．上記対角ベクトルで k が距離を表します．主対角要素からなる対角ベクトルの距離が 0 で, 上三角行列にある対角ベクトルの距離は正の整数で, 下三角行列にある対角ベクトルの距離は負の整数で表します．

NOFST $(m) = k$ なら，***DIAG*** $(i, m) = a_{i,i+k} (i = 1, ..., n)$ のように格納します．

2 つの配列 ***DIAG*** $(K,NDIAG)$ および ***NOFST*** $(NDIAG)$ を使います．係数行列は ***DIAG*** $(1{:}N,NDIAG)$ に格納されます．格納すべき対角ベクトルの本数を nd，係数行列の次数を n とすると，$NDIAG \geqq nd, K \geqq n$ です．

例：　係数行列 ***A*** を ***DIAG,NOFST*** を使って格納します．

$$A = \begin{bmatrix} 0 & 1 & 2 & 3 & 0 \\ 4 & 5 & 0 & 6 & \\ 0 & 7 & 8 & 9 & 0 \\ 0 & 0 & 10 & 11 & 0 & 0 \end{bmatrix} \Rightarrow \quad DIAG = \begin{bmatrix} 1 & 2 & 3 & 0 \\ 5 & 0 & 6 & 4 \\ 8 & 9 & 0 & 7 \\ 11 & 0 & 0 & 10 \end{bmatrix}$$

$$NOFST = (0\ 1\ 2\ -1)$$

3.2.1.2　正値対称スパース行列の格納方法

正規化された正値対称行列の上三角行列部分と下三角行列部分を ELLPACK 形式の格納方法，または対角形式の格納方法で，この順番に格納します．

正値対称な行列 $\boldsymbol{A}$ はその対角要素の平方根の逆数を対角要素に持つ対角行列によって，対角要素が 1 の対称行列 $\boldsymbol{A}^*$に正規化することができます．

$$\boldsymbol{A}^* = \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$$

ここで

$$\begin{aligned} D^{-1/2} &= diag(a_{11}^{-1/2}, a_{22}^{-1/2}, ..., a_{nn}^{-1/2}) \\ &= diag(d_1^{-1/2}, d_2^{-1/2}, ..., d_n^{-1/2}) \end{aligned}$$

次数 n の連立 1 次方程式

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

は，正規化された行列 $\boldsymbol{A}^*$の連立 1 次方程式に変換することができます．

$$(D^{-1/2}AD^{-1/2})(D^{1/2}x) = D^{-1/2}b$$

$$\boldsymbol{A}^*\boldsymbol{x}^* = \boldsymbol{b}^*$$

$$a_{ij}^* = a_{ij} d_i^{-1/2} d_j^{-1/2}, b_i^* = b_i d_i^{-1/2}$$

$$x_i^* = x_i d_i^{1/2}, i = 1, ..., n, \; j = 1, ..., n$$

a.　正値対称スパース行列の ELLPACK 形式の格納方法

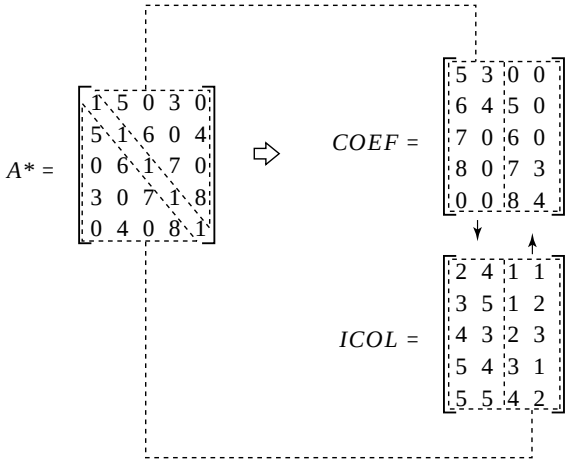
対角要素が 1 に正規化された正値対称スパース行列の上三角行列部分および下三角行列部分を，おのおの一般スパース行列の ELLPACK 形式の格納方法で格納したものを 1 つの配列 *COEF* にまとめて格納します．上三角行列部分を最初に格納しその後に下三角行列部分を格納します．

つまり，上三角行列部分の各行ベクトルでの非零要素の最大値を *NSU* とし，さらに下三角行列部分の各行ベクトルでの非零要素の最大値を *NSL* とします．

NSH＝max(*NSU*, *NSL*) としたとき，***COEF*** (＊, 1 : *NSH*) に上三角行列部分の非零要素を格納します．***COEF*** (＊, *NSH*＋1 : 2×*NSH*) に下三角行列部分の非零要素を格納します．

余った配列 *COEF* の要素には零を設定し，対応する ***ICOL*** の配列要素には，何番目の行ベクトルかを示す値を設定（***COEF*** (*i*, *j*)＝0,***ICOL*** (*i*, *j*)＝*i* とします）してください．

例：正規化された係数行列 ***A**** の上三角部分および下三角部分を ELLPACK 形式で格納します.



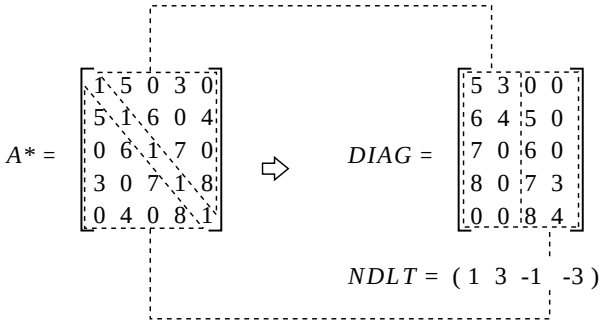
b.　正値対称スパース行列の対角形式の格納方法

対角要素が 1 に正規化された正値対称スパース行列の上三角行列部分および下三角行列部分をおのおのを一般スパース行列の対角形式の格納法で格納したものを, 1 つの配列 ***DIAG*** にまとめて格納します. 上三角行列部分（下三角行列部分）の非零要素のある対角ベクトルの本数を *NDT* とすると，***DIAG*** (1 : *NDT*) に上三角行列部分を，***DIAG*** (*NDT*＋1 : *NW*) に下三角行列部分を格納します.

このとき，上三角行列部分は距離 (***NDLT*** の値) に関して昇順に，下三角行列部分に関しては，降順に格納しなければなりません.

この方法は配列 ***DIAG*** (*K*, *NW*) および ***NDLT*** (*NW*) を利用します. *NW*＝2×*NDT* です.

例：正規化された係数行列 ***A**** の上三角部分および下三角部分を対角形式の格納法で格納します.



3.2.1.3　格納法の選択基準

スパース行列が，その非零要素が係数行列の対角方向ベクトルに集中して存在する構造を持つときは，対角形式の格納方法をお使いください.

3.2.2　偏微分演算子の離散化とその格納例

楕円型の偏微分方程式を離散化して，連立 1 次方程式を構成して問題を解く上で現れる，代表的なスパースな係数行列の構成方法について述べます．実際問題を解く上でこれらの係数行列は，ELLPACK 形式および対角形式で格納する必要があります．

離散化した結果が非対称スパース行列となる楕円型の偏微分演算子を，3 次元領域でディリクレ境界条件で離散化したときの係数行列の構成と，生成された係数行列をスパース行列の格納方法で格納するサブルーチンを以下に示します．

この係数行列を持つ連立 1 次方程式は，(D)VCRE または(D)VCRD で解くことができます．

a.　楕円型偏微分演算子の離散化と係数行列の構成

演算子 L

$$Lu = -\Delta u + a_1 \frac{\partial u}{\partial x} + a_2 \frac{\partial u}{\partial y} + a_3 \frac{\partial u}{\partial z} + cu$$

（ただし，Δ はラプラシアンです．$\Delta \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ ）

領域 $\Omega = [0,l_x]\times[0,l_y]\times[0,l_z]$

境界条件　ディリクレ境界条件　Ω の境界で $u=0$

ここで a_1, a_2, a_3 および c は定数です．

Ω の各次元を n_x+1, n_y+1, n_z+1 個の部分区間に等分割すると，Ω の内部に $n_x\times n_y\times n_z$ の格子点が存在します．格子点上での各変数 x, y, z の値を (x_i, y_j, z_k) で表すと（$1\leqq i\leqq n_x$, $1\leqq j\leqq n_y$, $1\leqq k\leqq n_z$）格子点上での関数 u の値は，$u_{i,j,k}=u(x_i, y_j, z_k)$ です．

このとき，変数 x に関する偏微分係数は以下のように近似します．

$$\frac{\partial u}{\partial x}(x_i, y_j, z_k) \fallingdotseq (u_{i+1,j,k} - u_{i-1,j,k})(n_x+1)/(2l_x)$$

$$\frac{\partial u^2}{\partial x^2}(x_i, y_j, z_k) \fallingdotseq (u_{i+1,j,k} - 2u_{i,j,k} + u_{i-1,j,k})(n_x+1)^2/l_x^2$$

変数 y および z に関する偏微分係数も同様に近似できます．

Ω の境界で関数 $u=0$ を考慮して，$\boldsymbol{L}u$ を離散化して係数行列 $\boldsymbol{A}$ を構成して，$\boldsymbol{L}u \fallingdotseq \boldsymbol{A}\boldsymbol{v}$ と近似します．

ここで v は関数 u の格子点上での値からなるベクトル $v=(v_1, v_2, ..., v_n)$ で，$v_m=u_{i,j,k}$, $m=(k-1)n_xn_y+(j-1)n_x+i$ の対応関係があります．

b.　係数行列をスパース行列の格納法で格納するサブルーチン

このようにして離散化した演算子を ELLPACK 形式または対角形式で格納するサブルーチンの例を以下に示します．

サブルーチン INIT_MAT_ELL は ELLPACK 形式で，INIT_MAT_DIAG は対角形式で係数行列を格納します．

引数の n_x, n_y, n_z は NX, NY, NZ に，l_x，l_y，l_z が XL, YL, ZL に，a_1, a_2, a_3 および c は VA1, VA2, VA3,VC に対応します．

INIT_MAT_ELL では係数行列は，A_L および ICOL_L に，INIT_MAT_DIAG では D_L, OFFSET に格納されます．
IWIDTH＝7 を指定して呼び出すと，3 次元の領域Ω での係数行列が生成されます．
NDIVP＝$n_x n_y n_z$ を指定して呼び出します．
（このサブルーチンを利用するときには，IWIDTH の値は 7 を超えてはいけません．7 以下のとき（例えば 5 または 3）対角列は対応して減少します．IWIDTH＝5 か IWIDTH＝3 のとき，問題はおのおの 2 次元と 1 次元の問題になります．IWIDTH＝5 のとき，NZ＝1 に，また IWIDTH＝3 のとき NZ＝1，NY＝1 にするとよいです．IWIDTH が 7, 5, 3 以外の場合は特別の意味はなく，テストのために使用できます．）

例 1：ELLPACK形式の格納法で上記偏微分演算子を離散化して格納するサブルーチンを示します．

```
      SUBROUTINE INIT_MAT_ELL(VA1,VA2,VA3,VC,A_L,ICOL_L,NX,NY,NZ,
     &    XL,YL,ZL,IWIDTH,NDIVP,LD)
      IMPLICIT NONE
      INTEGER NX,NY,NZ,IWIDTH,NDIVP,LD
      DOUBLE PRECISION A_L(LD,IWIDTH)
      DOUBLE PRECISION VA1,VA2,VA3,VC,XL,YL,ZL
      INTEGER ICOL_L(LD,IWIDTH)

      DOUBLE PRECISION HX,HY,HZ
      INTEGER I,J,L,JS,IWIDTH_LOC
      INTEGER I0,J0,K0

      IF (IWIDTH .LT. 1)THEN
          WRITE (*,*) 'SUBROUTINE INIT_MAT_ELL:'
          WRITE (*,*) ' IWIDTH SHOULD BE GREATER THAN OR EQUAL TO 1'
          RETURN
      ENDIF
      IWIDTH_LOC = IWIDTH
C     IWIDTH CANNOT BE GREATER THAN 7
      IF (IWIDTH .GT. 7) IWIDTH_LOC = 7

C     INITIAL SETTING
      HX = XL/(NX+1)
      HY = YL/(NY+1)
      HZ = ZL/(NZ+1)

      DO 110 J = 1,IWIDTH
          DO 100 I = 1,NDIVP
              A_L(I,J) = 0.0
```

```
              ICOL_L(I,J) = I
 100      CONTINUE
 110  CONTINUE

C     MAIN LOOP
         DO 200 J = 1,NDIVP
         JS = J
         L = 1
C        DECOMPOSE JS-1 = (K0-1)*NX*NY+(J0-1)*NX+I0-1
         K0 = (JS-1)/NX/NY+1
         IF (K0 .GT. NZ) RETURN
         J0 = (JS-1-NX*NY*(K0-1))/NX+1
         I0 = JS - NX*NY*(K0-1) - NX*(J0-1)

         IF (IWIDTH_LOC .GE. 7) THEN
             IF (K0 .GT. 1) THEN
                 A_L(J,L) = -(1.0/HZ+0.5*VA3)/HZ
                 ICOL_L(J,L) = JS-NX*NY
                 L =L+1
             ENDIF
         ENDIF
         IF (IWIDTH_LOC .GE. 5) THEN
             IF (J0 .GT. 1) THEN
                 A_L(J,L) = -(1.0/HY+0.5*VA2)/HY
                 ICOL_L(J,L) = JS-NX
                 L =L+1
             ENDIF
         ENDIF
         IF (IWIDTH_LOC .GE. 3) THEN
             IF (I0 .GT. 1) THEN
                 A_L(J,L) = -(1.0/HX+0.5*VA1)/HX
                 ICOL_L(J,L) = JS-1
                 L =L+1
             ENDIF
         ENDIF
         A_L(J,L) = 2.0/HX**2+VC
         IF (IWIDTH_LOC .GE. 5) THEN
             A_L(J,L) = A_L(J,L) + 2.0/HY**2
             IF (IWIDTH_LOC .GE. 7) THEN
                 A_L(J,L) = A_L(J,L) + 2.0/HZ**2
             ENDIF
         ENDIF
```

```
            ICOL_L(J,L) = JS
            L = L+1
            IF (IWIDTH_LOC .GE. 2) THEN
                IF (I0 .LT. NX) THEN
                    A_L(J,L) = -(1.0/HX-0.5*VA1)/HX
                    ICOL_L(J,L) = JS+1
                    L = L+1
                ENDIF
            ENDIF
            IF (IWIDTH_LOC .GE. 4) THEN
                IF (J0 .LT. NY) THEN
                    A_L(J,L) = -(1.0/HY-0.5*VA2)/HY
                    ICOL_L(J,L) = JS+NX
                    L = L+1
                ENDIF
            ENDIF
            IF (IWIDTH_LOC .GE. 6) THEN
                IF (K0 .LT. NZ) THEN
                    A_L(J,L) = -(1.0/HZ-0.5*VA3)/HZ
                    ICOL_L(J,L) = JS+NX*NY
                ENDIF
            ENDIF

 200  CONTINUE
      RETURN
      END
```

例 2：対角形式の格納法で上記偏微分演算子を離散化して格納するサブルーチンを示します.

```
      SUBROUTINE INIT_MAT_DIAG(VA1,VA2,VA3,VC,D_L,OFFSET,
     &    NX,NY,NZ,XL,YL,ZL,NDIAG,NDIVP,LD)
      IMPLICIT NONE
      INTEGER NX,NY,NZ,NDIAG,NDIVP,LD
      DOUBLE PRECISION D_L(LD,NDIAG)
      DOUBLE PRECISION VA1,VA2,VA3,VC,XL,YL,ZL
      INTEGER OFFSET(NDIAG)

      DOUBLE PRECISION HX, HY, HZ
      INTEGER I,J,L,JS,NXY,NDIAG_LOC
      INTEGER J0,I0,K0

      IF (NDIAG .LT. 1)THEN
          WRITE (*,*) 'SUBROUTINE INIT_MAT_DIAG:'
```

```
          WRITE (*,*) ' NDIAG SHOULD BE GREATER THAN OR EQUAL TO 1'
          RETURN
        ENDIF
        NDIAG_LOC = NDIAG
        IF (NDIAG .GT. 7) NDIAG_LOC = 7

C       INITIAL SETTING
        HX = XL/(NX+1)
        HY = YL/(NY+1)
        HZ = ZL/(NZ+1)

        DO 110 J = 1,NDIAG
          DO 100 I = 1,NDIVP
            D_L(I,J) = 0.0
 100      CONTINUE
 110    CONTINUE

C       OFFSET SETTING
        L = 1
        NXY = NX*NY
        IF (NDIAG_LOC .GE. 7) THEN
          OFFSET(L) = -NXY
          L = L+1
        ENDIF
        IF (NDIAG_LOC .GE. 5) THEN
          OFFSET(L) = -NX
          L = L+1
        ENDIF
        IF (NDIAG_LOC .GE. 3) THEN
          OFFSET(L) = -1
          L = L+1
        ENDIF
        OFFSET(L) = 0
        L = L+1
        IF (NDIAG_LOC .GE. 2) THEN
          OFFSET(L) = 1
          L = L+1
        ENDIF
        IF (NDIAG_LOC .GE. 4) THEN
          OFFSET(L) = NX
          L = L+1
        ENDIF
        IF (NDIAG_LOC .GE. 6) THEN
          OFFSET(L) = NXY
        ENDIF
```

```
C     MAIN LOOP
      DO 200 J = 1,NDIVP
         JS = J
C        DECOMPOSE JS-1 = (K0-1)*NX*NY+(J0-1)*NX+I0-1
         K0 = (JS-1)/NXY+1
         IF (K0 .GT. NZ) RETURN
         J0 = (JS-1-NXY*(K0-1))/NX+1
         I0 = JS - NXY*(K0-1) - NX*(J0-1)

         L = 1
         IF (NDIAG_LOC .GE. 7) THEN
             IF (K0 .GT. 1) D_L(J,L) = -(1.0/HZ+0.5*VA3)/HZ
             L =L+1
         ENDIF
         IF (NDIAG_LOC .GE. 5) THEN
             IF (J0 .GT. 1) D_L(J,L) = -(1.0/HY+0.5*VA2)/HY
             L =L+1
         ENDIF
         IF (NDIAG_LOC .GE. 3) THEN
             IF (I0 .GT. 1) D_L(J,L) = -(1.0/HX+0.5*VA1)/HX
             L =L+1
         ENDIF
         D_L(J,L) = 2.0/HX**2+VC
         IF (NDIAG_LOC .GE. 5) THEN
             D_L(J,L) = D_L(J,L) + 2.0/HY**2
             IF(NDIAG_LOC .GE. 7) THEN
                D_L(J,L) = D_L(J,L) + 2.0/HZ**2
             ENDIF
         ENDIF
         L = L+1
         IF (NDIAG_LOC .GE. 2) THEN
             IF (I0 .LT. NX) D_L(J,L) = -(1.0/HX-0.5*VA1)/HX
             L =L+1
         ENDIF
         IF (NDIAG_LOC .GE. 4) THEN
             IF (J0 .LT. NY) D_L(J,L) = -(1.0/HY-0.5*VA2)/HY
             L =L+1
         ENDIF
         IF (NDIAG_LOC .GE. 6) THEN
             IF (K0 .LT. NZ) D_L(J,L) = -(1.0/HZ-0.5*VA3)/HZ
         ENDIF
 200  CONTINUE
      RETURN
      END
```

第 4 章　連立 1 次方程式の反復解法と収束性

4.1 スケーリング

スパース行列の連立 1 次方程式の反復解法を利用するとき，行列の要素を均衡化するためにスケーリングを行なうことをすすめます．行列の正規化は数値的安定性と反復法の収束率を強力に改善します．
正規化された係数行列 $\hat{\boldsymbol{A}}$ は対角要素が非負でありかつ各行の要素の絶対値の和がほぼ 1 であるべきです．

$$\boldsymbol{Ax}=\boldsymbol{b} \tag{1}$$

連立一次方程式(1) の正規化された形は，係数行列に左から対角行列 $\boldsymbol{L}$ を，右から対角行列 $\boldsymbol{R}$ を掛けることで構成することができます．新しい変数 $\hat{\boldsymbol{x}} = \boldsymbol{R}^{-1}\boldsymbol{x}$ を導入して方程式(1)は次のようになります．

$$\boldsymbol{LAR}\,\hat{\boldsymbol{x}} = \boldsymbol{Lb} \quad \Leftrightarrow \quad \hat{\boldsymbol{A}}\hat{\boldsymbol{x}} = \hat{\boldsymbol{b}} \tag{2}$$

ここで，$\hat{\boldsymbol{A}} = \boldsymbol{LAR}$，$\hat{\boldsymbol{b}} = \boldsymbol{Lb}$ です．
$\boldsymbol{A}$ の代わりに正規化された行列 $\hat{\boldsymbol{A}}$ を反復法で使うことを薦めます．このとき，解法を呼び出す前に右辺の $\boldsymbol{b}$ に $\boldsymbol{L}$ を掛けることおよび返却された近似解に $\boldsymbol{R}$ を掛けて変換することが必要です．

$s_i = \sum_{j=1}^{n} \left| a_{ij} \right|$ を i 番目の行の要素の絶対値の和とします．

$$L_{ij} = \begin{cases} \dfrac{\mathrm{sgn}(a_{ii})}{\sqrt{s_i}} & i = j \\ \text{if} & \\ 0 & i \neq j \end{cases} \tag{3}$$

$$R_{ij} = \begin{cases} \dfrac{1}{\sqrt{s_i}} & i = j \\ \text{if} & \\ 0 & i \neq j \end{cases} \tag{4}$$

反復法の収束率に影響を与える正規化を行なう他の方法があることに注意してください．例えば，“付録　参考文献一覧表”の[44]を参照してください．
(3),(4)を選ぶと $\boldsymbol{A}$ が正値対称であれば，$\hat{\boldsymbol{A}}$ も正値対称であることに注意してください．

4.2 行列の対称性と反復解法

a. 対称行列
行列 A が対称，すなわち $a_{ij}=a_{ji}$ $(i, j = 1,...,n)$で正値であれば，古典的共役勾配法(“付録 参考文献一覧表”の[21]参照．)が連立一次方程式を解くために使えます．行列が正値でないと反復計算が続けられなくなります(ブレークダウンを起こす).

b. 非対称行列
係数行列が非対称または不定値の場合の解法も使えます．与えられた連立 1 次方程式に対する最適な解法は係数行列 $\boldsymbol{A}$ の特性 (または正規化された係数行列 $\hat{\boldsymbol{A}}$ の特性)によって変わります．異なったクラスの行列に対して以下の解法が利用できます．各解法の特性や反復を始めるために設定された初期ベクトルの値により，反復法の計算過程で計算が続けられなくなることがあります(ブレークダウンを起こす)．解法を変えるか，反復を始める初期ベクトルを変えて試みることを勧めます．

4.3 行列の固有値分布と反復解法の収束

a. MGCR 法
係数行列の固有値の分布が正の実軸に近いとき(図 4.3-1 参照)，MGCR 法（“付録　参考文献一覧表”の[25]参照．）で検索方向の数を小さく（5～10）して使えます（検索方向の意味はここでは省略）．固有値の虚部が大きい固有値があるとき，収束をよくするために検索方向の数を増やす必要があります．検索方向の数を増やすことは，計算過程をより安定にするはたらきがありますが, 計算ステップ毎の計算量とデータ格納量が増えます. MGCR 法では，与えられた問題に対して，まずは検索方向の数を小さくして試行して見るのがよいでしょう．

b. TFQMR 法:
係数行列の固有値の実部が正で複素平面の右半分にあり（図 4.3-2 参照），かつ虚部が大きい場合には，TFQMR 法（“付録　参考文献一覧表”の[12]参照．）を薦めます．また，解法は固有値の実部の最小値が出来る限り大きいとき最も収束が良くなります. そのため，例えば，実部がゼロでない非常に小さな正数値となる固有値があるとき，収束は悪くなります．TFQMR 法の収束は，検索方向数に大きな数を指定した MGCR 法よりも悪いかもしれません. しかし, TFQMR 法の反復の各ステップは大きな検索方向数を指定した MGCR 法よりコストは低いです．その結果少ない CPU time で計算できます．つまり TFQMR 法は堅固であるが，小さな検索方向数を指定した MGCR 法より遅いことになります．

c. BICGSTAB(l)
TFQMR 法に似て，BICGSTAB(l)法（“付録　参考文献一覧表”の[39]参照．）は固有値の

分布が実部が正の複素平面の右半分にあるときに適しています．また，解法は固有値の実部の最小値が出来る限り大きいとき最も収束が良くなります．そのため，例えば，実部がゼロでない非常に小さな正数値となる固有値があるとき，収束は悪くなります．係数行列の固有値の分布が正値の実軸に近いある種の問題で，BICBSTAB(*l*)法は，小さな検索方向数を指定した MGCR 法よりもさらに収束率が速いです．しかし，BICGSTAB(*l*)法の各反復ステップは，行列ベクトル積を２回必要なため非常にコストがかかります．このため，ある種の問題では MGCR 法または TFQMR 法の方が BICGSTAB(*l*)法より高速です．しかし BICGSTAB(*l*)法はより堅固な方法です．

正規化された係数行列の固有値についての情報がないときは，MGCR 法，TFQMR 法，BICGSTAB(*l*)法をそれぞれ試すことを提案します．３つの方法を試す順序については，まず，一般的に堅固でなくても速い方法を第一に選びます．その意味で MGCR 法を検索方向の数を 5～10 にして最初に試すことを薦めます．そこでうまくいかなければ他の堅固な方法に進みます．ある問題に対してどの方法が最も適合しているかを見るには，例えば収束判定値 EPS を 0.1 程度にして各方法の計算時間を測定してみるのが適当です．

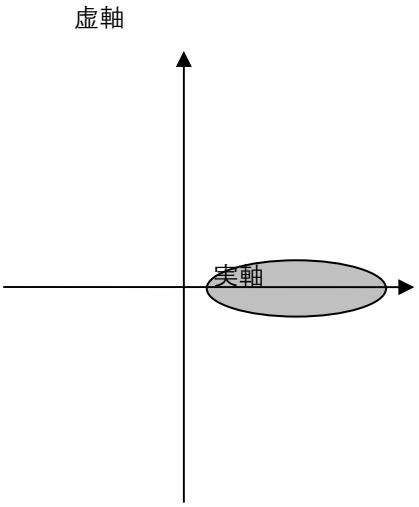


図 4.3-1：MGCR 法が収束する固有値の分布

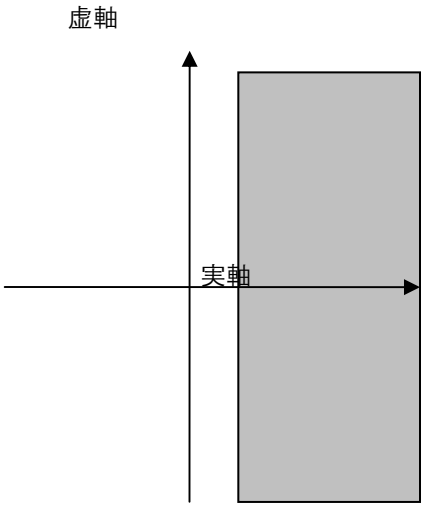


図 4.3-2：TFQMR 法および BICGSTAB(*l*)法が収束する固有値の分布

第 II 部　サブルーチンの使用方法

J11-20-0401 DVRAN3

正規乱数表の生成（倍精度）
CALL DVRAN3(DAM,DSD,IX,DA,N,DWORK,NWORK,ICON)

(1)　機能

与えられた平均 m と標準偏差σを持った正規分布の密度関数（1.1）から，擬似乱数を生成します.

$$f(x)=\frac{1}{\sigma\sqrt{2\pi}}\exp(\frac{-(x-m)^2}{2\sigma^2}) \tag{1.1}$$

(2)　パラメタ

DAM .....入力．正規分布の平均 m.
　　倍精度実数型.

DSD .....入力．正規分布の標準偏差 σ.
　　倍精度実数型.

IX .....入力．出発値.
　　初回呼び出しでは，IX に正の値を与え，2 回目以降の呼び出しでは返却値 0 のまま呼び出してください．初回呼び出しで指定する出発値が異なると，異なる乱数列が生成されます.
　　（(3) 使用上の注意 b.注意①参照）.
　　4 バイト整数型（INTEGER*4）.
　　出力．0.

DA.....出力．N 個の擬似乱数.
　　N の大きさをもつ倍精度実数型 1 次元配列.

N.....入力．DA に返却される正規分布擬似乱数の個数.
　　（(3) 使用上の注意 b.注意②参照）.

DWORK .....作業領域．大きさ NWORK の倍精度実数型 1 次元配列.
　　本サブルーチンを繰り返し呼び出す場合は, 内容を変更しないでください.
　　DWORK には，本サブルーチンが再度呼び出される場合に必要な情報が格納されています.
　　（(3) 使用上の注意 b.注意③参照）.

NWORK .....入力．配列 DWORK の大きさ．NWORK≧1156.

ICON.....出力．コンディションコード.
　　表 DVRAN3-1 参照.

表 DVRAN3-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	―
30001	NWORK が小さ過ぎた.	処理を打ち切る.
30002	IX＜0	
30003～30008	DWORK が変更されていた．または，初回呼び出しで IX に 0 が与えられた.	

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ DUF2G3,DUITG3,DURN3B,DURUG3,DUR2G3,DUSKG3,DUSQG3,
DUVRG3,DVRAU4,MGSSL

b.　注意

①　出発値 IX について

確定的な計算（deterministic program）で擬似乱数列を生成するには，何かランダムな入力が必要になります．このために，利用者が出発値 IX を与える必要があります．本サブルーチンの初回呼び出しでは，出発値 IX は正整数である必要があります（例外事項については，⑤を参照）．2 回目以降の呼び出しでは IX には 0 を指定してください. これは同じ乱数列から続いて乱数を求めるためです．プログラミングを容易にするため，本サブルーチンは初回呼び出し後 IX に 0 を返しています.

本サブルーチンは，最低でも周期が 10^{52} 以上である，長周期の一様乱数を利用する Polar 法で正規乱数を生成しています．異なる出発値は異なる乱数列，つまり長周期の乱数列を少なくとも $2^{60}＞10^{18}$ 以上の間隔を持って分割してできた異なる部分乱数列より乱数列を生成します．詳細は“DVRAU4（4）手法概要”を参照してください.

②　パラメタ N について

本サブルーチンは出発値 IX で定義される無限列から，次の N 個の擬似乱数を返却します．N≦0 の時は擬似乱数を返却しません.

効率良くするためには，利用者は N を十分大きく（例えば N＝100,000）します．それは，サブルーチン呼び出しのオーバーヘッドが減少し，しかもベクトル化されるためです．本サブルーチンを連続的に呼び出す途中で，N が変更される場合は，配列 DA を最大の N の大きさだけ確保する必要があります.

③　作業領域 DWORK について

DWORK は，複数回本サブルーチンを呼び出す場合，次の呼び出しのための情報を格納する作業領域として使用されます．従って，本サブルーチンを呼び出している間は，呼び出し側のプログラムで DWORK の内容を変更してはいけません.

④　パラメタ NWORK について

DWORK(1),..., DWORK(NWORK) は，本サブルーチンによって使用されます．NWORK は，本サブルーチンの各呼び出しで変えないでください．NWORK に

は少なくとも 1156 以上, ベクトルプロセッサ上では十分大きな数（例えば NWORK＝100,000）を与えると効率的です.

⑤ 同じ乱数の再生成について

DWORK(1),..., DWORK(NWORK) が保存される場合, その DWORK を再使用し, IX＝0 として本サブルーチンを呼び出すことにより,（DWORK が保存された時点からの）同じ乱数列が再生成されます.

c. 使用例

1,000,000 個の擬似正規乱数を生成し，1 次モーメント（変数値 X_i の度数を F_i としたとき， $\sum X_i F_i$ ），2 次モーメント（変数値 X_i の度数を F_i としたとき， $\sum X_i^2 F_i$ ）を計算します．出発値は 12345 とします.

```
C     **EXAMPLE**
      PARAMETER (NRAN = 1000000)
      PARAMETER (NSEED = 12345)
      PARAMETER (NWMAX = 100000)
      PARAMETER (NBUF = 120000)
      REAL*8 DA(NBUF)
      REAL*8 DWORK(NWMAX)
      REAL*8 DSUM,DSUM2
      REAL*8 DMEAN,DM2
      IA = NSEED
      PRINT *, ' Seed ', IA
      N = NBUF
      NWORK = NWMAX
      DSUM = 0.0D0
      DSUM2 = 0.0D0
C NGEN counts down to 0
      NGEN = NRAN
      PRINT *, ' Generating ', NGEN,
     $ ' numbers'
C Generate NRAN numbers ,
C maximum NBUF at a time
      KRPT = (NRAN+NBUF-1)/NBUF
      PRINT *, ' with ', KRPT,
     $ ' call to dvran3'
      DO 20 J = 1, KRPT
      N = MIN0 (NBUF, NGEN)
C First two arguments are mean
C add standard deviation
      CALL DVRAN3 (0.0D0, 1.0D0, IA,
     $ DA, N, DWORK, NWORK, ICON)
      IF (ICON .NE. 0) THEN
```

```
        PRINT *, ' Error Return ', ICON
        STOP
      ENDIF
C Accumulate sum of numbers generated
      DO 10 I = 1, N
      DSUM = DSUM + DA(I)
C Accumulate sum of squares
   10 DSUM2 = DSUM2 + DA(I)*DA(I)
   20 NGEN = NGEN - N
C Compute sample mean
      DMEAN = DSUM/DFLOAT(NRAN)
      PRINT *, ' First moment ', DMEAN
C Compute sample second moment about 0
      DM2 = DSUM2/DFLOAT(NRAN)
      PRINT *, ' Second moment ', DM2
      STOP
      END
```

(4)　手法概要

DVRAN3 は正規分布する擬似乱数を生成するために，初等関数の高速な計算法を利用した Polar 法（Polar method）を使用しています．この方法に必要な一様擬似乱数は DVRAU4 を利用して生成されます．

Polar 法は“付録　参考文献一覧表”の [24] に記述されています．実現方法の詳細および他の方法との比較は“付録　参考文献一覧表”の [4] を参照してください．

J11-20-0501 DVRAN4

正規乱数の生成（倍精度，Wallace 法）
CALL DVRAN4(DAM,DSD,IX,DA,N,DWORK,NWORK,ICON)

(1)　機能

与えられた平均 m と標準偏差σを持った正規分布の密度関数（1.1）から，擬似乱数を生成します．

$$f(x)=\frac{1}{\sigma\sqrt{2\pi}}\exp(\frac{-(x-m)^2}{2\sigma^2}) \tag{1.1}$$

(2)　パラメタ

DAM .....入力．正規分布の平均 m.
倍精度実数型.

DSD .....入力．正規分布の標準偏差 σ.
倍精度実数型.

IX .....入力．出発値.
初回呼び出しでは，IX に正の値を与え，2 回目以降の呼び出しでは返却値 0 のまま呼び出してください．初回呼び出しで指定する出発値が異なると，異なる乱数列が生成されます．（(3) 使用上の注意 b.注意①参照）.
4 バイト整数型（INTEGER*4）.
出力．0.

DA.....出力．N 個の擬似乱数.
N の大きさをもつ倍精度実数型 1 次元配列.

N.....入力．DA に返却される正規分布擬似乱数の個数.
（(3) 使用上の注意 b.注意②参照）.

DWORK .....作業領域．大きさ NWORK の倍精度実数型 1 次元配列.
本サブルーチンを繰り返し呼び出す場合は, 内容を変更しないでください.
DWORK には，本サブルーチンが再度呼び出される場合に必要な情報が格納されています.
（(3) 使用上の注意 b.注意③参照）.

NWORK .....入力．配列 DWORK の大きさ．NWORK≧1350.

ICON.....出力．コンディションコード.
表 DVRAN4-1 参照.

表 DVRAN4-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	—
30001	NWORK が小さ過ぎた．IX < 0, DSD ≤ 0	処理を打ち切る.
30002	内部エラー	
30003～30008	DWORK が変更されていた．または，初回呼び出しで IX に 0 が与えられた.	
30009	IX が大きすぎる.	
40001～40002	DWORK は上書きされたか，初回呼び出しで IX=0.	

(3)　使用上の注意

a.　使用する副プログラム

SSLII・・・DUF2G3,DUITG3,DURN3B,DURUG3,DUR2G3,DUSKG3,DUSQG3,
DUVRG3,DVRAU4,MGSSL

b.　注意

①　出発値 IX について

確定的な計算（deterministic program）で擬似乱数列を生成するには，何かランダムな入力が必要になります．このために，利用者が出発値 IX を与える必要があります．本サブルーチンの初回呼び出しでは，出発値 IX は正整数である必要があります（例外事項については，⑤を参照）．2 回目以降の呼び出しでは IX には 0 を指定してください．これは同じ乱数列から続いて乱数を求めるためです．プログラミングを容易にするため，本サブルーチンは初回呼び出し後 IX に 0 を返しています.

②　パラメタ N について

本サブルーチンは出発値 IX で定義される無限列から，次の N 個の擬似乱数を返却します．N≦0 の時は擬似乱数を返却しません.

効率良くするためには，利用者は N を十分大きく（例えば N＝100,000）します．それは，サブルーチン呼び出しのオーバーヘッドが減少し，しかもベクトル化されるためです．本サブルーチンを連続的に呼び出す途中で，N が変更される場合は，配列 DA を最大の N の大きさだけ確保する必要があります.

③　作業領域 DWORK について

DWORK は，複数回本サブルーチンを呼び出す場合，次の呼び出しのための情報を格納する作業領域として使用されます．従って，本サブルーチンを呼び出している間は，呼び出し側のプログラムで DWORK の内容を変更してはいけません.

④　パラメタ NWORK について

DWORK(1),..., DWORK(NWORK) は，本サブルーチンによって使用されます．NWORK は，本サブルーチンの各呼び出しで変えないでください．NWORK には少なくとも 1350 以上, ベクトルプロセッサ上では十分大きな数（例えば NWORK＝500,000）を与えると効率的です.

⑤ 同じ乱数の再生成について
DWORK(1),..., DWORK(NWORK) が保存される場合, その DWORK を再使用し, IX＝0 として本サブルーチンを呼び出すことにより,(DWORK が保存された時点からの）同じ乱数列が再生成されます.

⑥ DVRAN4 で使われている Wallace の方法は DVRAN3 で使われている Polar 法に比べて約３倍高速です．ただし，統計的特性が厳しく問われる場合は DVRAN3 の利用を奨めます.

c. 使用例

1,000,000 個の擬似正規乱数を生成し，1 次モーメント（変数値 X_i の度数を F_i としたとき，$\sum X_i F_i$ ），2 次モーメント（変数値 X_i の度数を F_i としたとき，$\sum X_i^2 F_i$ ）を計算します．出発値は 12345 とします.

```
C     ** EXAMPLE **
      PARAMETER  (NRAN  = 1000000)
      PARAMETER  (NSEED = 12345)
      PARAMETER  (NWMAX = 100000)
      PARAMETER  (NBUF  = 120000)
      REAL*8     DA(NBUF)
      REAL*8     DWORK(NWMAX)
      REAL*8     DSUM, DSUM2
      REAL*8     DMEAN, DM2
      IA = NSEED
      PRINT *, ' Seed ', IA
      N = NBUF
      NWORK = NWMAX
      DSUM  = 0.0D0
      DSUM2 = 0.0D0
C NGEN counts down to 0
      NGEN = NRAN
      PRINT *, ' Generating ', NGEN,
     $           ' numbers'
C Generate NRAN numbers ,
C maximum NBUF at a time
      KRPT = (NRAN+NBUF-1)/NBUF
      PRINT *, ' with ', KRPT,
     $           ' call to dvran4'
      DO 20 J = 1, KRPT
      N = MIN0 (NBUF, NGEN)
C First two arguments are mean
C add standard deviation
      CALL DVRAN4 (0.0D0, 1.0D0, IA,
     $ DA, N, DWORK, NWORK, ICON)
```

```
      IF (ICON .NE. 0) THEN
        PRINT *, ' Error Return ', ICON
        STOP
        ENDIF
C Accumulate sum of numbers generated
      DO 10 I = 1, N
      DSUM = DSUM + DA(I)
C Accumulate sum of squares
 10   DSUM2 = DSUM2 + DA(I)*DA(I)
 20   NGEN  = NGEN - N
C Compute sample mean
      DMEAN = DSUM/DFLOAT(NRAN)
      PRINT *, ' First moment ', DMEAN
C Compute sample second moment about 0
      DM2 = DSUM2/DFLOAT(NRAN)
      PRINT *, ' Second moment ', DM2
      STOP
      END
```

(4)　手法概要

DVRAN4 は正規分布する擬似乱数を生成するために，Wallace 法の一種の変形を使用しています．この方法に必要な一様擬似乱数は DVRAU4 を利用して生成されます．

Wallace の方法は“付録　参考文献一覧表”の[43]に記述されています．実現方法の詳細および他の方法との比較は“付録 参考文献一覧表”の[4]および[5]を参照してください．

J11-11-0301 DVRAU4

一様乱数［0,1）の生成（倍精度）
CALL DVRAU4(IX,DA,N,DWORK,NWORK,ICON)

(1)　機能

区間［0, 1）上での一様分布から擬似乱数列を生成します.

(2)　パラメタ

IX .....入力．出発値.
初回呼び出しでは，IX に正の値を与え，2 回目以降の呼び出しでは返却値 0 のまま呼び出してください.
（(3) 使用上の注意 b.注意①参照）.
初回呼び出しで指定する IX の値が異なると, 異なる乱数列が生成されます.
（(4) 手法概要参照）.
4 バイト整数型（INTEGER*4）.
出力．0.

DA.....出力．区間［0, 1）で一様な N 個の擬似乱数.
N の大きさをもつ倍精度実数型 1 次元配列.

N.....入力．DA に返却される一様分布擬似乱数の個数.
（(3) 使用上の注意 b.注意②参照）.

DWORK .....作業領域．少なくとも大きさ NWORK の倍精度実数型 1 次元配列.
本サブルーチンを繰り返し呼び出す場合は, 内容を変更してはなりません.
DWORK には，本サブルーチンが現在の位置から再度呼び出される場合に必要な，全ての現情報が格納されます.
（(3) 使用上の注意 b.注意③参照）.

NWORK .....入力．配列 DWORK の大きさ．NWORK≧388.

ICON.....出力．コンディションコード.
表 DVRAU4-1 参照.

表 DVRAU4-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	―
30001	NWORK が小さ過ぎた.	処理を打ち切る.
30002	IX＜0	
30003～30008	DWORK が変更されていた．または，初回呼び出しで IX に 0 が与えられた.	

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ DUITG3,DURUG3,DUR2G3,DUF2G3,DUSKG3,DUSQG3,DUVRG3,
MGSSL

b.　注意

①　出発値 IX について

確定的な計算（deterministic program）で擬似乱数列を生成する場合は，何かランダムな入力が必要になります．従って，利用者は出発値 IX を与えなければなりません．この出発値はしばしば“たね（*seed*）”と呼ばれます．本サブルーチンの初回呼び出しでは，出発値 IX は正整数である必要があります（例外事項については，⑤を参照）．2 回目以降の呼び出しでは IX には 0 を指定してください．これは同じ乱数列から続いて乱数を求めるためです．プログラミングを容易にするため，本サブルーチンは初回呼び出し後 IX に 0 を返しています．

②　パラメタ N について

本サブルーチンは出発値 IX で定義される無限列から，次の N 個の擬似乱数を返却します．N≦0 の時は擬似乱数を返却しません．

効率良くするためには，利用者は N を十分大きく（例えば N=100,000）します．それは，サブルーチン呼び出しのオーバーヘッドが減少し，しかもベクトル化されるためです．初回呼び出しで指定される N と NWORK の大きさから乱数列生成のパラメタが決定されます．詳細は（4）手法概要を参照してください．本サブルーチンを連続的に呼び出す途中で，N が変更される場合は，配列 DA を最大の N の大きさだけ確保する必要があります．

③　作業領域 DWORK について

DWORK は，複数回本サブルーチンを呼び出す場合，次の呼び出しのための情報を格納する作業領域として使用されます．従って，本サブルーチンを呼び出している間は，呼び出し側のプログラムで DWORK の内容を変更してはいけません．

④　パラメタ NWORK について

DWORK(1),..., DWORK(NWORK) は，本サブルーチンによって使用されます．NWORK は，本サブルーチンの各呼び出しで変えないでください．NWORK には少なくとも 388以上, ベクトルプロセッサ上では十分大きな数（例えばNWORK＝45000）を与えると効率的です．初回呼び出しで指定される N と NWORK の大きさから乱数列生成のパラメタが決定されます．詳細は（4）手法概要を参照してください．

⑤　同じ乱数の再生成について

DWORK(1),..., DWORK(NWORK) が保存される場合, その DWORK を再使用し，IX＝0 として本サブルーチンを呼び出すことにより,（DWORK が保存された時点からの）同じ乱数列が再度生成されます．

c.　使用例

1,000,000 個の擬似一様乱数を生成し, 平均値を計算します. 出発値は 123 とします.

```
C       **EXAMPLE**
        PARAMETER (NRAN = 1000000)
```

```
      PARAMETER (NSEED = 123)
      PARAMETER (NWMAX = 45000)
      PARAMETER (NBUF = 160000)
      REAL*8 DA(NBUF)
      REAL*8 DWORK(NWMAX)
      REAL*8 DSUM, DMEAN, DSIG
      IX = NSEED
      PRINT *, ' SEED ', IX
      N = NBUF
      NWORK = NWMAX
      DSUM = 0.0D0
C NGEN counts down to 0
      NGEN = NRAN
      PRINT *, ' Generating ', NGEN,
     $ ' Numbers'
C Generate NRAN numbers,
C Maximum NBUF at a time
      KRPT = (NRAN+NBUF-1)/NBUF
      PRINT *, ' with ', KRPT,
     $ ' calls to dvrau4'
      DO 20 J = 1, KRPT
      N = MIN0 (NBUF, NGEN)
      CALL DVRAU4 (IX, DA, N,
     $ DWORK, NWORK, ICON)
      IF (ICON .NE. 0) THEN
        PRINT *, ' Error return ', ICON
        STOP
      ENDIF
C Accumulate sun of numbers generated
      DO 10 I = 1, N
   10 DSUM = DSUM + DA(I)
   20 NGEN = NGEN - N
C Compute mean
      DMEAN = DSUM/DFLOAT(NRAN)
      PRINT *, ' Mean ', DMEAN
C Compute deviation from 0.5 normalized
C by expected value 1/sqrt(12*NRAN).
C This should be (approximately) normally
C distributed with mean 0, variance 1.
      DSIG = DMEAN - 0.5D0
      DSIG = DSIG*DSQRT(12.0D0*NRAN)
      PRINT *, ' Norm. deviation ', DSIG
```

```
      STOP
      END
```

(4)　手法概要

本サブルーチンは一般化されたフィボナッチ法（Generalized Fibonacci method）を用いています．擬似乱数列を X(1), X(2) ,..., とするとき，

$X(J) = \alpha * X(J-r) + \beta * X(J-s) (modulo\ 1)$

$J > r > s$

ここで，r と s は固定された正整数で，lags と呼ばれます．αとβは小さな奇数の整数です．初回呼び出し（つまり，IX＞0 としての呼び出し）において，本サブルーチンは 2 を法とした原始三項式（a primitive trinomial(mod 2)）を定義する組（r, s）と，それに対応する線型再帰式（a liner recurrence）を決めます．ここでは，14 個の可能な組（r, s）がありますが，N および NWORK の大きさが許す範囲内で，最大の r をもつ組が選択されます．従って，利用者は，以下の方法が選択できます．

・適度に長い周期を持ち，初期化のオーバーヘッドが小さく，格納域が節約（例えば NWORK＝1000）できる比較的良い生成法

・極めて長い周期を持ち，初期化のオーバーヘッドは大きく，格納域も必要（例えば NWORK＝133000）だが，非常に良い生成法

・組（r, s）の選択方法の正確な詳細はわからない，ある妥協した中間的な方法

本サブルーチンの使用する組（r, s）を表 DVRAU4-2 に示します．原始三項式の表については，“付録　参考文献一覧表”の [20] を参照してください．

表 DVRAU4-2　組（r, s）

r	s	r	s
127	97	4423	2325
258	175	9689	5502
521	353	19937	10095
607	334	23209	13470
1279	861	44497	23463
2281	1252	110503	56784
3217	2641	132049	79500

本サブルーチンは，r≦1000 の時，(α , β) ＝(7, 9) を，r＞1000 の時，(α , β) ＝(1,15) をそれぞれパラメタとして選択します．その理論的根拠は，α＞1 の時，統計的検定（statistical test）において性能が向上するようですが，それは r がより小さい時のみに限定されることなのです．大きな r に対しては，たとえα＝1 でも統計的検定の性能は良く，この選択であれば乱数生成の性能は向上します．

r を 127（最小の NWORK に対して）から 132049(N≧264098,NWORK≧132056) の範囲の数とすると，乱数列の周期は，$W(2^r-1)$ です．因子 W は，語長（word length）に依存して決まります．（Fujitsu VPP シリーズでは，$W=2^{48}$ であり，周期は最低でも 10^{52} 以上です．）

本サブルーチンの初期化では，異なる出発値 IX に対して返却される擬似乱数列が，全周

期列の中で少なくとも2^{60}＞10^{18}の間隔をもって分かれていることを保証しています. 従ってあらゆる実用目的に対して，異なる出発値 IX は異なる擬似乱数列生成をします．手法および実現方法の詳細は，“付録　参考文献一覧表”の [2] および [3] により詳しく記述されています．更に詳しい説明や他の手法との比較については，“付録　参考文献一覧表”の [1] , [11] , [22] および [28] を参照してください.

(5)　一様乱数の検定

表 DVRAU4-3 に，NWORK＝44504(r＝44497,s＝23463) の時に，DVRAU4 が生成した擬似乱数についての統計的仮説検定（testing of statistical hypotheses）の結果を示します.
この表で，χ^2検定の自由度 f は極めて大きく（100 万程度），この場合，
式$\sqrt{2\chi^2}-\sqrt{2f-1}$ は，単位分散を持つ正規分布（a normal deviate with unit variance）に極めて近くなります.

表 DVRAU4-3　χ^2検定結果（n 次元単位超立方体での一様分布）

次元[注1]	サイズ[注2]	res_l[注3]	res_v[注4]	密度[注5]	$\sqrt{2\chi^2}-\sqrt{2f-1}$
1	10^9	5×10^7	50000000	20.00	1.21
1	0.8×10^9	1.25×10^7	12500000	64.00	-0.67
2	10^9	7071	49999041	10.00	-0.10
2	2×10^9	3535	12496225	80.02	-0.37
3	2×10^9	368	49836032	13.38	1.40
3	2×10^9	232	12487168	53.39	-0.96
4	2×10^9	84	49787136	10.04	0.76
4	2×10^9	59	12117361	41.26	-0.38

注 1　次元：単位超立方体の次元
注 2　サイズ：生成される擬似乱数の個数
注 3　res_l：各次元での区間［0,1）を等分割した部分区間の数
注 4　res_v：単位超立方体を等分割した超立方体の数
注 5　密度：小超立方体あたりの乱数点（random point）の平均値

A72-27-0101 VBCSD,DVBCSD

非対称または不定値のスパース実行列の連立 1 次方程式 （BICGSTAB(*l*)法，対角形式格納法）
CALL VBCSD(A,K,NDIAG,N,NOFST,B,ITMAX,EPS,IGUSS,L,X,ITER,VW,ICON)

(1)　機能

n×*n* の非対称／不定値なスパース行列を係数行列とする連立 1 次方程式を *l* 次安定化双対共役勾配法（BICGSTAB(*l*): Bi-Conjugate Gradient Stabilized(*l*) method）で解きます．

$$\boldsymbol{Ax} = \boldsymbol{b}$$

n×*n* の係数行列は，対角形式の格納法で格納します．ベクトル $\boldsymbol{b}$ および $\boldsymbol{x}$ は *n* 次元ベクトルです．

反復解法の収束および利用指針に関しては，“第 I 部 概説 第 4 章 連立一次方程式の反復解法と収束性”を参照してください．

(2)　パラメタ

A.....入力．係数行列の非零要素を格納します．
A（K,NDIAG）なる 2 次元配列．A（1:N,NDIAG）に係数行列 $\boldsymbol{A}$ を対角形式で格納します．
対角形式の格納方法については，“第 I 部 概説 3.2.1.1 一般スパース行列の格納方法 b.一般スパース行列の対角形式の格納方法”を参照してください．

K.....入力．配列 A の整合寸法．

NDIAG.....入力．係数行列 $\boldsymbol{A}$ の非零要素を含む対角ベクトル列の総数．
配列 A の 2 次元目の大きさ．

N.....入力．行列 $\boldsymbol{A}$ の次数 *n*．

NOFST.....入力．A に格納される対角ベクトルに対応した主対角ベクトルからの距離を格納します．上対角ベクトル列は正，下対角ベクトル列は負の値で表します．
NOFST（NDIAG）なる 1 次元配列．

B.....入力．大きさ *n* の 1 次元配列．連立 1 次方程式の右辺の定数ベクトルを格納します．

ITMAX .....入力．BICGSTAB(*l*)法の反復回数の上限値（＞0）．2000 程度で十分です．

EPS.....入力．収束判定に用いられる判定値．
EPS が 0.0 以下のとき，EPS は倍精度ルーチンでは 10^{-6} が，単精度ルーチンでは 10^{-4} が設定されます．
（(3)使用上の注意 b.注意①参照）．

IGUSS.....入力．配列 X に指定された解ベクトルの近似値から反復計算を行うかを指定する制御情報．
0 のとき　解ベクトルの近似値を指定しません．
0 以外のとき 配列 X に指定された解ベクトルの近似値から反復計算を開始します．

L.....入力．BICGSTAB(*l*)法で安定化を行うときの安定化のステップ数 L (1≦L≦8）．ほとんどの場合１か２で十分です．
（(3)使用上の注意 b.注意②参照）．

X.....入力．大きさ *n* の 1 次元配列．解ベクトルの近似値を指定することができます．
出力．解ベクトルが格納されます．

ITER.....出力．BICGSTAB(*l*)法での実際の反復回数．

VW.....作業域. 大きさ, K×(4＋2×L)＋N＋NBANDL＋NBANDR なる 1 次元配列．
NBANDL は下バンド幅，NBANDR は上バンド幅の大きさ．
バンド幅や行列の次数が定数でない場合は，K×(4＋2×L)＋3×K の大きさの配列をとっておけば十分です．

ICON.....出力．コンディションコード．
表 VBCSD-1 参照．

表 VBCSD-1　コンディションコード

コード	意　味	処理内容
0	エラーなし．	ー
20000	break down を起こした．	処理を打切る．
20001	反復回数の上限に達した．	処理を打ち切る．配列 X には，そのときまでに得られている近似値を出力するが，精度は保証できない．
30000	K＜1，N＜1，K＜N，L＜1， L＞8，NDIAG＜1，K＜NDIAG， または ITMAX≦0	処理を打ち切る．
32001	｜NOFST（I）｜＞N－1	

(3)　使用上の注意

a.　使用する副プログラム

SSLII・・・AMACH, UBCRL UBCSD UBGRS UQITB URELT URIPA URITI
URITT URMVD URSTE USVCN USVCP USVN2 MGSSL UMGSL

b.　注意

①　収束判定

BICGSTAB(*l*)法は，残差のユーグリッドノルムが最初の残差のユーグリッドノルムと EPS の積以下になったとき収束したと見なします．正確な解と求められた近似解の誤差はほぼ行列 ***A*** の条件数と EPS の積に等しくなります．
収束判定に使われる残差は，反復法におけるベクトルとして順次計算されるもので，真の残差の値とは多少異なります．

② L の値について
L の値は，1 から 8 までの整数値を与えます．L＝1 のとき，BICGSTAB 法と同じアルゴリズムになります．L の値が大きいと，反復一回あたりのコストは増加しますが，反復回数が減り，よい収束が得られる場合があります．

③ 対角形式を使う上での注意
係数行列 ***A*** の外側の対角ベクトルの要素は零を設定する必要があります．
対角ベクトル列を配列 A に格納する順序に制限は特にありません．

④ 対角要素によるスケーリング
本ルーチン呼び出し前に，係数行列の対角要素が 1 になるように方程式をスケーリングしておくことで，収束が良くなる場合があります．

⑤ 本方法では，反復計算が係数行列や初期ベクトルの特性のため続けられない場合（break down）があります．この場合，出力解を初期値として再呼び出しすることで収束する場合があります．または，他の手法による解法ルーチンをお使い下さい．

c.　使用例

領域［0,1］×［0,1］×［0,1］でディリクレ境界条件（境界で関数値は零）のもとで，“第 I 部 概説 3.2.2 偏微分演算子の離散化とその格納例”に記述された偏微分演算子を離散化して得られる係数行列を持つ連立 1 次方程式を解きます．
INIT_MAT_DIAG は，“第 I 部 概説 3.2.2 偏微分演算子の離散化とその格納例”を参照してください．GET_BANDWIDTH_DIAG はバンド幅見積りのルーチン，INIT_SOL は，求めるべき解ベクトルを乱数で生成するルーチンです．

```
c     **EXAMPLE**
      PROGRAM TEST_ITER_SOLVERS
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER MACH
      PARAMETER (MACH = 0)
      PARAMETER (K = 10000)
      PARAMETER (NX = 20,NY = 20,NZ = 20,N = NX*NY*NZ)
      PARAMETER (NDIAG = 7, LEN = N+400+400)
      PARAMETER (L = 4)
      PARAMETER (NVW = (4+2*L)*K+LEN)
      DOUBLE PRECISION A(K,NDIAG),X(N),B(N),SOLEX(N)
      INTEGER NOFST(NDIAG)
      DOUBLE PRECISION VW(NVW)
C
      CALL INIT_SOL(SOLEX,N,1D0,MACH)
      PRINT*,'EXPECTED SOLUSIONS'
      PRINT*,'X(1) = ',SOLEX(1),' X(N) = ',SOLEX(N)
C
      PRINT *
      PRINT *,'    BiCGstab(l) METHOD'
```

```
      PRINT *,'    DIAGONAL FORMAT'
C
      VA1 = 3D0
      VA2 = 1D0/3D0
      VA3 = 5D0
      VC = 1.0
      XL = 1.0
      YL = 1.0
      ZL = 1.0
C
      CALL INIT_MAT_DIAG(VA1,VA2,VA3,VC,A,NOFST
     &            ,NX,NY,NZ,XL,YL,ZL,NDIAG,N,K)
      CALL GET_BANDWIDTH_DIAG(NOFST,NDIAG,NBANDL,NBANDR)
          DO 110 I = 1,N
              VW(I+NBANDL) = SOLEX(I)
 110      CONTINUE
      CALL DVMVSD(A,K,NDIAG,N,NOFST,NBANDL,VW,B,ICON)
      PRINT*,'DVMVSD ICON= ',ICON
C
      EPS = 1D-10
      IGUSS = 0
      ITMAX = 2000
      CALL DVBCSD(A,K,NDIAG,N,NOFST,B,ITMAX
     &             ,EPS,IGUSS,L,X,ITER,VW,ICON)
C
      PRINT* ,'ITER = ',ITER
      PRINT* ,'DVBCSD ICON = ',ICON
      PRINT*, 'COMPUTED VALUES'
      PRINT*, 'X(1) = ',X(1),' X(N) = ',X(N)
      STOP
      END
```

(4)　手法概要

BICG アルゴリズムについては“付録　参考文献一覧表”の[38]に記述されています. BICGSTAB(*l*)法は BICGSTAB 法の変形です. “付録　参考文献一覧表”の[42]および[16]を参照してください.

A72-28-0101 VBCSE,DVBCSE

非対称または不定値のスパース実行列の連立 1 次方程式 （BICGSTAB(*l*)法，ELLPACK 形式格納法）
CALL VBCSE(A,K,IWIDT,N,ICOL,B,ITMAX,EPS,IGUSS,L,X,ITER,VW,ICON)

(1)　機能

n×*n* の非対称／不定値なスパース行列を係数行列とする連立 1 次方程式を *l* 次安定化双対共役勾配法（BICGSTAB(*l*): Bi-Conjugate Gradient Stabilized(*l*) method）で解きます．

$$\boldsymbol{Ax} = \boldsymbol{b}$$

n×*n* の係数行列は，ELLPACK 形式の格納法で格納します．ベクトル ***b*** および ***x*** は *n* 次元ベクトルです．

反復解法の収束および利用指針に関しては，“第 I 部 概説 第 4 章 連立一次方程式の反復解法と収束性”を参照してください．

(2)　パラメタ

A.....入力．係数行列の非零要素を格納します．
　　A（K,IWIDT）なる 2 次元配列．
　　ELLPACK 形式の格納方法については，“第 I 部 概説 3.2.1.1 一般スパース行列の格納方法”を参照してください．

K.....入力．A および ICOL の整合寸法（≧*n*）．

IWIDT.....入力．係数行列 ***A*** の非零要素の行ベクトル方向の最大個数．
　　A および ICOL の 2 次元目の大きさ．

N.....入力．行列 A の次数 *n*．

ICOL .....入力．ELLPACK 形式で使用される列指標で，A の対応する要素がいずれの列ベクトルに属すかを示します．
　　ICOL（K,IWIDT）なる 2 次元配列．

B.....入力．大きさ *n* の 1 次元配列．連立 1 次方程式の右辺の定数ベクトルを B に格納します．

ITMAX .....入力．BICGSTAB(*l*)法の反復回数の上限値（＞0）．2000 程度で十分です．

EPS.....入力．収束判定に用いられる判定値．
　　EPS が 0.0 以下のとき，EPS は倍精度ルーチンでは 10^{-6} が，単精度ルーチンでは 10^{-4} が設定されます．
　　（(3) 使用上の注意 b.注意①参照）．

IGUSS.....入力．配列 X に指定された解ベクトルの近似値から反復計算を開始するかを示す制御情報．
　　0 のとき　解ベクトルの近似値を指定しません．

0 以外のとき 配列 X に指定された解ベクトルの近似値から反復計算を開始します.

L.....入力. BICGSTAB(*l*)法で安定化を行うときの安定化のステップ数 L (1≦L≦8). ほとんどの場合, 1 か 2 で十分です.
（(3)使用上の注意 b.注意②参照）.

X.....入力. 大きさ *n* の 1 次元配列. 解ベクトルの近似値を指定することができます.
出力. 解ベクトルが格納されます.

ITER.....出力. BICGSTAB(*l*)法での実際の反復回数.

VW.....作業域. 大きさ, K×(4+2×L)なる 1 次元配列.

ICON.....出力. コンディションコード.
表 VBCSE-1 参照.

表 VBCSE-1　コンディションコード

コード	意　味	処理内容
0	エラーなし.	ー
20000	break down を起こした.	処理を打切る.
20001	反復回数の上限に達した.	処理を打ち切る. 配列 X には, そのときまでに得られている近似値を出力するが, 精度は保証できない.
30000	K ＜ 1, N ＜ 1, K ＜ N, L ＜ 1, L ＞ 8, IWIDT ＜ 1, K ＜ IWIDT, または ITMAX ≦ 0	処理を打ち切る.

(3)　使用上の注意

a.　使用する副プログラム

SSLII・・・AMACH, UBCRL, UBCSE, UBGRS, UQITB, URELT,
URIPA, URITI, URITT, URMVE, URSTE, USVCN, USVCP
USVN2, MGSSL, UMGSL

b.　注意

①　収束判定

BICGSTAB(*l*)法は, 残差のユーグリッドノルムが最初の残差のユーグリッドノルムと EPS の積以下になったとき収束したと見なします.
正確な解と求められた近似解の誤差はほぼ行列 ***A*** の条件数と EPS の積に等しくなります.
収束判定に使われる残差は, 反復法におけるベクトルとして順次計算されるもので, 真の残差の値とは多少異なります.

②　L の値について

L の値は, 1 から 8 までの整数値を与えます. L＝1 のとき, BICGSTAB 法と同じアルゴリズムになります. L の値が大きいと, 反復一回あたりのコストは増加しますが, 反復回数が減り, よい収束が得られる場合があります.

③　対角要素によるスケーリング

本ルーチン呼び出し前に, 係数行列の対角要素が 1 になるように方程式をスケーリングしておくことで，収束が良くなる場合があります.

④ 本方法では, 反復計算が係数行列や初期ベクトルの特性のため続けられない場合（break down）があります．この場合，出力解を初期値として再呼び出しすることで収束する場合があります．または，他の手法による解法ルーチンをお使い下さい.

c.　使用例

領域［0,1］×［0,1］×［0,1］でディリクレ境界条件（境界で関数値は零）のもとで，“第 I 部 概説 3.2.2 偏微分演算子の離散化とその格納例”に記述された偏微分演算子を離散化して得られる係数行列を持つ連立 1 次方程式を解きます．INIT_MAT_ELL は，“第 I 部 概説 3.2.2 偏微分演算子の離散化とその格納例”を参照してください. INIT_SOL は，求めるべき解ベクトルを乱数で生成するルーチンです.

```
C     **EXAMPLE**
      PROGRAM TEST_ITER_SOLVERS
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER MACH
      PARAMETER (MACH = 0)
      PARAMETER (K = 10000)
      PARAMETER (NX = 20,NY = 20,NZ = 20,N = NX*NY*NZ)
      PARAMETER (IWIDT = 7, L = 4)
      PARAMETER (NVW = (4+2*L)*K)
      DOUBLE PRECISION A(K,IWIDT),X(N),B(N),VW(NVW),SOLEX(N)
      INTEGER ICOL(K,IWIDT)
C
      CALL INIT_SOL(SOLEX,N,1D0,MACH)
      PRINT*,'EXPECTED SOLUTION'
      PRINT*,'X(1) = ',SOLEX(1),' X(N) = ',SOLEX(N)
C
      PRINT *
      PRINT *,'    BiCGstab(l) METHOD'
      PRINT *,'    ELLPACK FORMAT'
C
      AVI = 3D0
      AV2 = 1D0/3D0
      AV3 = 5D0
      VC = 1.0
      XL = 1.0
      YL = 1.0
      ZL = 1.0
C
```

```
      CALL INIT_MAT_ELL(VA1,VA2,VA3,VC,A,ICOL,
     &          NX,NY,NZ,XL,YL,ZL,IWIDT,N,K)
      CALL DVMVSE(A,K,IWIDT,N,ICOL,SOLEX,B,ICON)
      PRINT*,'DVMVSE ICON = ',ICON
C
      EPS = 1D-10
      IGUSS = 0
      ITMAX = 2000
      CALL DVBCSE(A,K,IWIDT,N,ICOL,B,ITMAX,
     &            EPS,IGUSS,L,X,ITER,VW,ICON)
C
      PRINT*,'DVBCSE ICON = ',ICON
      PRINT*,'COMPUTED VALUE'
      PRINT*,'X(1) = ',X(1),' X(N) = ',X(N)
      STOP
      END
```

(4)　手法概要

BICG アルゴリズムについては“付録　参考文献一覧表”の[38]に記述されています.
BICGSTAB(*l*)法は BICGSTAB 法の変形です. “付録　参考文献一覧表”の[42]および[16]を参照してください.

A53-31-0102 VBLDL,DVBLDL

正値対称バンド行列の LDL^T 分解（変形コレスキー分解）
CALL VBLDL(A,N,NH,EPSZ,ICON)

(1)　機能

n×*n*，上，下バンド幅 *h* の正値対称バンド行列 ***A*** を変形コレスキー法によって LDL^T 分解します．

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^T$$

ただし，***L*** は下バンド幅 *h* の単位下バンド行列，***D*** は対角行列です．

n＞*h*≧0 を満たさなければなりません．

本ルーチンでは，ベクトル計算機の性能を引き出す上で適した，列ベクトル順に格納する方法を採用しています．

(2)　パラメタ

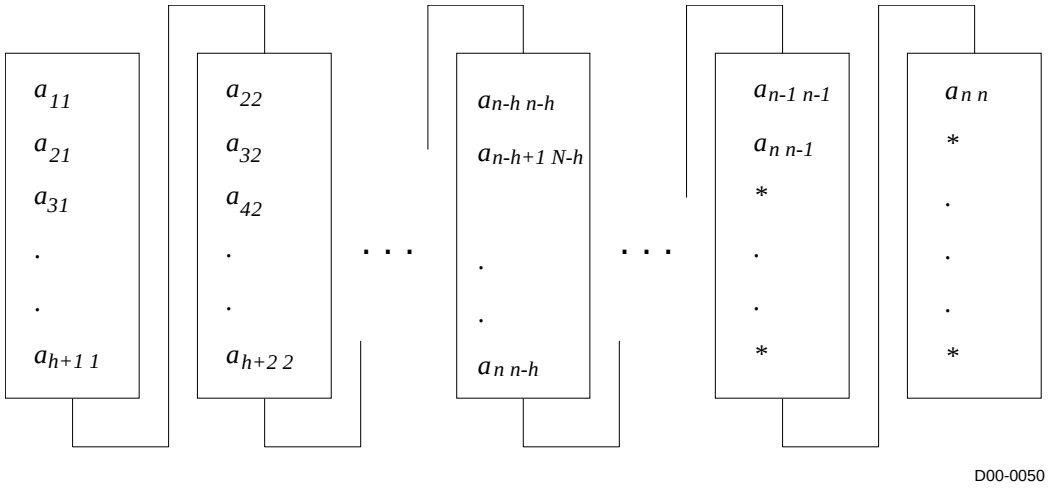
A.....入力．大きさ (*h*＋1)×*n* の 1 次元配列．
　　係数行列 *A* の対角要素と下バンド行列を格納します．
　　行列 *A* の格納法については，図 VBLDL-1 参照．
　　出力．LDL^T 分解された *D* および *L* が格納されます．
　　行列 *L* および *D* の格納法については，図 VBLDL-2 参照．

N.....入力．行列 *A* の次数 *n*．

NH.....入力．下バンド幅 *h*．

EPSZ .....入力．ピボットの相対零判定値（≧0.0）．
　　0.0 のときは標準値が採用されます．
　　（(3) 使用上の注意 b.注意①参照）．

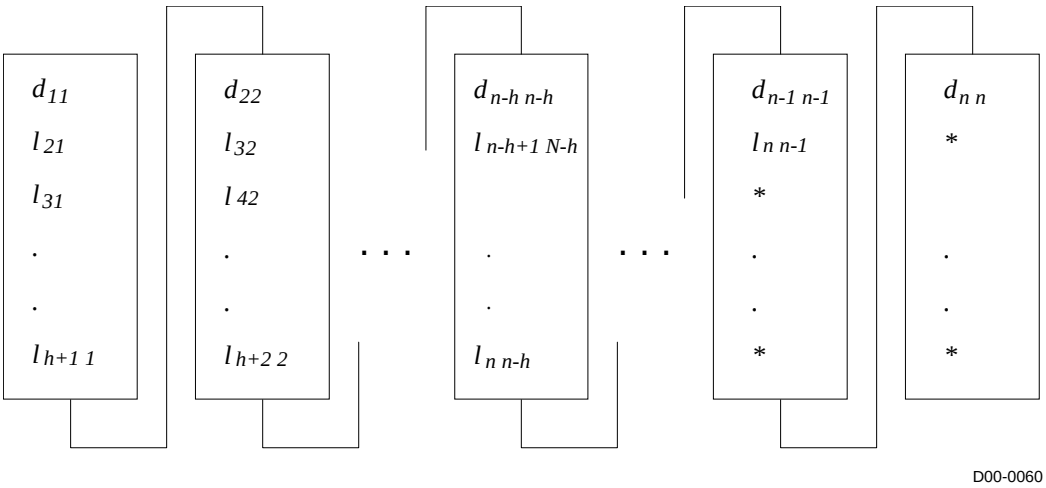
ICON.....出力．コンディションコード．
　　表 VBLDL-1 参照．



＊：不定値を示します.

図 VBLDL-1　配列 A への行列 *A* の格納方法

A の下バンド行列の *i* 列ベクトルを
A((*h*＋1)×(*i*－1)＋*j*－*i*＋1)＝a_{ji}
のように格納します. *j*＝*i*, ..., *i*＋*h*, *i*＝1, ..., *n*



＊：不定値を示します.

図 VBLDL-2　配列 A への行列 *L* および行列 *D* の格納方法

d_{ii} が A((*h*＋1)×(*i*－1)＋1) に格納されます.
l_{ji} が A((*h*＋1)×(*i*－1)＋*j*－*i*＋1) に格納されます.
j = *i* + 1, ..., *i* + *h*, *i* = 1, ..., *n*

表 VBLDL-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	—
10000	ピボットが負となった．行列 $\boldsymbol{A}$ は正値ではない.	処理は続行する.
20000	ピボットが相対的に零となった．行列 $\boldsymbol{A}$ は非正則の可能性が強い.	処理を打ち切る.
30000	NH＜0, NH≧N または, EPSZ＜0.0 であった.	

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ AMACH,MGSSL

b.　注意

① 本サブルーチンでは, EPSZ よりピボットの値が小さいとき相対的に零と見なし, ICON＝20000 として処理を打ち切ります.
EPSZ の標準値は丸め誤差の単位を u としたとき，EPSZ＝16×u です.

② 分解過程でピボットが負となった場合，係数行列は正値ではない．本サブルーチンでは，このときの処理を続行しますが ICON＝10000 とします.

③ 分解結果の行列 $\boldsymbol{L}$ の要素は配列 A 上に, 図 VBLDL-2 で示すとおり格納されます. よって行列式は，n 個の対角要素，A((h＋1)×(i－1)＋1)，i＝1, ..., n の積を計算することで得られます.

c.　使用例

n＝256×256,h＝256 の正値対称バンド行列を LDL^T 分解して，行列式を計算します.

```
C     **EXAMPLE**
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER(NH=128)
      PARAMETER(N=128*128)
      DIMENSION A((NH+1)*N),C(NH+1,N)
      EQUIVALENCE(A,C)
C
C     Zero clear
C
      DO 10 I=1,N*(NH+1)
      A(I)=0.0
   10 CONTINUE
C
C     Coefficient Matrix is built
C     b = A*y , where y=(1,1,....,1)
```

```
C
      DO 20 I=1,N
      C(1,I)=1.0
      IF(I+NH.LE.N)THEN
      C(NH+1,I)=-0.25
      ENDIF
      IF(I+1.LE.N.AND.MOD(I,NH).NE.0)THEN
      C(2,I)=-0.25
      ENDIF
   20 CONTINUE
C
C     LDL^T decomposition
C
      EPSZ=0.0D0
      CALL DVBLDL(A,N,NH,EPSZ,ICON)
      PRINT*,'ICON=',ICON
      IF(ICON.NE.0)STOP
C
      DET=1.0D0
      DO 30 I=1,N
      DET=DET*C(1,I)
   30 CONTINUE
C
      PRINT*,'DETERMINANT=',DET
      STOP
      END
```

(4)　手法概要

外積形式の修正コレスキー分解（“付録 参考文献一覧表”の [32] 参照）で LDL^T 分解を行っています.

A53-31-0202 VBLDX,DVBLDX

LDL^{T}分解された正値対称バンド行列の連立 1 次方程式
CALL VBLDX(B,FA,N,NH,ICON)

(1)　機能

LDL^{T}分解された正値対称バンド行列を係数行列に持つ連立 1 次方程式

$$\boldsymbol{LDL}^T\boldsymbol{x} = \boldsymbol{b} \tag{1.1}$$

を解きます．ただし $\boldsymbol{L}$，$\boldsymbol{D}$ はそれぞれ $n\times n$，下バンド幅 h の単位下バンド行列，対角行列，$\boldsymbol{b}$ は n 次元の実定数ベクトル，$\boldsymbol{x}$ は n 次元の解ベクトルです．

$n > h \geqq 0$ を満たさなければなりません．

(2)　パラメタ

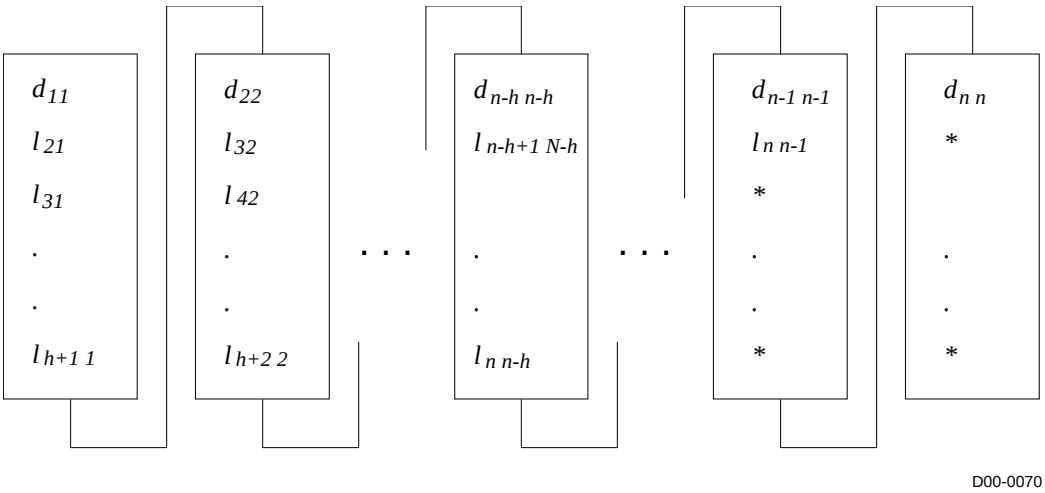
B.....入力．定数ベクトル b.
　　出力．解ベクトル x.
　　大きさ n の 1 次元配列.

FA .....入力．大きさ $(h+1)\times n$ の 1 次元配列．LDL^{T}分解された $\boldsymbol{L}$ および $\boldsymbol{D}$ の格納方法については，図 VBLDX-1 参照.

N.....入力．行列 $\boldsymbol{A}$ の次数 n.

NH.....入力．下バンド幅 h.

ICON.....出力．コンディションコード.
　　表 VBLDX-1 参照.



＊：不定値を示します.

図 VBLDX-1　配列 FA への行列 L および行列 D の格納方法

d_{ii} を FA($(h+1)\times(i-1)+1$) に格納します.

l_{ji} を FA($(h+1)\times(i-1)+j-i+1$) に格納します.

$j = i + 1, ..., i + h, i = 1, ..., n$

表 VBLDX-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	―
10000	係数行列が正値でなかった.	処理は続行する.
30000	NH＜0,NH≧N であった.	処理を打ち切る.

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ UBLTS,UBUTS,MGSSL

b.　注意

①　本ルーチンをサブルーチン VBLDL に続けて呼び出すことにより, 連立 1 次方程式を解くことができます. しかし, 通常はサブルーチン VLSBX を 1 回呼び出せば, 一度に解を求めることができます.

c.　使用例

バンド幅 h＝256,n＝256×256 の正値対称行列を LDL^T 分解し $Ax=b$ を解きます.

```
C     **EXAMPLE**
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER(NH=128)
      PARAMETER(N=128*128)
      DIMENSION A((NH+1)*N),B(N),C(NH+1,N)
      EQUIVALENCE(A,C)
C
C     Zero clear
C
      DO 10 I=1,N*(NH+1)
      A(I)=0.0
   10 CONTINUE
C
      DO 15 I=1,N
      B(I)=0.0
   15 CONTINUE
C
C     Coefficient Matrix is built
C     b = A*y , where y=(1,1,....,1)
C
      DO 20 I=1,N
      C(1,I)=4.0
      B(I)=B(I)+4.0
C
      IF(I+NH.LE.N)THEN
      C(NH+1,I)=-1.0
```

```
      B(I+NH)=B(I+NH)-1.0
      B(I)=B(I)-1.0
      ENDIF
C
      IF(I+1.LE.N.AND.MOD(I,NH).NE.0)THEN
      C(2,I)=-1.0
      B(I+1)=B(I+1)-1.0
      B(I)=B(I)-1.0
      ENDIF
   20 CONTINUE
C
C     Solve Symmetric Positive Definite linear equation
C
      EPSZ=0.0D0
      CALL DVBLDL(A,N,NH,EPSZ,ICON)
      PRINT*,'VBLDL ICON=',ICON
      IF(ICON.NE.0)STOP
      CALL DVBLDX(B,A,N,NH,ICON)
      PRINT*,'VBLDX ICON=',ICON
      IF(ICON.NE.0)STOP
C
      PRINT*,'B(1)= ',B(1)
      PRINT*,'B(N)= ',B(N)
      STOP
      END
```

(4)　手法概要

前進代入および後退代入により解を求めます.

A53-11-0102 VBLU,DVBLU

実バンド行列の LU 分解（ガウスの消去法）
CALL VBLU(A,N,NH1,NH2,EPSZ,IS,IP,VW,ICON)

(1)　機能

$n \times n$，下バンド巾 h_1，上バンド巾 h_2 のバンド行列をガウスの消去法により LU 分解します．

$$\boldsymbol{PA} = \boldsymbol{LU}$$

$\boldsymbol{P}$ は部分ピボッティングによる行の入替えを行う置換行列，$\boldsymbol{L}$ は単位下バンド行列，$\boldsymbol{U}$ は上バンド行列です．

$n > h_1 \geqq 0$, $n > h_2 \geqq 0$ を満たす必要があります．

本ルーチンでは，ベクトル計算機の性能を引き出すために適したバンド行列の格納法を採用しています．

(2)　パラメタ

A.....入力．バンド係数行列 $\boldsymbol{A}$ を格納する，大きさ（$2 \times h_1 + h_2 + 1) \times n$ の大きさの 1 次元配列．
行列 $\boldsymbol{A}$ の格納法については，図 VBLU-1 参照．
出力．LU 分解された $\boldsymbol{L}$ および $\boldsymbol{U}$ が格納されます．
行列 $\boldsymbol{L}$ および $\boldsymbol{U}$ の格納方法については，図 VBLU-2 参照．

N.....入力．行列 $\boldsymbol{A}$ の次数．

NH1.....入力．行列 $\boldsymbol{A}$ の下バンド巾 h_1．

NH2.....入力．行列 $\boldsymbol{A}$ の上バンド巾 h_2．

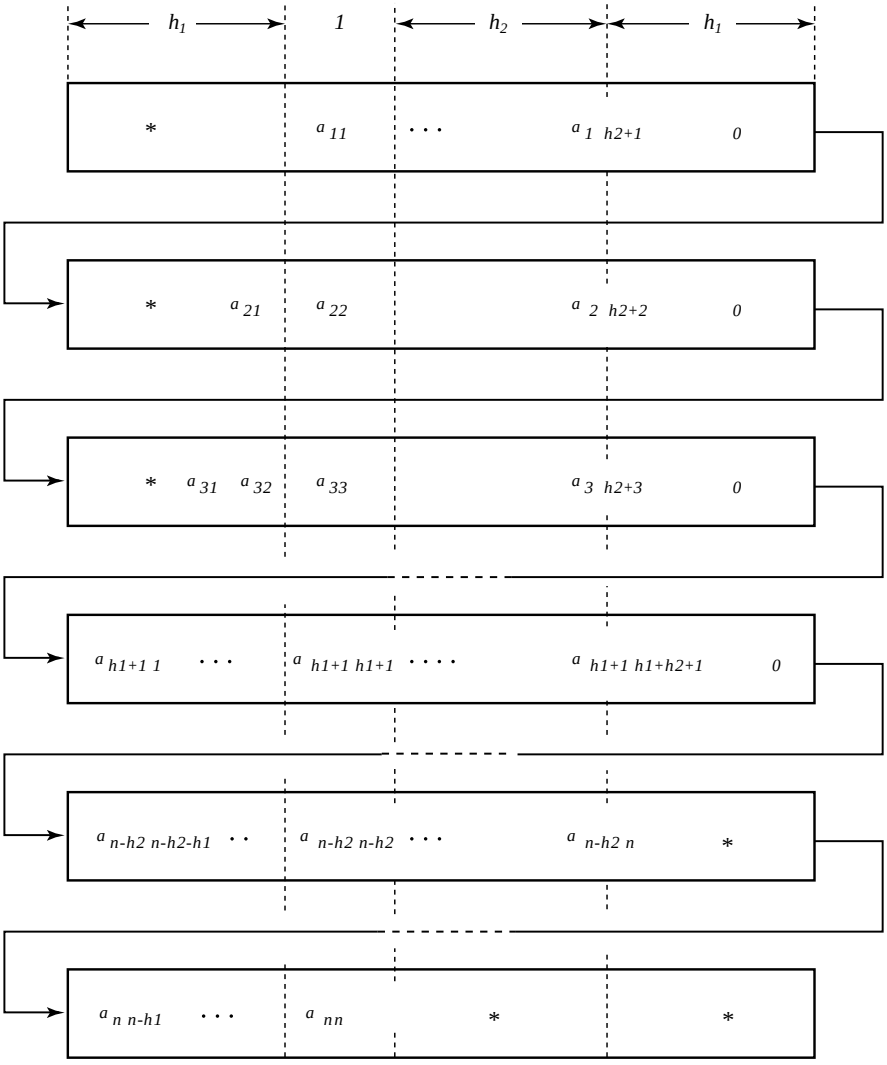
EPSZ.....入力．ピボットの相対零判定値（≧0.0）．0.0 のときは，標準値が設定されます．

IS.....出力．行列 $\boldsymbol{A}$ の行列式を求めるための情報．
（(3) 使用上の注意 b.注意②参照）．

IP.....出力. 部分ピボッティングによる行の入替えの履歴を示すトランスポジションベクトル．大きさ n の 1 次元配列．

VW.....作業領域．大きさ n の 1 次元配列．

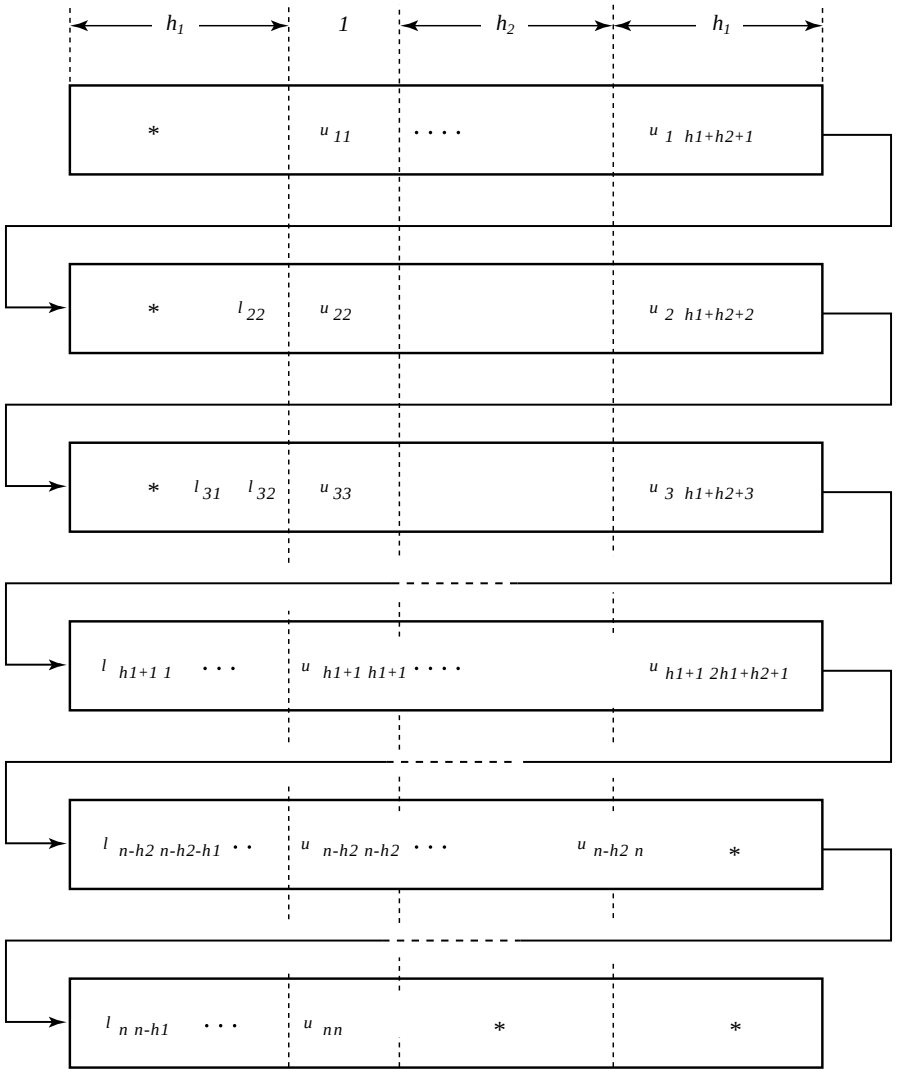
ICON.....出力．コンディションコード．
表 VBLU-1 参照



＊ ： 不定値を示します。

図 VBLU-1　配列 A におけるバンド行列の格納方法

係数行列 ***A*** の i 番目の行ベクトルは，A($(2\times h_1+h_2+1)\times(i-1)+1 : (2\times h_1+h_2+1)\times i$)に連続に格納します．対角要素 a_{ii} が，A($(2\times h_1+h_2+1)\times(i-1)+h_1+1$)に格納されるようにします．格納するとき，バンド部分の外側の係数行列の要素にはゼロを設定します．



＊： 不定値を示します。

図 VBLU-2　配列 A における *L* および *U* の格納方法

行列 *L* の対角要素を除いた第 *i* 行ベクトルが，A((2×h_1＋h_2＋1)×(*i*－1)＋1 : (2×h_1＋h_2＋1)×(*i*－1)＋h_1) に格納されます．また行列 ***U*** の *i* 行ベクトルが A((2×h_1＋h_2＋1)×(*i*－1)＋h_1＋1 : (2×h_1＋h_2＋1)×*i*) に対角要素から連続的に格納されます．

表 VBLU-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし．	—
20000	行列 *A* のある行の要素がすべて零であったか，又はピボットが相対的に零となった．行列 *A* は非正則の可能性が強い．	処理を打ち切る．
30000	N≦NH1,N≦NH2,NH1＜0,NH2＜0 または EPSZ＜0.0 であった．	

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ AMACH,MGSSL

b.　注意

① 本サブルーチンでは, EPSZ よりピボットの値が小さいとき相対的に零と見なし, ICON＝20000 として処理を打ち切ります.
EPSZ の標準値は丸め誤差の単位を u としたとき，EPSZ＝16×u です.

② 行列 U の要素は配列 A 上に，図 VBLU-2 で示すとおり格納されます．よって行列式は，n 個の対角要素，即ち，
A((2×h_1＋h_2＋1)×(i－1)＋h_1＋1)，i＝1, ..., n の積と IS の値を掛け合わせて得られます.

③ 本サブルーチンでは，部分ピボッティングに伴い，行列 A の内容を実際交換しています．すなわち，分解の第 J 段階目（J＝1,2, ..., n－1）において第 I 行（I≧J）がピボット行として選択された場合には, 配列 A の第 I 行と第 J 行の内容が交換されます．そして，その履歴を示すために IP（J）に I が格納されます.

④ 本サブルーチンでは，バンド行列の特性を考慮して，データ格納領域を節約できるように，バンド行列を格納していますが，バンド幅の大きさによっては, VALU よりデータ格納領域を多く必要とする場合があります. そのときは, VALU を用いた方がデータ格納領域を節約できます.
本サブルーチンの特性を生かせるのは，n＞2×h_1＋h_2＋1 のときです.

c.　使用例

h_1＝h_2＝160,n＝160×160 の非対称バンド行列の行列式を計算します.

```
C     **EXAMPLE**
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER(NH=80)
      PARAMETER(NH1=NH)
      PARAMETER(NH2=NH)
      PARAMETER(N=NH*NH)
      PARAMETER(ALPHA=0.5/(NH1+1)/4,BETA=-ALPHA)
      DIMENSION A((2*NH1+NH2+1)*2*N)
      DIMENSION C(2*NH1+NH2+1,N),IP(N),VW(N)
      EQUIVALENCE(A,C)
C
C     Zero clear
C
      DO 10 I=1,N*(3*NH+1)
      A(I)=0.0
   10 CONTINUE
C
C     Coefficient Matrix is built
C
```

```
      DO 20 I=1,N
      C(NH1+1,I)=1.0
      IF(I.GT.NH)THEN
      C(1,I)=-0.25+ALPHA
      ENDIF
      IF(I+NH.LE.N)THEN
      C(1+NH1+NH2,I)=-0.25+BETA
      ENDIF
      IF(I.GT.1.AND.MOD(I-1,NH).NE.0)THEN
      C(NH1,I)=-0.25+ALPHA
      ENDIF
      IF(I+1.LE.N.AND.MOD(I,NH).NE.0)THEN
      C(NH1+2,I)=-0.25+BETA
      ENDIF
   20 CONTINUE
C
C     LU decomposition
C
      EPSZ=0.0D0
      ICON=0
      CALL DVBLU(A,N,NH1,NH2,EPSZ,IS,IP,VW,ICON)
      PRINT*,'ICON= ',ICON
      IF(ICON.NE.0)STOP
C
      DET=IS
      DO 30 I=1,N
      DET=DET*C(NH1+1,I)
   30 CONTINUE
C
      PRINT*,'DETERMINANT=',DET
      STOP
      END
```

(4)　手法概要

外積形式の LU 分解（“付録　参考文献一覧表”の [14] 参照）で LU 分解を行います.

A53-11-0202 VBLUX,DVBLUX

LU 分解された実バンド行列の連立 1 次方程式
CALL VBLUX(B,FA,N,NH1,NH2,IP,ICON)

(1)　機能

$n \times n$, 下バンド巾 h_1, 上バンド巾 h_2 のバンド行列をガウスの消去法により LU 分解した結果

$$\boldsymbol{PA} = \boldsymbol{LU}$$

より，前進代入および後退代入で連立 1 次方程式を解きます.

$$\boldsymbol{Ax} = \boldsymbol{b}$$

$\boldsymbol{P}$ は部分ピボッティングによる行の入替えを行う置換行列，$\boldsymbol{L}$ は単位下バンド行列，$\boldsymbol{U}$ は上バンド行列です.
$n > h_1 \geqq 0,\ n > h_2 \geqq 0$ を満たす必要があります.

(2)　パラメタ

B.....入力．定数ベクトル $\boldsymbol{b}$.
　　出力．解ベクトル $\boldsymbol{x}$.
　　大きさ n の 1 次元配列.

FA .....入力．LU 分解された $\boldsymbol{L}$ および $\boldsymbol{U}$ を格納します.
　　大きさ $(2 \times h_1 + h_2 + 1) \times n$ の 1 次元配列.
　　行列 $\boldsymbol{L}$ および $\boldsymbol{U}$ の格納法については，図 VBLUX-1 参照.

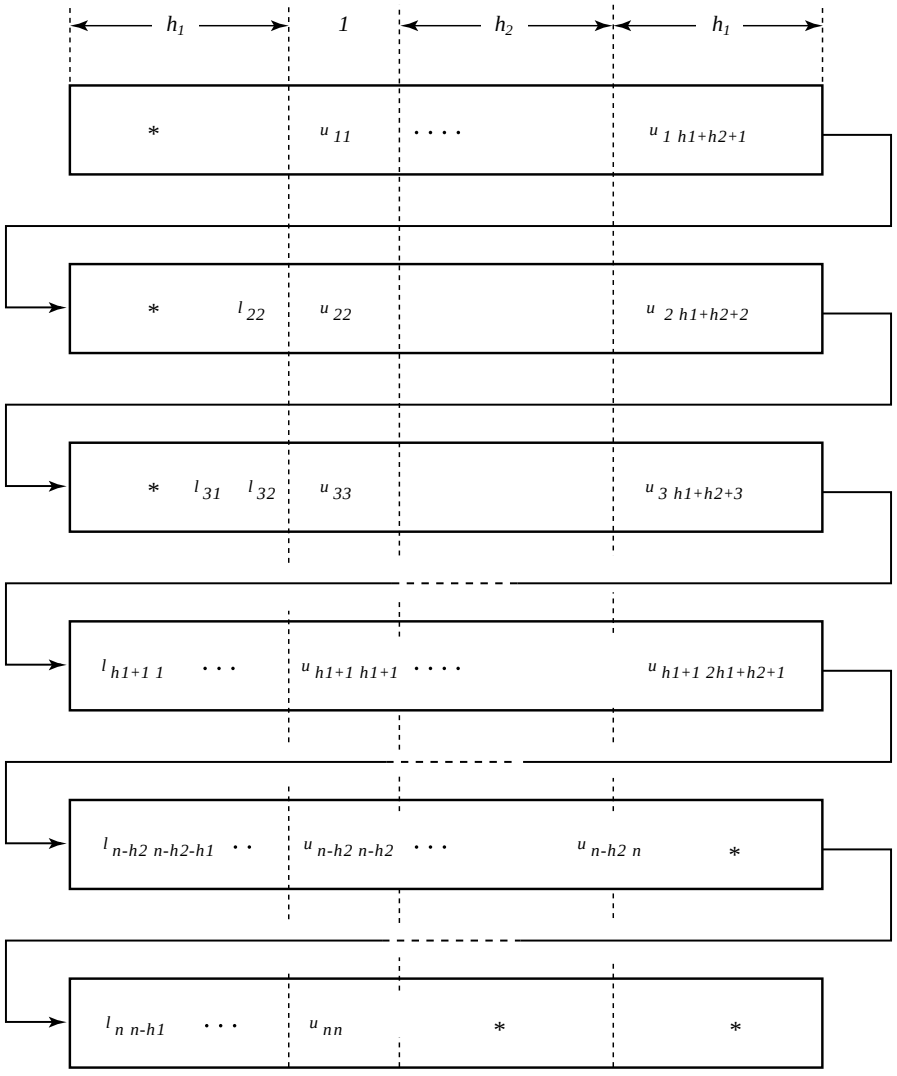
N.....入力．行列 $\boldsymbol{A}$ の次数.

NH1.....入力．行列 $\boldsymbol{A}$ の下バンド巾 h_1.

NH2.....入力．行列 $\boldsymbol{A}$ の上バンド巾 h_2.

IP.....入力. 部分ピボッティングによる行の入替えの履歴を示すトランスポジションベクトル．大きさ n の 1 次元配列.

ICON.....出力．コンディションコード.
　　表 VBLUX-1 参照.



＊： 不定値を示します。

図 VBLUX-1　配列 FA における *L* および *U* の格納方法

行列 ***L*** の対角要素を除いた *i* 行ベクトルが，FA((2×h_1＋h_2＋1)×(*i*－1)＋1 : (2×h_1＋h_2＋1)×(*i*－1)＋h_1) に格納します．また行列 ***U*** の *i* 行ベクトルを FA((2×h_1＋h_2＋1)×(*i*－1)＋h_1＋1 : (2×h_1＋h_2＋1)×*i*) に対角要素から連続に格納します．

表 VBLUX-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	—
20000	係数行列が非正則であった.	処理を打ち切る.
30000	N≦NH1,N≦NH2,NH1＜0,NH2＜0 または IP に誤りがあった.	

(3)　使用上の注意

a.　使用する副プログラム

SSL II・・・MGSSL

b.　注意

①　連立 1 次方程式を解く場合, サブルーチン VBLU を呼び出してから, 本サブルーチンを呼び出せば方程式を解くことができます．このとき，本サブルーチンの入力パラメタ（定数ベクトルは除く）は，サブルーチン VBLU の出力パラメタをそのまま指定してください.

c.　使用例

h_1＝h_2＝160,n＝160×160 なる非対称行列 $\boldsymbol{A}$ を係数行列とする，連立 1 次方程式 $\boldsymbol{Ax}=\boldsymbol{b}$ を解きます.

```
C     **EXAMPLE**
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER(NH=80)
      PARAMETER(NH1=NH)
      PARAMETER(NH2=NH)
      PARAMETER(N=NH*NH)
      PARAMETER(ALPHA=0.5/(NH1+1),BETA=-ALPHA)
      DIMENSION A((2*NH1+NH2+1)*2*N),B(N)
      DIMENSION C(2*NH1+NH2+1,N),IP(N),VW(N)
      EQUIVALENCE(A,C)
C
C     Zero clear
C
      DO 10 I=1,N*(3*NH+1)
      A(I)=0.0
   10 CONTINUE
C
      DO 15 I=1,N
      B(I)=0.0
      IP(I)=0
   15 CONTINUE
C
C     Coefficient Matrix is built
C
      DO 20 I=1,N
      C(NH1+1,I)=4.0
      B(I)=B(I)+4.0
      IF(I.GT.NH)THEN
      C(1,I)=-1.0+ALPHA
      B(I)=B(I)-1.0+ALPHA
      ENDIF
```

```
      IF(I+NH.LE.N)THEN
      C(1+NH1+NH2,I)=-1.0+BETA
      B(I)=B(I)-1.0+BETA
      ENDIF
      IF(I.GT.1.AND.MOD(I-1,NH).NE.0)THEN
      C(NH1,I)=-1.0+ALPHA
      B(I)=B(I)-1.0+ALPHA
      ENDIF
      IF(I+1.LE.N.AND.MOD(I,NH).NE.0)THEN
      C(NH1+2,I)=-1.0+BETA
      B(I)=B(I)-1.0+BETA
      ENDIF
   20 CONTINUE
C
C     Solve Banded linear equation
C
      EPSZ=0.0D0
      ICON=0
      CALL DVBLU(A,N,NH1,NH2,EPSZ,IS,IP,VW,ICON)
      PRINT*,'VBLU ICON= ',ICON
      IF(ICON.NE.0)STOP
      CALL DVBLUX(B,A,N,NH1,NH2,IP,ICON)
      PRINT*,'VBLUX ICON= ',ICON
      IF(ICON.NE.0)STOP
      PRINT*,'B(1)= ',B(1)
      PRINT*,'B(N)= ',B(N)
      STOP
      END
```

(4)　手法概要

前進代入，後退代入で次の式を解きます.

$LUx = Pb$

F17-13-0301 VCCVF,DVCCVF

複素データの離散畳み込み および 離散相関
CALL VCCVF(ZX,K,N,M,ZY,IVR,ISW,TAB,ICON)

(1)　機能

複素数データの 1 次元，多重の離散畳み込み または 離散相関をフーリエ変換を利用して計算します．

① 畳み込み

フィルターと，1 本のデータをそれぞれ $\{y_i\}$ と $\{x_j\}$ としたとき，以下で定義される畳み込みをデータ配列の各列に対して行い，$\{z_k\}$ を計算します．

$$z_k = \sum_{i=0}^{n-1} x_{k-i} y_i, \qquad k = 0,...,n-1$$

② 相関

フィルターと 1 本のデータをそれぞれ $\{y_i\}$ と $\{x_j\}$ としたとき，以下で定義される相関データ配列の各列に対して行い，$\{z_k\}$ を計算します．

$$z_k = \sum_{i=0}^{n-1} x_{k+i} \overline{y_i}, \qquad k = 0,...,n-1$$

ここで，x_j は周期 n の周期的データです．（(3)使用上の注意 b.注意①参照）

(2)　パラメタ

ZX .....入力．m 本の $\{x_j\}$ データ, j=0,...,n–1 を ZX(1:N,1:M)に格納します．
出力．m 本の $\{z_k\}$ k=0,...,n–1 が ZX(1:N,1:M)に格納されます．
ZX(K,M)なる複素型 2 次元配列．

K.....入力．配列 ZX の整合寸法．K ≥ N

N.....入力．1 本のデータの要素数 n．（(3)使用上の注意 b.注意②参照）

M.....入力．多重度 m.

ZY .....入力．フィルター $\{y_i\}$，i=0,...,n–1 を ZY(1:N)に格納します．
ZY(N)なる複素型 1 次元配列．（(3)使用上の注意 b.注意③,④参照）
ISW=0 または 2 のとき，値は保存されません．

IVR.....入力．畳み込みと相関の処理を選択します．
0　のとき　　畳み込みを計算します．
1　のとき　　相関を計算します．

ISW .....入力．制御情報．
0　のとき　1 回の CALL で処理を完了します．
以下の ISW 指定で順次呼び出すことで，処理のステップをわけて利用できます．（(3)使用上の注意 b.注意③参照）
1　のとき　TAB 設定のみ．
2　のとき　ZY のフーリエ変換を行います．

3　のとき　設定済みの TAB と変換済みの ZY を使用して畳み込みまたは相関を計算します.

TAB .....作業域．三角関数表を格納します．TAB(2×N)の 1 次元配列.
（(3)使用上の注意 b.注意③参照）

ICON.....出力．コンディションコード.
表 VCCVF-1 参照.

表 VCCVF-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	—
30000	N≦0, K<N, M≦0, ISW≠0,1,2,3, IVR≠0,1	処理を打ち切る.

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ UZFB2,UZFB3,UZFB4,UZFB5, UZFB8, UZFB6, UZFF2, UZFF3, UZFF4,UZFF5, UZFF8,UZFF6, UZFPB,UZFPF, UZFTB,UZFTF, UZUNI,MGSSL

b.　注意

①　非周期的データに対する畳み込みおよび相関について
データ x の長さが n_x，y の長さが n_y であるとき，$n \geq n_x + n_y - 1$ として
ZX(n_x+1：n,＊)および ZY(n_y+1：n)には 0 を設定しておくことで，非周期データに対する畳み込みおよび相関が計算できます．（c.使用例の例２を参照）
このとき，相関の z_k ($k = -n_y+1,...,-1$)に相当する値は，ZX($n-n_y$+2：n,＊)に格納されることになります.

②　n の値について
1 本のデータの要素数 n は任意の数が指定可能ですが, 2,3,5 の因子に分解される場合が高速です.

③　TAB および ZY の再利用について
同じ N で本ルーチンを繰り返し呼び出す場合，初回に ISW=0 または 1 で呼び出した後，2 回目以降は ISW=2 と 3 で呼び出すことにより TAB 作成を省略し，処理の高速化ができます．さらに，同じフィルターy を使用する場合には，ISW=0 または 2 で変換された ZY を ISW=3 の呼び出しで利用することで処理が効率化できます.
このように本ルーチンを繰り返し呼び出す場合，ZY に対して 2 回以上の変換が行われないように注意が必要です.

④　自己相関計算について
引数 ZX と ZY に同一の実引数を指定することで，自己相関の計算が可能です．このとき，ISW=2 の指定は無視されます．（c.使用例の例３を参照）

⑤　スタックサイズについて
本サブルーチンでは，内部で使用する作業域を自動割付け配列としてスタックに確保しています．このため，スタックが不足すると異常終了する可能性があ

ります．本サブルーチンが自動割付け配列でスタックに必要とするサイズは，単精度ルーチンでは 8×N byte．倍精度ルーチンでは，16×N byte になります．この他に，固定サイズの作業領域や，ユーザープログラムで必要とするスタック領域もあるので，limit コマンドまたは ulimit コマンドなどで十分な大きさのスタックサイズを指定することを薦めます．

c.　使用例

例１：周期的データに対する畳み込みを計算します．

```
C     ** PERIODIC CONVOLUTION EXAMPLE **
      IMPLICIT REAL*8(A-H,O-Z)
      PARAMETER(K=8,M=3)
      COMPLEX*16 X(K,M),Y(K)
      DIMENSION TAB(K*2)

      N=8

C     --SET SAMPLE DATA--
      DO 100 J=1,M
      DO 100 I=1,N
      X(I,J)=DCMPLX(FLOAT(I+J-1),FLOAT(I-J))
  100 CONTINUE
      DO 110 I=1,N
      Y(I)=DCMPLX(FLOAT(I*I),FLOAT(10-I))
  110 CONTINUE

      WRITE(*,*)'--INPUT DATA--'
      DO 120 J=1,M
      WRITE(*,900)J,(X(I,J),I=1,N)
  120 CONTINUE
      WRITE(*,910)(Y(I),I=1,N)

C     --CALL DVCCVF--
      IVR=0
      ISW=0
      CALL DVCCVF(X,K,N,M,Y,IVR,ISW,TAB,ICON)

      WRITE(*,*)'--OUTPUT DATA--'
      DO 130 J=1,M
      WRITE(*,900)J,(X(I,J),I=1,N)
  130 CONTINUE

  900 FORMAT('X(*,',I2,') :'/(12X,4('(',F8.2,',',F8.2,')')))
```

```
  910 FORMAT('Filter Y:'/(12X,4('(',F8.2,',',F8.2,')')))
      STOP
      END
```

例２：非周期的データに対する畳み込みを計算します.

```
C     ** NONPERIODIC CONVOLUTION EXAMPLE **
      IMPLICIT REAL*8(A-H,O-Z)
      PARAMETER(K=16,M=3)
      COMPLEX*16 X(K,M),Y(K)
      DIMENSION TAB(K*2)

      NX=7
      NY=9
      N=NX+NY-1
      IF(MOD(N,2).NE.0)N=N+1

C     --SET SAMPLE DATA--
      DO 100 J=1,M
      DO 110 I=1,NX
      X(I,J)=DCMPLX(FLOAT(I+J-1),FLOAT(I-J))
  110 CONTINUE
      DO 120 I=NX+1,N
      X(I,J)=(0.0D0,0.0D0)
  120 CONTINUE
  100 CONTINUE
      DO 130 I=1,NY
      Y(I)=DCMPLX(FLOAT(I*I),FLOAT(10-I))
  130 CONTINUE
      DO 140 I=NY+1,N
      Y(I)=(0.0D0,0.0D0)
  140 CONTINUE

      WRITE(*,*)'--INPUT DATA--'
      DO 150 J=1,M
      WRITE(*,900)J,(X(I,J),I=1,N)
  150 CONTINUE
      WRITE(*,910)(Y(I),I=1,N)

C     --CALL DVCCVF--
      IVR=0
      ISW=0
      CALL DVCCVF(X,K,N,M,Y,IVR,ISW,TAB,ICON)
```

```
      WRITE(*,*)'--OUTPUT DATA--'
      DO 160 J=1,M
      WRITE(*,900)J,(X(I,J),I=1,N)
  160 CONTINUE

  900 FORMAT('X(*,',I2,') :'/(12X,4('(',F8.2,',',F8.2,')')))
  910 FORMAT('Filter Y:'/(12X,4('(',F8.2,',',F8.2,')')))
      STOP
      END
```

例３：自己相関を計算します.

```
C     ** AUTOCORRELATION EXAMPLE **
      IMPLICIT REAL*8(A-H,O-Z)
      PARAMETER(K=8,M=3)
      COMPLEX*16 X(K,M)
      DIMENSION TAB(K*2)

      NX=4
      N=NX*2

C     --SET SAMPLE DATA--
      DO 100 J=1,M
      DO 110 I=1,NX
      X(I,J)=DCMPLX(FLOAT(I+J-1),FLOAT(I-J))
  110 CONTINUE
      DO 120 I=NX+1,N
      X(I,J)=(0.0D0,0.0D0)
  120 CONTINUE
  100 CONTINUE

      WRITE(*,*)'--INPUT DATA--'
      DO 130 J=1,M
      WRITE(*,900)J,(X(I,J),I=1,N)
  130 CONTINUE

C     --CALL DVCCVF--
      IVR=1
      ISW=1
      CALL DVCCVF(X,K,N,M,X,IVR,ISW,TAB,ICON)
      ISW=3
      CALL DVCCVF(X,K,N,M,X,IVR,ISW,TAB,ICON)
```

```
      WRITE(*,*)'--OUTPUT DATA--'
      DO 140 J=1,M
      WRITE(*,900)J,(X(I,J),I=1,N)
  140 CONTINUE

  900 FORMAT('X(*,',I2,') :'/(12X,4('(',F8.2,',',F8.2,')')))
      STOP
      END
```

(4)　手法概要

離散畳み込み，および離散相関はフーリエ変換を利用することで効率的に計算できることが知られています．離散畳み込みではデータとフィルターそれぞれの離散フーリエ変換を計算した後，それらの要素毎の積を計算し，逆変換することで結果が得られます．
離散相関でもほぼ同様に，データの離散フーリエ変換と，フィルターの離散フーリエ変換の共役を計算して，それらの要素毎の積を逆フーリエ変換することで計算されます．
詳細については“付録 参考文献一覧表”の[26]を参照してください．

F17-11-0501 VCFM1,DVCFM1

1 次元離散型複素フーリエ変換（2，3，5 および 7 の混合基底）
CALL VCFM1(X,N,ISW,ISN,W,ICON)

(1)　機能

1 次元複素フーリエ変換の正変換または逆変換を行います.

変換データ長 n は，2，3，5，7 の巾の積として表される数でなければなりません.

a.　1 次元フーリエ変換

$\{x_j\}$を入力し，（1.1）で定義する変換を行い，$\{n\ \alpha_k\}$を求めます.

$$n\alpha_k = \sum_{j=0}^{n-1} x_j \omega_n^{-jk} \quad ,k = 0,1,...,n-1 \qquad ,\omega_n = \exp(2\pi i / n) \tag{1.1}$$

b.　1 次元フーリエ逆変換

$\{\alpha_k\}$を入力し，（1.2）で定義する変換を行い，$\{x_j\}$を求めます.

$$x_j = \sum_{k=0}^{n-1} \alpha_k \omega_n^{jk} \quad ,j = 0,1,...,n-1 \qquad ,\omega_n = \exp(2\pi i / n) \tag{1.2}$$

(2)　パラメタ

X.....入力. 複素数データ. 変換するデータ$\{x_j\}$又は$\{\alpha_k\}$を，X(1:N)に格納します.
出力．複素数データ．変換された結果$\{n\ \alpha_k\}$又は$\{x_j\}$が，X(1:N)に格納されます.
X(N)なる複素数型 1 次元配列.

N.....入力．変換データ長(n).

ISW .....入力．制御情報.
ISW = 1 のとき：初回呼び出し．W に三角関数テーブルおよび制御情報を格納しフーリエ変換を行います.
ISW ≠ 1 のとき：同じ大きさのデータの変換を初回以降引き続き行います．このとき，初回に W に格納された三角関数テーブルおよび制御情報を利用するため，N，W の内容を初回呼び出し後に変更してはなりません.
出力．ISW=1 を設定して呼び出したとき，ISW=0 が設定されます．このため 2 回目以降特に ISW に値を設定せずに，配列 X に変換データを格納して呼び出すことで効率よく変換を行うことができます.

ISN.....入力．変換か逆変換かを指定.
変換　：ISN=1
逆変換：ISN= –1

W.....作業域.
ISW = 1 のとき，データ長 N の変換用の三角関数テーブルが格納されます.

ISW≠1のとき, ISW = 1 を指定した初回呼び出しで作成された三角関数テーブルを利用します．（(3) 使用上の注意 b.注意②参照）.

W(2N+70)なる複素数型 1 次元配列.

ICON.....出力．コンディションコード.

表 VCFM1-1 参照.

表 VCFM1-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	―
20000	2 回目以降の呼び出しで N の値が初回呼び出しのときの値と異なる.	処理を打ち切る.
30000	ISN の値が正しくない.	
30008	N の変換数が 2，3，5，7 を基底としていない.	

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ UCFM1X,UFR10T,UFR10TI,UFR16,UFR16T,UFR16TI,UFR2T,UFR2TI, UFR3T,UFR3TI,UFR4,UFR4_5,UFR4_5I,UFR4_8,UFR4_8I,UFR4_R,UFR4_RI, UFR4_S,UFR4_SI,UFR4_U,UFR4_UI,UFR4I,UFR4T,UFR4TI,UFR5T,UFR5TI, UFR6T,UFR6TI,UFR7_7,UFR7T,UFR7TI,UFR8,UFR8_6,UFR8_6I,UFR8_7I, UFR8_9,UFR8_9I,UFR8I,UFR8T,UFR8TI,UFR9T,UFR9TI,UFRDX2,UFTBL, UFTBLMX2,UFTBLX,UFTBLX8,UFTINY,UFTINYI,UPBITR,UPERM, UPERM5,UPERM6,UPERM7,UPERM8,UPERM9,MGSSL

b.　注意

①　一般的なフーリエ変換の定義

1 次元離散型複素フーリエ変換および，逆変換は一般的に（3.1），（3.2）で定義されます.

$$\alpha_k = \frac{1}{n}\sum_{j=0}^{n-1} x_j \omega_n^{-jk} \quad ,k = 0,1,...,n-1 \tag{3.1}$$

$$x_j = \sum_{k=0}^{n-1} \alpha_k \omega_n^{jk} \quad , j = 0,1,...,n-1 \tag{3.2}$$

ここで，$\omega_n = \exp(2\pi i/n)$

本サブルーチンでは，（3.1），（3.2）の左辺に対応して$\{n\alpha_k\}$または$\{x_j\}$を求めます．したがって，結果の正規化は必要に応じて行って下さい.

②　作業域 W について

同じ N で複数回本サブルーチンを呼び出す場合, 初回に ISW=1 で呼び出した後, 2 回目以降は ISW≠1 で呼び出すことにより, 配列 W 内に作成した三角関数テーブルが再利用されます．配列 W は作業域としても利用されるため，2 回目以降

の呼び出しであっても書き込みが行われることに注意してください.

c.　使用例

1 次元 FFT を計算します.

```
c     **example**
      implicit real*8(a-h,o-z)
      parameter(n=640)
      complex*16 x(n),w(2*n+70)
c
      do i=1,n
      x(i)=i/dble(n)
      enddo
c
c     do the forward transform
c
      isw=1
      call dvcfm1(x,n,isw,1,w,icon)
      if(icon.ne.0)then
      print*,'icon = ',icon
      stop
      endif
c
c     do the reverse transform
c
      call dvcfm1(x,n,isw,-1,w,icon)
      if(icon.ne.0)then
      print*,'icon = ',icon
      stop
      endif
c
      tmp=0.0d0
      do i=1,n
      tmp=max(tmp,abs(x(i)/dble(n)-i/dble(n)))
      enddo
c
      print*,'error=',tmp
      stop
      end
```

(4)　手法概要

本ルーチンは，1 次元の複素フーリエ変換を行います.

以下の 4-step algorithm をベースとしたスカラ計算機向けのアルゴリズムを利用しています．変換を行うデータをキャッシュ上で繰り返し連続に参照し，三角関数テーブルの利

用を削減する工夫を行っています.

① 変数の次数を $n = pq$ と分解します.

② 次数が $n = pq$ と分解できたら，以下の 4-step algorithm を実行します.

$$z^{(1)}_{j1+k0q} = \sum_{j0=0}^{p-1} \omega_p^{k0\,j0} x_{j1+j0q} \qquad j_1 = 0,...,q-1, k_0 = 0,...,p-1 \tag{4.1}$$

$$z^{(2)}_{j1+k0q} = \omega_n^{k0\,j1} z^{(1)}_{j1+k0q} \qquad k_0 = 0,...,p-1, j_1 = 0,...,q-1 \tag{4.2}$$

$$z^{(3)}_{k0+j1p} = z^{(2)}_{j1+k0q} \qquad k_0 = 0,...,p-1, j_1 = 0,...,q-1 \tag{4.3}$$

$$y_{k0+k1p} = \sum_{j1=0}^{q-1} \omega_q^{k1\,j1} z^{(3)}_{k0+j1p} \qquad k_0 = 0,...,p-1, k_1 = 0,...,q-1 \tag{4.4}$$

第 1 および第 4 ステップはそれぞれ次数 p および q の多重フーリエ変換です.

③ $N = N_1 \times N_2 \times ... \times N_m$ と分解できたとき，順次 4-step algorithm を適用してフーリエ変換を計算します.

F17-11-0701 VCFT3,DVCFT3

1 次元離散型複素フーリエ変換（2 基底, 一定間隔を持つデータ）
CALL VCFT3(X,N,NDIST,ISW,ISN,W,ICON)

(1)　機能

1 次元複素フーリエ変換の正変換または逆変換を行います.
変換データ長 n は，2 の巾として表される数でなければなりません.

a.　1 次元フーリエ変換

$\{x_j\}$を入力し，（1.1）で定義する変換を行い，$\{n\ \alpha_k\}$を求めます.

$$n\alpha_k = \sum_{j=0}^{n-1} x_j \omega_n^{-jk} \quad ,k = 0,1,...,n-1 \qquad ,\omega_n = \exp(2\pi i / n) \tag{1.1}$$

b.　1 次元フーリエ逆変換

$\{\alpha_k\}$を入力し，（1.2）で定義する変換を行い，$\{x_j\}$を求めます.

$$x_j = \sum_{k=0}^{n-1} \alpha_k \omega_n^{jk} \quad ,j = 0,1,...,n-1 \qquad ,\omega_n = \exp(2\pi i / n) \tag{1.2}$$

(2)　パラメタ

X.....入力．複素数データ．変換するデータ$\{x_j\}$又は$\{\alpha_k\}$を，X(1)，X(1+ NDIST), …，X(1+ (n-1)*NDIST)に格納します.
出力．複素数データ．変換された結果$\{n\ \alpha_k\}$又は$\{x_j\}$が，X(1)，X(1+ NDIST), …，X(1+ (n-1)*NDIST)に格納されます.
複素数型 1 次元配列.

N.....入力．変換データ長(n).

NDIST.....入力. 配列 X にデータを格納する一定の間隔. 正の整数. NDIST=1 のとき，連続に格納されます.

ISW .....入力．制御情報.
ISW = 1 のとき：初回呼び出し．W に三角関数テーブルおよび制御情報を格納しフーリエ変換を行います.
ISW ≠ 1 のとき：同じ大きさのデータの変換を初回以降引き続き行います. このとき，初回に W に格納された三角関数テーブルおよび制御情報を利用するため，N，W の内容を初回呼び出し後に変更してはなりません.
出力．ISW=1 を設定して呼び出したとき，ISW=0 が設定されます．このため 2 回目以降特に ISW に値を設定せずに，配列 X に変換データを格納して呼び出すことで効率よく変換を行うことができます.

ISN.....入力．変換か逆変換かを指定.
変換　：ISN=1

逆変換：ISN= –1

W.....作業域.

ISW = 1 のとき，データ長 N の変換用の三角関数テーブルが格納されます.

ISW ≠ 1 のとき, ISW = 1 を指定した初回呼び出しで作成された三角関数テーブルを利用します．（(3) 使用上の注意 b.注意②参照）.

W(2N+70)なる複素数型 1 次元配列.

ICON.....出力．コンディションコード.

表 VCFT3-1 参照.

表 VCFT3-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	—
20000	2 回目以降の呼び出しで N の値が初回呼び出しのときの値と異なる.	処理を打ち切る.
30000	ISN の値が正しくない. NDIST が正の整数ではない.	
30008	N の変換数が 2 を基底としていない.	

(3)　使用上の注意

a.　使用する副プログラム

SSL II・・・ UFFT16,UFFT32,UFFT16INV,UFFT32INV,UFFT64,UFFT64INV,UFFT32_1,
UFFT32INV_1,U1DFFTDST16,UFFT16DST,UFFT16INVDST,U1DFFTDST32,
UFFT32DST,UFFT32INVDST,U1DFFTDST64,UFFT64DST,UFFT32_1DST,
UFFT64INVDST,UFFT32INV_1DST,UFFT256,UFFT8TWIST2,UFFT128,
UFFT8TWIST1,UFFT8TWIST4,UFFT512,UFFT1024,UFFT4096,UFFT8TWIST3,
UFFT8TWIST43,U2DFFT512_2,U2DFFT256_2,U2DFFT128_2,UFFT8TWIST43_16,
UFFT2048,UFFT8192,UFFT256INV,UFFT8TWISTINV2,UFFT128INV,
UFFT8TWISTINV1,UFFT8TWISTINV4,UFFT512INV,UFFT1024INV,
UFFT4096INV,UFFT8TWISTINV3,UFFT8TWISTINV43,U2DFFT512INV_2,
U2DFFT256INV_2,U2DFFT128INV_2,UFFT8TWISTINV43_16,UFFT2048INV,
UFFT8192INV,UFFT16TWIST1,UFFT16TWIST2,UFFT16TWIST4_8,
UFFT16TWIST4,UFFT16TWISTINV1,UFFT16TWISTINV2, UFFT16TWISTINV4_8,
UFFT16TWISTINV4,U1DFFTDST4096,UFFT4096DST,UFFT4096INVDST,
UFFT8TWIST1DST,UFFT8TWIST43DST,UFFT8TWISTINV1DST,
UFFT8TWISTINV43DST,U1DFFTDST512,UFFT512DST,UFFT512INVDST,
UFFT8TWIST4DST,UFFT8TWISTINV4DST,U1DFFTDST1024,UFFT1024DST,
UFFT1024INVDST,UFFT16TWIST4DST,UFFT16TWISTINV4DST,
U1DFFTDST1024,U1DFFTDST2048,UFFT16TWIST1DST,
UFFT16TWISTINV1DST,UFFT2048DST,UFFT2048INVDST,
UFFT8TWIST43_16DST,UFFT8TWISTINV43_16DST,U1DFFTDST256,
UFFT16TWIST2DST,UFFT16TWISTINV2DST,UFFT256DST,UFFT256INVDST,

U1DFFTDST128,UFFT128DST,UFFT128INVDST,UFFT8TWIST2DST,
UFFT8TWISTINV2DST,U1DFFTDST8192,UFFT16TWIST4_8DST,
UFFT16TWISTINV4_8DST,UFFT8192DST,UFFT8192INVDST,UCFTDIS3X,
UFR16T,UFR16TI,UFR2T,UFR2TDSTPE,UFR2TI,UFR2TIDSTPE,UFR4T,
UFR4TDSTPE,UFR4TI,UFR4TIDSTPE,UFR8T,UFR8TDSTPE,UFR8TI,
UFR8TIDSTPE,UFTBL,UFTBLMX2,UFTBLX, UFTBLX8, MGSSL

b.　注意

①　一般的なフーリエ変換の定義

1 次元離散型複素フーリエ変換および，逆変換は一般的に（3.1），（3.2）で定義されます．

$$\alpha_k = \frac{1}{n}\sum_{j=0}^{n-1} x_j \omega_n^{-jk} \quad ,k = 0,1,...,n-1 \tag{3.1}$$

$$x_j = \sum_{k=0}^{n-1} \alpha_k \omega_n^{jk} \quad , j = 0,1,...,n-1 \tag{3.2}$$

ここで，$\omega_n = \exp(2\pi i/n)$

本サブルーチンでは，（3.1），（3.2）の左辺に対応して$\{n\alpha_k\}$または$\{x_j\}$を求めます．したがって，結果の正規化は必要に応じて行って下さい．

②　作業域 W について

同じ N で複数回本サブルーチンを呼び出す場合，初回に ISW=1 で呼び出した後，2 回目以降は ISW≠1 で呼び出すことにより，配列 W 内に作成した三角関数テーブルが再利用されます．配列 W は作業域としても利用されるため，2 回目以降の呼び出しであっても書き込みが行われることに注意してください．

c.　使用例

複数の一定の間隔をもったデータ列に対して，1 次元 FFT を計算します．

```
c     **example**
      implicit real*8(a-h,o-z)
      parameter(n=1024,mult=16,npad=3,ndist=mult+npad)
      complex*16 x(ndist,n),w(2*n+70)
c
      do j=1,mult
      do i=1,n
      x(j,i)=i/dble(n)+j
      enddo
      enddo
c
c     multiple forward transform
c
      isw=1
      do j=1,mult
```

```
      call dvcft3(x(j,1),n,ndist,isw,1,w,icon)
      if(icon.ne.0)then
      print*,'icon = ',icon
      endif
      enddo
c
c     multiple reverse transform
c
      do j=1,mult
      call dvcft3(x(j,1),n,ndist,isw,-1,w,icon)
      if(icon.ne.0)then
      print*,'icon = ',icon
      endif
      enddo
c
      tmp=0.0d0
      do j=1,mult
      do i=1,n
      tmp=max(tmp,abs(x(j,i)/dble(n)-(i/dble(n)+j)))
      enddo
      enddo
c
      print*,'error = ',tmp
      stop
      end
```

(4)　手法概要

本ルーチンは, 一定の間隔をもつデータ列に対してスカラ計算機で 1 次元の 2 基底の複素フーリエ変換を高速に計算します.

A72-11-0101 VCGD,DVCGD

正値対称スパース行列の連立 1 次方程式（前処理付き CG 法，対角形式格納法）
CALL VCGD(A,K,NW,N,NDLT,B,IPC,ITMAX,ISW,OMEGA,EPS,IGUSS,X, ITER,RZ,VW,IVW,ICON)

(1)　機能

$n \times n$ の正規化された正値対称なスパース行列を係数行列とする連立 1 次方程式を前処理付き CG 法で解きます．

$$\boldsymbol{Ax} = \boldsymbol{b}$$

$n \times n$ の係数行列は，対角要素が 1 になるように正規化されており，対角要素を除いた非零要素を対角形式のスパース行列の格納法で格納します．

正値対称なスパース行列を係数行列とする連立 1 次方程式の正規化および対角形式の格納方式については，“第 I 部 概要 3.2.1.2 正値対称スパース行列の格納方法”を参照してください．対角形式の格納法は，係数行列 $\boldsymbol{A}$ の非零要素が主対角ベクトルに平行な幾つかの対角線上のベクトルにのみ存在することを仮定しています．

偏微分方程式を，特に定義される領域の境界と平行な格子で，離散化して得られる連立 1 次方程式の場合，このような構造を持ちます．この場合，各要素が係数行列の何列ベクトルにあるかを示す情報が不要で，主対角ベクトルからの距離のみを要し，効率的です．

(2)　パラメタ

A.....入力．A（K,NW）なる 2 次元配列．正規化された正値対称スパース行列の係数行列の非零要素を対角形式で格納します．
正規化された正値対称スパース行列の対角形式の格納方法については，“第 I 部 概説 3.2.1.2 正値対称スパース行列の格納方法 b.正値対称スパース行列の対角形式の格納方法”参照．

K.....入力．配列 A の整合寸法（≧N）．

NW.....入力．配列 A の 2 次元目の大きさ．対角形式の格納方法で係数行列 $\boldsymbol{A}$ を格納する，対角方向のベクトルの本数．偶数．

N.....入力．行列 $\boldsymbol{A}$ の次数 n.

NDLT.....入力．NDLT（NW）なる 1 次元配列で，主対角ベクトルからの距離を示します．

B.....入力．大きさ n の 1 次元配列．連立 1 次方程式の右辺の定数ベクトルを格納します．

IPC .....入力．前処理の制御情報．
1 のとき　前処理なし．
2 のとき　Neumann の前処理．
3 のとき　不完全コレスキー分解による前処理．
このとき OMEGA を指定する必要があります．
（(3) 使用上の注意 b.注意③参照）．

ITMAX .....入力．反復回数の上限（＞0）．

ISW .....入力．制御情報．

1 のとき　初回の呼び出し．

2 のとき　2 回目以降の呼び出し．ただし A,NDLT,VW,IVW の値は初回呼び出しで設定された値を使用するため，変更してはなりません．

（(3) 使用上の注意 b.注意①参照）．

OMEGA.....入力．不完全コレスキー分解のための修正値．0≦OMEGA≦1

IPC＝3 のとき使用されます．

（(3) 使用上の注意 b.注意③参照）．

EPS.....入力．収束判定に用いられる判定値．

RZ＜EPS を満たしたとき収束したと見なします．

EPS が 0.0 以下のとき，$\varepsilon \times |b|$ が EPS として設定されます．εは倍精度ルーチンでは，10^{-6}，単精度ルーチンでは 10^{-4} が設定されます．

（(3) 使用上の注意 b.注意②参照）．

IGUSS.....入力．配列 X に指定した解ベクトルの近似値から反復を開始するかどうかの情報を設定します．

0 のとき　　解ベクトルの近似を指定しません．

0 以外のとき　配列 X に指定された解ベクトルの近似から反復計算を開始します．

X.....入力．大きさ *n* の 1 次元配列．連立 1 次方程式の解ベクトルの近似値を指定することができます．

出力．大きさ *n* の 1 次元配列．連立 1 次方程式の解ベクトルが格納されます．

ITER.....出力．実際行った反復の回数．

RZ .....出力．収束判定を行った残差 *rz* の平方根の値．

（(3) 使用上の注意 b.注意②参照）．

VW.....作業域．

1) IPC＝3 のとき

大きさ，K×(NW＋6)＋2×*NBAND* なる 1 次元配列．*NBAND* は下バンド幅または上バンド幅の大きさ．

2) その他のとき

大きさ，K×5＋2×*NBAND*，なる 1 次元配列．*NBAND* は下バンド幅または上バンド幅の大きさ．

IVW .....作業域．大きさ (K＋1)×4 なる 1 次元配列．

ICON.....出力．コンディションコード．表 VCGD-1 参照．

表 VCGD-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	—
20001	反復回数の上限に達した.	処理を打ち切る．配列 X には，そのときまでに得られている近似値を出力するが精度は保証できない.
20003	ブレークダウンを起こした.	
30003	ITMAX≦0	処理を打ち切る.
30005	K＜N	
30006	不完全 LL^T 分解ができなかった.	
30007	ピボットが負になった.	
30089	NW が偶数でない.	
30091	*NBAND*＝0	
30092	NW≦0，N≦0	
30093	K≦0	
30096	OMEGA＜0，OMEGA＞1	
30097	IPC＜1,IPC＞3	
30102	上三角部分が正しく格納されてない.	
30103	下三角部分が正しく格納されてない.	
30104	上三角の本数が下三角の本数と等しくない.	
30105	ISW≠1,2 であった.	
30200	｜NDLT(*i*)｜＞*n*-1 または NDLT(*i*)＝0	

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ AMACH,UGCRI,UGULD,UGECD,UGFCD,UGINP,UGIPD,UGITB, UGITI,UGITN,UGITT,UGMVD,UGCGP,USSCP,USSPS,UGSD2,UGSD3, UGSTE,UGSWD,USVAD,USVCN,USVCP,USVSC,USVSU,USVUP,USVN1,USVN2,USVNM,UGWVD,URELT,MGSSL

b.　注意

①　同じ係数行列を持ち定数ベクトルの異なる複数組の連立 1 次方程式を IPC＝3 で解く場合，1 回目は ISW＝1 で解き，2 回目以降は ISW＝2 で解きます．2 回目以降では，1 回目の呼び出しで得られた不完全コレスキー分解の結果を再利用して解きます.

② 収束判定

n 回目の反復で求められた解が収束したかどかは次のように判定します.

r は残差ベクトル，$\boldsymbol{M}$ は前処理行列,

$$\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_n$$

$rz = r^T \boldsymbol{M}^{-1} r$　としたとき,

RZ＝$\sqrt{(rz)}$ ＜EPS　を満たすとき

③ 前処理

2 種類の前処理および前処理なしの機能が提供されています.

楕円型の偏微分方程式を離散化して解く場合は，不完全コレスキー分解による前処理が有効です.

$A = I - N$ としたとき，連立 1 次方程式 $(\boldsymbol{I}-\boldsymbol{N})\boldsymbol{x}=\boldsymbol{b}$ の前処理 M は以下のようになります.

IPC＝1	前処理なし	$\boldsymbol{M}=\boldsymbol{I}$
IPC＝2	Neumann	$\boldsymbol{M}^{-1}=(\boldsymbol{I}+\boldsymbol{N})$
IPC＝3	不完全コレスキー分解	$\boldsymbol{M}=\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$

IPC＝2 のとき，前処理行列も正定値であることが必要です．例えば$(\boldsymbol{I}+\boldsymbol{N})$の対角優位性はそのための十分条件になります．また，行列 $\boldsymbol{N}$ の固有値の絶対値で最大の値が 1 より大きく前処理行列が $\boldsymbol{A}$ の逆行列の良い近似にならない場合などでは，前処理を適用しても収束が改善しない可能性があります.

IPC＝3 のとき，OMEGA(0≦OMEGA≦1) に値を設定する必要があります.

OMEGA＝0 のときは，不完全コレスキー分解，OMEGA＝1 のときは修正不完全コレスキー分解です.

偏微分方程式の離散化から得られる連立 1 次方程式に対しては，OMEGA＝0.92 から 1.00 が最適であることが経験から分かっています.

IPC＝3 のとき，方程式は前処理の効率化のためにウェーブフロント順に並び換えられます.

c. 使用例

n＝51200 で対角ベクトルの距離が+5,-5 の正値対称スパース行列の連立 1 次方程式を解きます.

```
C     **EXAMPLE**
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (N=51200,K=N+1)
      PARAMETER (NW=2,IWKS=4,N2=K+1)
      PARAMETER (NVW=K*(NW+6)+10)
      REAL*8 B(N),EPS,OMEGA,RZ,VW(NVW),X(N)
      INTEGER NDLT(NW)
      REAL*8 A(K,NW)
      INTEGER IVW(N2,IWKS)
C
```

```
C INITIALISE A
C
      CALL SET(A,NDLT,K,NW,N)
      ISHIFT=0
      DO 10 J=1,NW
      ISHIFT=MAX(ISHIFT,ABS(NDLT(J)))
   10 CONTINUE
C COMPUTE RHS SO AX=B SO WE KNOW SOLUTION X (X(I)=I)
      DO 30 I=1,N
   30 VW(I+ISHIFT)=I
C
C B=(A-E)*X+X
C
      CALL DVMVSD(A,K,NW,N,NDLT,ISHIFT,VW,B,ICON)
      DO 70 I=1,N
      B(I)=B(I)+VW(I+ISHIFT)
   70 CONTINUE
C
      ITMAX=8*SQRT(N+0.1)
      EPS=1D-10
      OMEGA=0D0
      ISW=1
      IGUSS=0
      DO 100 IPC=1,3
      IF(IPC.EQ.3) OMEGA=0.98
      CALL DVCGD(A,K,NW,N,NDLT,B,IPC,ITMAX,ISW,OMEGA,
     &           EPS,IGUSS,X,ITER,RZ,VW,IVW,ICON)
      IF(ICON.NE.0) WRITE(6,*)'ICON=',ICON
      IF(RZ.LE.EPS) WRITE(6,41)'CONVERGED. ACCURACY=',RZ
      IF(RZ.GT.EPS) WRITE(6,41)'FAILED. ACCURACY=',RZ
      WRITE(6,*)'X'
      DO 60 I=1,MIN(N,16),4
   60 WRITE(6,42) I,(X(M),M=I,I+3)
  100 CONTINUE
   42 FORMAT(1X,I3,4(1X,F20.10))
   41 FORMAT(A,2X,E10.3)
      STOP
      END

      SUBROUTINE SET(A,NDLT,K,NW,N)
      REAL*8 A(K,NW)
      INTEGER NDLT(NW)
```

```
      DO 10 J=1,NW
      DO 10 I=1,K
   10 A(I,J)=0D0
      N3=5
      NDLT(1)=N3
      NDLT(2)=-N3
      DO 20 I=1,N
      L=I
      IF(L.LE.N-N3) THEN
      A(I,1)=-0.25D0
      ENDIF
   20 CONTINUE
      DO 30 I=1,N
      L=I
      IF(L.GE.N3+1.AND.L.LE.N) THEN
      A(I,2)=-0.25D0
      ENDIF
   30 CONTINUE
      RETURN
      END
```

(4)　手法概要

標準的な共役勾配法のアルゴリズム（“付録　参考文献一覧表”の [14] 参照）が使われています.

不完全コレスキー分解による前処理の方法は“付録　参考文献一覧表”の [30] を参照してください. また Wavefront ordering によるベクトル化は“付録 参考文献一覧表”の [23] を参照してください.

(5)　謝辞

ITPACK および NSPCG の著者に，修正不完全コレスキー分解および Wavefront ordering のルーチンを利用することを許可していただいたことに感謝致します.

A72-12-0101 VCGE,DVCGE

正値対称スパース行列の連立 1 次方程式 （前処理付き CG 法，ELLPACK 形式格納法）
CALL VCGE(A,K,NW,N,ICOL,B,IPC,ITMAX,ISW,OMEGA,EPS,IGUSS, X,ITER,RZ,VW,IVW,ICON)

(1)　機能

$n \times n$ の正規化された正値対称なスパース行列を係数行列とする連立 1 次方程式を前処理付き CG 法で解きます.

$$\boldsymbol{Ax} = \boldsymbol{b}$$

$n \times n$ の係数行列は，対角要素が 1 になるように正規化されたもので，対角要素を除く非零要素を ELLPACK 形式の格納法で格納します.

正値対称なスパース行列を係数行列とする連立 1 次方程式の正規化については，“第 I 部　概説　3.2.1.2　正値対称スパース行列の格納方法”を参照してください.

(2)　パラメタ

A.....入力．A(K,NW) なる 2 次元配列．係数行列の非零要素を A(1:N,NW) の部分に格納します.
IPC＝3 のとき，A(＊,1:NW/2) に上三角行列部分が，A(＊,NW/2＋1:NW) に下三角行列部分が格納されていないとき，このように格納し直します.
正規化された正値対称スパース行列の ELLPACK 形式の格納法については，“第 I 部 概説 3.2.1.2 正値対称スパース行列の格納方法　a. 正値対称スパース行列の ELLPACK 形式の格納方法”を参照してください.
（(3) 使用上の注意 b.注意①参照）.

K.....入力．配列 A,ICOL 整合寸法（≧N）.

NW.....入力．配列 A の 2 次元目の大きさ.
上三角行列の行ベクトルの非零要素の最大値を *NSU* とし，下三角行列の行ベクトルの非零要素の最大値を *NSL* とするとき，$2 \times max(NSU,NSL)$.
詳しくは，“第 I 部 概説 3.2.1.2 正値対称スパース行列の格納方法　a.正値対称スパース行列の ELLPACK 形式の格納方法”を参照してください.

N.....入力．行列 *A* の次数 *n*.

ICOL .....入力．ICOL(K,NW) なる 2 次元配列で，非零要素がどの列ベクトルに属するかの情報を ICOL(1:N,NW) の部分に格納します.

B.....入力．大きさ *n* の 1 次元配列．連立 1 次方程式の右辺の定数ベクトルを格納します.

IPC .....入力．前処理の制御情報.
1 のとき　前処理なし.

2 のとき　Neumann の前処理.

3 のとき　不完全コレスキー分解による前処理.

このとき OMEGA を指定する必要があります.

（(3) 使用上の注意 b.注意④参照）.

ITMAX .....入力．反復回数の上限（＞0）.

ISW .....入力．制御情報.

1 のとき　初回の呼び出し.

2 のとき　2 回目以降の呼び出し．ただし A,ICOL,VW,IVW の値は 1 回目に設定された値を使用するため，変更してはなりません.

（(3) 使用上の注意 b.注意②参照）.

OMEGA.....入力．不完全コレスキー分解のための修正値．0≦OMEGA≦1

EPS.....入力．収束判定に用いられる判定値.

RZ＜EPS を満たしたとき収束したと見なします.

EPS が 0.0 以下のとき，$\varepsilon\times|\boldsymbol{b}|$ が EPS として設定されます．εは倍精度ルーチンでは 10^{-6} が，単精度ルーチンでは 10^{-4} が設定されます.

（(3) 使用上の注意 b.注意③参照）.

IGUSS.....入力．配列 X に指定された解ベクトルの近似値から反復計算を開始するかを示す制御情報.

0 のとき　　解ベクトルの近似値は指定しません.

0 以外のとき　配列 X に指定された解ベクトルの近似値から反復を開始します.

X.....入力．大きさ *n* の 1 次元配列．連立 1 次方程式の解の近似ベクトルを指定することができます.

出力．連立 1 次方程式の解ベクトルが格納されます.

ITER.....出力．実際行った反復の回数.

RZ .....出力．収束判定を行った残差 *rz* の平方根.

（(3) 使用上の注意 b.注意②参照）.

VW.....作業域.

－　IPC＝3 のとき

大きさ，K×NW＋4×N なる 1 次元配列.

－　その他のとき

大きさ N×3 の 1 次元配列.

IVW .....作業域.

－　IPC＝3 のとき

大きさ，K×NW＋N×4 なる 1 次元配列.

－　その他のとき

大きさ N×4 の 1 次元配列.

ICON.....出力．コンディションコード.

表 VCGD-1 参照.

表 VCGE-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	—
10000	A,ICOL の要素を ***U/L*** と並び換えた.	処理は続行する.
20001	反復回数の上限に達した.	処理を打ち切る．配列 X には，そのときまでに得られている近似値を出力するが精度は保証できない.
20003	ブレークダウンを起こした.	
30003	ITMAX≦0	処理を打ち切る.
30005	K＜N	
30006	不完全 LL^T 分解ができなかった.	
30007	ピボットが負になった.	
30092	NW≦0	
30093	K≦0，N≦0	
30096	OMEGA＜0，OMEGA＞1	
30097	IPC＜1，IPC＞3	
30098	ISW≠1,2 であった.	
30100	NW≠2×max(NSU,NSL)	
30104	上三角又は下三角部分が正しく格納されていない.	
負の数	第（-icon）行に非零対角要素がある.	

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ AMACH,UGECP,UGEUL,UGEFA,UGEMV,UGEPD,UGECG,UGEPM, UGEPV,UGESV,UGEWV,URELT,MGSSL

b.　注意

① スパース行列は ELLPACK 形式（“付録　参考文献一覧表”の [23] , [34] 参照）と呼ばれる格納方法で格納します.

上三角部分を A(＊,1:NW/2) に下三角部分を A(＊,NW/2＋1:NW)に格納します. ここで NW＝2×*max*(*NSU*,*NSL*).

IPC＝3 以外（不完全コレスキー分解による前処理以外の前処理を指定するとき）では，“第 I 部 概説 3.2.1.2 正値対称スパース行列の格納方法 a.正値対称スパース行列の ELLPACK 形式の格納方法”で記述されているより，条件が緩い格納方法でも受入れます．つまり，正規化された正値対称スパース行列で対角要素を除いたものを，一般スパース行列の ELLPACK 形式の格納方法で格納したものも入力として受け入れます．このとき NW の値は，2×*max*(*NSU*,*NSL*) である必要はありません.

② 同じ係数行列を持ち定数ベクトルの異なる複数組の連立 1 次方程式を IPC＝3 で解く場合，1 回目は ISW＝1 で解き，2 回目以降は ISW＝2 で解きます．2 回目以降では，1 回目の呼び出しで得られた不完全コレスキー分解の結果を再利用します．

③ 収束判定

n 回目の反復で求められた解が収束したかどかは次のように判定します．

r は残差ベクトル，***M*** は前処理行列，

$$\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_n$$

$rz = r^{\mathrm{T}}\boldsymbol{M}^{-1}r$　としたとき，

RZ＝$\sqrt{(rz)}$ ＜EPS　を満たすとき

④ 前処理

2 種類の前処理および前処理なしの機能が提供されています．

$\boldsymbol{A}=\boldsymbol{I}-\boldsymbol{N}$ としたとき，連立 1 次方程式 $(\boldsymbol{I}-\boldsymbol{N})\boldsymbol{x}=\boldsymbol{b}$ の前処理 ***M*** は以下のようになります．

IPC＝1	前処理なし	$\boldsymbol{M}=\boldsymbol{I}$
IPC＝2	Neumann	$\boldsymbol{M}^{-1}=(\boldsymbol{I}+\boldsymbol{N})$
IPC＝3	不完全コレスキー分解	$\boldsymbol{M}=\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$

IPC＝2 のとき，前処理行列も正定値であることが必要です．例えば$(\boldsymbol{I}+\boldsymbol{N})$の対角優位性はそのための十分条件になります．また，行列 ***N*** の固有値の絶対値で最大の値が 1 より大きく前処理行列が ***A*** の逆行列の良い近似にならない場合などでは，前処理を適用しても収束が改善しない可能性があります．

IPC＝3 のとき，OMEGA(0≦OMEGA≦1) を設定する必要があります．

OMEGA＝0 のときは不完全コレスキー分解，OMEGA＝1 のときは修正不完全コレスキー分解です．

楕円型の偏微分方程式の離散化から得られる連立 1 次方程式に対しては，OMEGA＝0.92 から 1.00 が最適であることが経験から分かっています．

IPC＝3 のとき，方程式は前処理の効率化のためにウェーブフロント順に並び換えます．

c. 使用例

n＝51200，対角要素よりの距離が±*int*(*sqrt*(*n*＋0.001)) の所にのみ，非零要素がある正値対称スパース行列の連立 1 次方程式を解きます．

```
C     **EXAMPLE**
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (NW=2,N=51200,K=N+1)
      REAL*8 B(N),X(N),EPS,OMEGA,RZ,
     &       A(K,NW),VW(K*NW+4*N)
      INTEGER ICOL(K,NW),IVW(K*NW+4*N)
      WRITE(6,*) ' EXAMPLE DVCGE '
C INITIALISE A,ICOL
      CALL SET(A,ICOL,K,NW,N)
C GENERATE RHS B
```

```
      DO 10 I=1,N
   10 VW(I)=I
C COMPUTE RHS SO AX=B SO WE KNOW SOLUTION X (X(I)=I)
C
C B = (A-E)*X + E*X
      CALL DVMVSE(A,K,NW,N,ICOL,VW,B,ICON)
      PRINT*,'ERROR CODE =',ICON
      DO 20 I=1,N
      B(I)=B(I)+VW(I)
   20 CONTINUE
C
      ITMAX=4000
      EPS=1D-10
      ISW=1
      IGUSS=0
      DO 30 IPC=1,3
      IF(IPC.EQ.3)OMEGA=0.98
      CALL DVCGE(A,K,NW,N,ICOL,B,IPC,ITMAX,ISW,OMEGA
     &          ,EPS,IGUSS,X,ITER,RZ,VW,IVW,ICON)
C
      PRINT*,'ERROR CODE= ',ICON
      IF(RZ.LE.EPS) WRITE(6,41)'CONVERGED. ACCURACY=',RZ
      IF(RZ.GT.EPS) WRITE(6,41)'FAILED. ACCURACY=',RZ
      WRITE(6,*)'X'
      DO 60 I=1,MIN(N,16),4
   60 WRITE(6,42) I,(X(M),M=I,I+3)
   30 CONTINUE
   42 FORMAT(I3,4(F12.4))
   41 FORMAT(A,2X,E10.3)
      STOP
      END

      SUBROUTINE SET(A,ICOL,K,NW,N)
      INTEGER ICOL(K,NW)
      REAL*8 A(K,NW)
      N3=SQRT(N+0.001)
      DO 10 I=1,NW
      DO 10 J=1,N
      A(J,I)=0.0D0
      ICOL(J,I)=J
   10 CONTINUE
      DO 20 I=1,N-N3
```

```
      A(I,1)=-0.49D0
      ICOL(I,1)=I+N3
   20 CONTINUE
      DO 30 I=N3+1,N
      A(I,2)=-0.49D0
      ICOL(I,2)=I-N3
   30 CONTINUE
      RETURN
      END
```

(4)　手法概要

標準的は共役勾配法（“付録　参考文献一覧表”の [14] 参照）を用いています．不完全コレスキー分解による前処理の詳細は“付録　参考文献一覧表”の [30] を参照してください．また Wavefront ordering によるベクトル化は“付録 参考文献一覧表”の [23] を参照してください．

(5)　謝辞

ITPACK および NSPCG の著者達に, 修正不完全コレスキー分解の前処理および Wavefront ordering のルーチンを利用することを許可していただいたことに感謝いたします．

F17-11-0601 VCPF1,DVCPF1

1 次元素因子離散型複素フーリエ変換
CALL VCPF1(X,N,ISW,ISN,IOUT,Y,W,IW,ICON)

(1)　機能

1 次元複素フーリエ変換の正変換または逆変換を行います.

変換データ長 n は，次の条件を満たす数です.

ー　次の数の中で互いに素な因子の積で表せること.

因子 p($p \in \{2, 3, 4, 5, 7, 8, 9, 16, 25\}$)

a.　1 次元フーリエ変換

$\{x_j\}$を入力し，（1.1）で定義する変換を行い，$\{n\ \alpha_k\}$を求めます.

$$n\alpha_k = \sum_{j=0}^{n-1} x_j \omega_n^{-jk} \quad ,k = 0,1,...,n-1 \qquad ,\omega_n = \exp(2\pi i / n) \tag{1.1}$$

b.　1 次元フーリエ逆変換

$\{\alpha_k\}$を入力し，（1.2）で定義する変換を行い，$\{x_j\}$を求めます.

$$x_j = \sum_{k=0}^{n-1} \alpha_k \omega_n^{jk} \quad ,j = 0,1,...,n-1 \qquad ,\omega_n = \exp(2\pi i / n) \tag{1.2}$$

(2)　パラメタ

X.....入力. 複素数データ. 変換するデータ$\{x_j\}$又は$\{\alpha_k\}$を，X(1:N)に格納します.

出力．複素数データ．変換された結果$\{n\ \alpha_k\}$又は$\{x_j\}$が，X(1:N)に格納されます.

X(N)なる複素数型 1 次元配列.

N.....入力．変換データ長(n).

ISW .....入力．制御情報.

ISW = 1 のとき：初回呼び出し．W，IW に三角関数テーブルおよび制御情報を格納しフーリエ変換を行います.

ISW ≠ 1 のとき：同じ大きさのデータの変換を初回以降引き続き行います. このとき，初回に W，IW に格納された三角関数テーブルおよび制御情報を利用するため，N，ISN，W，IW の内容を初回呼び出し後に変更してはなりません.

出力．ISW=1 を設定して呼び出したとき，ISW=0 が設定されます．このため 2 回目以降特に ISW に値を設定せずに，配列 X に変換データを格納して呼び出すことで効率よく変換を行うことができます.

ISN .....入力．変換か逆変換かを指定.

変換　：ISN=1

逆変換：ISN= –1

IOUT.....出力．変換するデータの大きさ N によって結果が格納される領域が異なるため，どこに格納されるかを示す情報.
IOUT = 1 ：変換した結果は Y(1:N)に格納されます.
IOUT ≠ 1 ：変換した結果は X(1:N)に格納されます.

Y.....出力．IOUT = 1 のとき，変換された複素データが格納されます．配列 X とは別の領域でなければなりません．Y(N)なる複素数型 1 次元配列.

W.....入出力.
ISW = 1 のとき，データ長 N の ISN で指定される変換の三角関数テーブルが格納されます.
ISW ≠ 1 のとき, ISW = 1 を指定した初回呼び出しで作成された三角関数テーブルを入力として利用します.
W(N)なる複素数型 1 次元配列.

IW .....入出力．変換のための制御情報.
ISW = 1 のとき，データ長 N の ISN で指定された変換の制御情報が格納されます.
ISW ≠ 1 のとき，ISW = 1 を指定した初回呼び出しで作成された制御情報を入力として利用します.
大きさ 20 の 4 バイト整数型の 1 次元配列. IW(20).

ICON.....出力．コンディションコード.
表 VCPF1-1 参照.

表 VCPF1-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	—
20000	N が 2，3，4，5，7，8，9，16，25 の中に互いに素な因子の積に分解できなかった.	処理を打ち切る.
20100	2 回目以降の呼び出しで N，ISN の値が初回呼び出しのときの値と異なる.	

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ UFTPR2,UFTPR2I,UFTPR3,UFTPR3I,UFTPR5,UFTPR5I,UFTPRFNL,
UFTPRFNLI,URARNG,UTBL,MGSSL

b.　注意

①　一般的なフーリエ変換の定義

1 次元離散型複素フーリエ変換および，逆変換は一般的に（3.1），(3.2）で定義されます.

$$\alpha_k = \frac{1}{n}\sum_{j=0}^{n-1} x_j \omega_n^{-jk} \quad ,k = 0,1,...,n-1 \tag{3.1}$$

$$x_j = \sum_{k=0}^{n-1} \alpha_k \omega_n^{jk} \quad , j = 0,1...,n-1 \tag{3.2}$$

ここで，$\omega_n = \exp(2\pi i/n)$

本サブルーチンでは，（3.1），(3.2）の左辺に対応して{$n\alpha_k$}または{x_j}を求めます．したがって，結果の正規化は必要に応じて行って下さい．

c.　使用例

1 次元 FFT を計算します．

```
c     **example**
      implicit real*8(a-h,o-z)
      parameter(n=560)
      complex*16 x(n),y(n),w(n)
      integer iw(20)
c
      do i=1,n
      x(i)=i/dble(n)
      enddo
c
c     do the forward transform
c
      isw=1
      call dvcpf1(x,n,isw,1,iout,y,w,iw,icon)
      if(icon.ne.0)then
      print*,'icon = ',icon
      stop
      endif
c
c     do the reverse transform
c
      if(iout.ne.1)then
      isw=1
      call dvcpf1(x,n,isw,-1,iout,y,w,iw,icon)
      if(icon.ne.0)then
      print*,'icon = ',icon
      stop
      endif
      else
      isw=1
      call dvcpf1(y,n,isw,-1,iout,x,w,iw,icon)
      if(icon.ne.0)then
      print*,'icon = ',icon
      stop
```

```
      endif
      endif
c
      tmp=0.0d0
      do i=1,n
      tmp=max(tmp,abs(x(i)/dble(n)-i/dble(n)))
      enddo
c
      print*,'error=',tmp
      stop
      end
```

F16-15-0401 VCPF3,DVCPF3

3 次元素因子離散型複素フーリエ変換
CALL VCPF3(A,B,L,M,N,ISN,VW1,VW2,ICON)

(1)　機能

3 次元（各元の項数 $N1$,$N2$,$N3$）の複素時系列データ $\{x_{J1,J2,J3}\}$ が与えられたとき，離散型複素フーリエ変換，またはその逆変換を素因子フーリエ変換（a prime factor FFT）により行います．各元の項数は，次の条件を満たす数です．

ー次の数の中で互いに素な因子の積で表せること．
　因子 p(p∈ {2,3,4,5,7,8,9,16})

このルーチンで，パラメタ N に 1 を設定して呼び出すと 2 次元の複素素因子高速フーリエ変換になります．さらに，パラメタ N に 1 を，そしてパラメタ M に 1 を設定して呼び出すと 1 次元の複素素因子高速フーリエ変換になります．

① 3 次元複素フーリエ変換

3 次元フーリエ変換は $\{x_{J1,J2,J3}\}$ を入力し（1.1）で定義する変換を行い，$\{N1\times N2\times N3\times \alpha_{K1,K2,K3}\}$ を求めます．

$$N1\times N2\times N3\times \alpha_{K1,K2,K3} = \sum_{J1=0}^{N1-1}\sum_{J2=0}^{N2-1}\sum_{J3=0}^{N3-1} x_{J1,J2,J3}\omega_1^{-J1.K1}\omega_2^{-J2.K2}\omega_3^{-J3.K3} \qquad (1.1)$$

$$,K1 = 0, 1, ..., N1-1$$
$$,K2 = 0, 1, ..., N2-1$$
$$,K3 = 0, 1, ..., N3-1$$
$$,\omega_i = exp(2\pi i/Nj),\ j = 1,2,3$$

② 3 次元複素フーリエ逆変換

3次元フーリエ逆変換は$\{\alpha_{K1,K2,K3}\}$を入力し，(1.2)で定義する変換を行い，$\{x_{J1,J2,J3}\}$ を求めます．

$$x_{J1,J2,J3} = \sum_{K1=0}^{N1-1}\sum_{K2=0}^{N2-1}\sum_{K3=0}^{N3-1}\alpha_{K1,K2,K3}\omega_1^{J1.K1}\omega_2^{J2.K2}\omega_3^{J3.K3} \qquad (1.2)$$

$$,J1 = 0, 1, ..., N1-1$$
$$,J2 = 0, 1, ..., N2-1$$
$$,J3 = 0, 1, ..., N3-1$$
$$,\omega_i = exp(2\pi i/Nj),\ j = 1,2,3$$

(2)　パラメタ

A.....入力．$\{x_{J1,J2,J3}\}$ または，フーリエ変換された $\{\alpha_{K1,K2,K3}\}$ の実部.
出力．フーリエ変換した $\{\alpha_{K1,K2,K3}\}$ または逆変換された $\{x_{J1,J2,J3}\}$ の実部.
A(L,M,N) なる 3 次元配列.
1 次元目，2 次元目，3 次元目のデータ数はそれぞれ L,M,N です.

B.....入力．$\{x_{J1,J2,J3}\}$ または，フーリエ変換された $\{\alpha_{K1,K2,K3}\}$ の虚部.
出力．フーリエ変換した $\{\alpha_{K1,K2,K3}\}$ または，逆変換された $\{x_{J1,J2,J3}\}$ の虚部.
B(L,M,N) なる 3 次元配列.
1 次元目，2 次元目，3 次元目のデータ数はそれぞれ L,M,N です.

L.....入力．1 次元のデータ数.
L≦5040

M.....入力．2 次元のデータ数.
M≦5040

N.....入力．3 次元のデータ数.
N≦5040

ISN.....入力．変換か逆変換かを指定します.
ISN≧0（非負の整数）なら変換.
ISN＜0（負の整数）なら逆変換.

VW1.....作業域．A または B と同じ大きさの 3 次元配列.

VW2.....作業域．A または B と同じ大きさの 3 次元配列.

ICON.....出力．コンディションコード.
表 VCPF3-1 参照.

表 VCPF3-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	―
20000	L,M,N のいずれかが 5040 を超えるか，2,3,4,5,7,8,9,16の中の互いに素な因子の積に分解できなかった.	処理を打ち切る.
30000	L,M,Nのいずれかが0または負であった.	

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ UTRSP,UPFT1,UPFT2,MGSSL

b.　注意

①　一般的な 3 次元フーリエ変換の定義について

3 次元フーリエ変換および，逆変換は一般的に（3.1），（3.2）で定義されます.

$$\alpha_{K1,K2,K3} = \frac{1}{N1\times N2\times N3}\sum_{J1=0}^{N1-1}\sum_{J2=0}^{N2-1}\sum_{J3=0}^{N3-1} x_{J1,J2,J3}\omega_1^{-J1.K1}\omega_2^{-J2.K2}\omega_3^{-J3.K3} \qquad (3.1)$$

$$x_{J1,J2,J3} = \sum_{K1=0}^{N1-1}\sum_{K2=0}^{N2-1}\sum_{K3=0}^{N3-1}\alpha_{K1,K2,K3}\omega_1^{J1.K1}\omega_2^{J2.K2}\omega_3^{J3.K3} \qquad (3.2)$$

本サブルーチンでは，（3.1），（3.2）の左辺に対応して $\{N1\times N2\times N3\times\alpha_{K1,K2,K3}\}$ または，$\{x_{J1,J2,J3}\}$ を求めるので結果の正規化は必要に応じて行ってください．ちなみに，本サブルーチンにより変換・逆変換を正規化せずに連続して実行すると，入力データの各要素は $N1\cdot N2\cdot N3$ 倍されて出力されます．

② 項数について

項数は次の数の中で互いに素な因子の積で表せる数です．

最大値は 5×7×9×16＝5040 です．

因子 $p(p\in\{2,3,4,5,7,8,9,16\})$

③ データの格納方法について

複素数データ $\{x_{J1,J2,J3}\}$，$\{N1\times N2\times N3\times\alpha_{K1,K2,K3}\}$ は，実部と虚部が配列 A,B に別々に格納されます．

c. 使用例

$N1, N2, N3$.項の複素時系列データ $\{x_{J1,J2,J3}\}$ を入力し，フーリエ変換し，その結果をフーリエ逆変換して $\{x_{J1,J2,J3}\}$ を求めます．

ここでは，$N1$＝12, $N2$＝12, $N3$＝12 とします．

```
C     **EXAMPLE**
      DIMENSION A(12,12,12),B(12,12,12),NI(3)
      DIMENSION VW1(12,12,12),VW2(12,12,12)
      DATA NI/12,12,12/,L,M,N/12,12,12/
      READ(5,500) (((A(I,J,K),B(I,J,K),I=1,NI(1)),
     *             J=1,NI(2)),K=1,NI(3))
      WRITE(6,600) (NI(I),I=1,3),
     *             (((I,J,K,A(I,J,K),B(I,J,K),I=1,NI(1)),
     *             J=1,NI(2)),K=1,NI(3))
C     NORMAL TRANSFORM
      CALL VCPF3(A,B,L,M,N,1,VW1,VW2,ICON)
      WRITE(6,610) ICON
      IF(ICON.NE.0) STOP
C     INVERSE TRANSFORM
      CALL VCPF3(A,B,L,M,N,-1,VW1,VW2,ICON)
      NT=NI(1)*NI(2)*NI(3)
      DO 10 K=1,NI(3)
      DO 10 J=1,NI(2)
      DO 10 I=1,NI(1)
```

```
      A(I,J,K)=A(I,J,K)/FLOAT(NT)
      B(I,J,K)=B(I,J,K)/FLOAT(NT)
   10 CONTINUE
      WRITE(6,620) (((I,J,K,A(I,J,K),B(I,J,K),I=1,NI(1)),
     *             J=1,NI(2)),K=1,NI(3))
      STOP
  500 FORMAT(2E20.7)
  600 FORMAT('0',10X,'INPUT DATA',5X,
     * '(',I3,',',I3,',',I3,')'/
     * (15X,'(',I3,',',I3,',',I3,')',
     * 2E20.7))
  610 FORMAT('0',10X,'RESULT ICON=',I5)
  620 FORMAT('0',10X,'OUTPUT DATA'/
     * (15X,'(',I3,',',I3,',',I3,')',
     * 2E20.7))
      END
```

(4)　手法概要

3 次元実フーリエ変換を，基底が互いに素な因子に分解される数を基底とする高速変換手法（aprime factor FFT）により行います．

①　3 次元変換

（1.1）で定義される 3 次元変換は，共通項を整理することにより（4.1）のような順序に変えることができます．*J*1,*J*2,*J*3 に関する和をとる順序を入れ替えることも可能です．

$$N1\times N2\times N3\times \alpha_{K1,K2,K3} = \sum_{J1=0}^{N1-1}\omega_1^{-J1.K1}\sum_{J2=0}^{N2-1}\omega_2^{-J2.K2}\sum_{J3=0}^{N3-1}x_{J1,J2,J3}\omega_3^{-J3.K3} \tag{4.1}$$

（4.1）のΣ_{J3}は *N*3 項の 1 次変換を *N*1×*N*2 組行い，Σ_{J2}は *N*2 項の 1 次変換を *N*1×*N*3 組行い，Σ_{J1}は *N*1 項の 1 次変換を *N*2×*N*3 組行います．

本ルーチンでは各次元に対する 1 次元変換を行うために，互いに素な因子を基底にもつ高速フーリエ変換を適用しています．

②　素因子高速フーリエ変換

3 次元実フーリエ変換は，1 次元のフーリエ変換を複数組 3 回行うことで，計算できます．このとき 1 次元のフーリエ変換を素因子高速フーリエ変換（a prime factor FFT）で行います．

以下に 1 次元の素因子高速フーリエ変換について説明します．

$$c_k = \sum_{J=0}^{N-1} x_J \omega_N^{JK} \qquad \omega_N = exp(-2\pi i/N)$$

$$K = 0, ..., N-1 \tag{4.2}$$

N が互いに素な 2 つの因子 N_1, N_2 に因子分解できたとすると，この 1 次元の高速フーリエ変換は，2 次元の高速フーリエ変換とみなすことができます．
$<>_N$ は N の剰余を，(　,) は最大公約数を表します．
適当な K_i があり，写像（4.3）と（4.4）が決まります．

$$N = N_1 N_2$$
$$j = <K_1 j_1 + K_2 j_2>_N \tag{4.3}$$

$$k = <K_3 k_1 + K_4 k_2>_N \tag{4.4}$$
$$j, k = 0, ..., N-1$$
$$j_1, k_1 = 0, ..., N_1-1$$
$$j_2, k_2 = 0, ..., N_2-1$$

この写像の存在は，次の Chinese remainder Theorem からわかります．
$N_1, N_2, ..., N_k$ が互いに素で,$n_1, n_2, ..., n_k$ を任意の整数とするとき，連立合同式

$$x \equiv n_i \quad (mod\ N_i) \qquad i = 1, ..., k \tag{4.5}$$

の解は，$N = N_1 \times N_2 \times ... \times N_k$ としたとき，N を法として一意的に存在して，

$$x \equiv \sum_{i=0}^{k} n_i M_i q_i \qquad (mod\ N) \tag{4.6}$$

$$M_i = N/N_i, \quad M_i q_i \equiv 1 \qquad (mod\ N_i) \quad i = 1, ..., k \tag{4.7}$$

次に，この写像を利用して,（4.2）の 1 次元の素因子高速フーリエ変換を展開します．

$$X_{j1,j2} = x_{K1j1 + K2j2}$$
$$C_{k1,k2} = c_{K3k1 + K4k2}$$
$$C_{k1,k2} = \sum_{j1=0}^{N1-1} \sum_{j2=0}^{N2-1} X_{j1,j2} \omega_N^{K1K3j1k1} \omega_N^{K1K4j1k2} \omega_N^{K2K3j2k1} \omega_N^{K2K4j2k2} \tag{4.8}$$

次のように K_i を選ぶと（4.2）は，2 次元高速フーリエ変換になります．

$$<K_1 K_3>_N = N_2, <K_2 K_4>_N = N_1$$
$$<K_1 K_4>_N = 0 \quad , <K_2 K_3>_N = 0$$
$$C_{k1,k2} = \sum_{j1=0}^{N1-1} \sum_{j2=0}^{N2-1} X_{j1,j2} \omega_{N1}^{j1k1} \omega_{N2}^{j2k2} \tag{4.9}$$

さらに互いに素な因子に分解されれば，因子の数の次元を持つ多次元のフーリエ変換になります．
分解された因子に対しフーリエ変換は in place に行え，結果として in place なアルゴリズムになります．
次に並び換えについて 2 次元の例で説明します．これは多次元に容易に拡張できます．
$(j_1, j_2),(k_1, k_2)$ は 2 次元の添字に見えますが，写像（4.3）と（4.4）によって一次元の添字に変換されています．

$$j -> (j_1, j_2) : (k_1, k_2) -> k \tag{4.10}$$

次の一般化された Chinese remainder Theorem が成立します．N が互いに素な因子に分解され，次の式が成り立ちます．

$$N = N_1 \times N_2 \times \cdots \times N_m$$
$$n_i = <a_i n>_{Ni} \quad \text{かつ} \quad (a_i, N_i) = 1 \tag{4.11}$$

$$K_i = <N/N_i <(N/N_i)>_{Ni}^{-1} a_i>_N^{-1} \tag{4.12}$$

このとき n は以下のように一意的に表すことができます．

$$n = \sum_i K_i n_i \tag{4.13}$$
$$0 \leqq n \leqq N-1$$
$$0 \leqq n_i \leqq N_i - 1 \qquad i = 1, ..., m$$

（4.3）と（4.4）で定義される写像により，2 次元添字（k_1, k_2）の 1 次元での実際の位置は以下の関係で決まります．

$$j = <K_1 k_1 + K_2 k_2>_N$$
$$k = <K_3 k_1 + K_4 k_2>_N$$

ここに一般化された Chinese remainder Theorem を適用して，2 次元の（k_1, k_2）が写像（4.3）と（4.4）により（4.14）の関係にあることが分かり，2 次元変換したあと計算結果を並び変えることができます．

$$j = <K_1 <a_1 k>_{N1} + K_2 <a_2 k>_{N2}>_N$$
$K_1 = \alpha N_2$, $K_2 = \beta N_1$ の条件を付加して，
$$j = <<K_1 a_1 k>_N + <K_2 a_2 k>_N>_N$$
$$= <(K_1 a_1 + K_2 a_2) k>_N \tag{4.14}$$

並び換えと各因子の基底による高速フーリエ変換について詳しくは，“付録　参考文献一覧表”の [6] , [46] を参照してください．

A72-21-0101 VCRD,DVCRD

非対称または不定値のスパース実行列の連立 1 次方程式 （MGCR 法，対角形式格納法）
CALL VCRD(A,K,NDIAG,N,NOFST,B,ITMAX,EPS,IGUSS,NDIRV,X, ITER,VW,ICON)

(1)　機能

$n \times n$ の非対称／不定値なスパース行列を係数行列とする連立 1 次方程式を修正一般化共役残差法（MGCR:Modified Generalized Conjugate Residuals method）で解きます.

$$\boldsymbol{Ax} = \boldsymbol{b}$$

$n \times n$ の係数行列は，2 つの配列を使用する，対角形式の格納法で格納します.
ベクトル b および x は n 次元ベクトルです.
反復解法の収束および利用指針に関しては，“第 I 部 概説 第 4 章 連立一次方程式の反復解法と収束性”を参照してください.

(2)　パラメタ

A.....入力．係数行列の非零要素を格納します.
A(K,NDIAG) なる 2 次元配列．A(1:N,NDIAG) に係数行列を格納します.
対角形式の格納方法については，“第 I 部 概説 3.2.1.1 一般スパース行列の格納方法 b.一般スパース行列の対角形式の格納方法”を参照してください.

K.....入力．配列 A の整合寸法.

NDIAG.....入力．係数行列 A の非零要素を含む対角ベクトル列の総数.
配列 A の 2 次元目の大きさ.

N.....入力．行列 $\boldsymbol{A}$ の次数 n.

NOFST.....入力．NOFST(NDIAG) なる 1 次元配列．配列 A に格納した対角ベクトルに対応した主対角ベクトルからの距離を格納します. 上対角ベクトル列は正，下対角ベクトル列は負の値で表します.

B.....入力．大きさ n の 1 次元配列．連立 1 次方程式の右辺の定数ベクトルを格納します.

ITMAX .....入力．MGCR 法の反復回数の上限値（＞0）.

EPS.....入力．収束判定に用いられる判定値.
EPS が 0.0 以下のとき，EPS は倍精度ルーチンでは 10^{-6} が，単精度ルーチンでは 10^{-4} が設定されます.
（(3) 使用上の注意 b.注意①参照）.

IGUSS.....入力．配列 X に指定された解ベクトルの近似値から反復計算を行うかを指定する制御情報.
0 のとき　　解ベクトルの近似値を指定しません.
0 以外のとき　配列 X に指定された解ベクトルの近似値から反復計算を開始します.

NDIRV.....入力．MGCR 法で使われる検索方向ベクトルの数（≧1）．
一般に 10 から 100 の小さな数．
X.....入力．大きさ *n* の 1 次元配列．解ベクトルの近似値を指定することができます．
出力．解ベクトルが格納されます．
ITER.....出力．MGCR 法での実際の反復回数．
VW.....作業域．大きさ，N×(NDIRV＋5)＋NDIRV×(NDIRV＋1)なる 1 次元配列．
ICON.....出力．コンディションコード．
表 VCRD-1 参照．

表 VCRD-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし．	―
20001	反復回数の上限に達した	処理を打ち切る．配列 X には，そのときまでに得られている近似値を出力するが精度は保証できない．
30000	N ＜ 1,K ＜ 1,N ＞ K または NDIAG＜1，ITAMAX≦0	処理を打ち切る．
30004	NDIRV＜1	
32001	｜NOFST(I)｜＞N-1	

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ AMACH,URGWD,URIPA,URITI,URITT,URMDG,URMVD,URMGD,
URRCI,URRAN,USSCP,URSTE,URSTI,USVAD,USVCN,USVCP,USVSC,
USVSU,USVUP,USVN1,USVN2,USVNM,URELT,MGSSL

b.　注意

①　MGCR 法は，残差のユーグリッドノルムが最初の残差のユーグリッドノルムと EPS の積以下になったとき収束したと見なします．正確な解と求められた近似解の誤差はほぼ行列 ***A*** の条件数と EPS の積に等しくなります．

②　対角形式を使う上での注意

係数行列 ***A*** の外側の対角ベクトルの要素は零を設定する必要があります．
対角ベクトル列を配列 A に格納する順序に制限は特にありません．
この方法の利点は，行列ベクトル積が間接指標を使わずに計算できる点です．
対角構造を持たない行列は効率よく格納できないという点が欠点です．

c.　使用例

領域［0,1］×［0,1］×［0,1］でディリクレ境界条件（境界で関数値が零）のもとで，“第 I 部 概説 3.2.2 偏微分演算子の離散化とその格納例”に記述された偏微分演算子を離散化して得られる係数行列を持つ連立 1 次方程式を解きます．
INIT_MAT_DIAG は“第 I 部 概説 3.2.2 偏微分演算子の離散化とその格納例”を参照してください．GET_BANDWIDTH_DIAG はバンド幅見積もりのルーチン，INIT_SOL

は，求めるべき解ベクトルを乱数で生成するルーチンです．

```
C     **EXAMPLE**
      PROGRAM TEST_ITER_SOLVERS
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER MACH
      PARAMETER (MACH = 0)
      PARAMETER (K = 10000,NDIRV = 50)
      PARAMETER (NX=20,NY=20,NZ=20,N=NX*NY*NZ)
      PARAMETER (NDIAG = 7,NVW=N*(NDIRV+5)+NDIRV*(NDIRV+1))
      REAL*8 A(K,NDIAG),X(N),B(N),VW(NVW),SOLEX(N)
      INTEGER NOFST(NDIAG)
C
      CALL INIT_SOL(SOLEX,N,1D0,MACH)

      PRINT*,'EXPECTED SOLUSIONS'
      PRINT*,'X(1)= ',SOLEX(1),' X(N)=',SOLEX(N)
C
      PRINT *
      PRINT *,' MGCR METHOD'
      PRINT *,' DIAGONAL FORMAT'
C
      VA1=3D0
      VA2=1D0/3D0
      VA3=5D0
      VC=1.0
      XL=1.0
      YL=1.0
      ZL=1.0
C
      CALL INIT_MAT_DIAG(VA1,VA2,VA3,VC,A,NOFST,
     &     NX,NY,NZ,XL,YL,ZL,NDIAG,N,K)
      CALL GET_BANDWIDTH_DIAG(NOFST,NDIAG,NBANDL,NBANDR)
      DO 110 I = 1,N
      VW(I+NBANDL) = SOLEX(I)
  110 CONTINUE
      CALL DVMVSD(A,K,NDIAG,N,NOFST,NBANDL,VW,B,ICON)
      PRINT*,'DVMVSD ICON= ',ICON
      ITMAX=2000
      IGUSS=0
      EPS = 1D-10
      CALL DVCRD(A,K,NDIAG,N,NOFST,B,ITMAX,EPS,IGUSS,NDIRV,
     &           X,ITER,VW,ICON)
```

```
C
      PRINT* ,'ITER = ',ITER
      PRINT* ,'DVCRD ICON= ',ICON
      PRINT*, 'COMPUTED VALUES'
      PRINT*, 'X(1)= ',X(1),' X(N)= ',X(N)
      STOP
      END
```

(4)　手法概要

MGCR 法は“付録　参考文献一覧表”の [25] を参照してください．アルゴリズムは一般化共役残差法を修正したものです．このアルゴリズムは堅固な解法ですが，常に GMRES 法（“付録　参考文献一覧表”の [35] 参照）よりも高速です．

A72-22-0101 VCRE,DVCRE

非対称または不定値のスパース実行列の連立 1 次方程式 （MGCR 法，ELLPACK 形式格納法）
CALL VCRE(A,K,IWIDT,N,ICOL,B,ITMAX,EPS,IGUSS,NDIRV, X,ITER,VW,ICON)

(1)　機能

$n \times n$ の非対称／不定値なスパース行列を係数行列とする連立 1 次方程式を修正一般化共役残差法（MGCR:Modified Generalized Conjugate Residuals method）で解きます.

$$\boldsymbol{Ax} = \boldsymbol{b}$$

$n \times n$ の係数行列は，2 つの配列を使用する，ELLPACK 形式の格納法で格納します.
ベクトル b および x は n 次元ベクトルです.
反復解法の収束および利用指針に関しては，“第 I 部 概説 第 4 章 連立一次方程式の反復解法と収束性”を参照してください.

(2)　パラメタ

A.....入力．係数行列の非零要素を格納します.
A(K,IWIDT) なる 2 次元配列.
ELLPACK 形式の格納方法については，“第 I 部 概説 3.2.1.1 一般スパース行列の格納方法”を参照してください.

K.....入力．配列 A および配列 ICOL の整合寸法（≧n）.

IWIDT.....入力．係数行列 $\boldsymbol{A}$ の非零要素の行ベクトル方向の最大個数.
ICOL および A の 2 次元目の大きさ.

N.....入力．行列 $\boldsymbol{A}$ の次数 n.

ICOL .....入力．ELLPACK 形式で使用される列指標で，A の対応する要素がいずれの列ベクトルに属すかを示します.
ICOL(K,IWIDT) なる 2 次元配列.

B.....入力．大きさ n の 1 次元配列．連立 1 次方程式の右辺の定数ベクトルを B に格納します.

ITMAX .....入力．MGCR 法の反復回数の上限値（＞0）.

EPS.....入力．収束判定に用いられる判定値.
EPS が 0.0 以下のとき，EPS は倍精度ルーチンでは 10^{-6} が，単精度ルーチンでは 10^{-4} が設定されます.
（(3) 使用上の注意 b.注意①参照）

IGUSS.....入力．配列 X に指定された解ベクトルの近似値から反復計算を開始するかを示す制御情報.
0 のとき　　解ベクトルの近似値を指定しません.
0 以外のとき　配列 X に指定された解ベクトルの近似値から反復計算を開始します.

NDIRV.....入力．MGCR 法で使われる検索方向ベクトルの数（≧1）．
一般に 10 から 100 の小さな数．
X.....入力．大きさ *n* の実数型の 1 次元配列．解ベクトルの近似値を指定することができます．
出力．解ベクトルが格納されます．
ITER.....出力．MGCR 法での実際の反復回数．
VW.....作業域．大きさ，N×(NDIRV＋5)＋NDIRV×(NDIRV＋1)なる 1 次元配列．
ICON.....出力．コンディションコード．
表 VCRE-1 参照．

表 VCRE-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし．	―
20001	最大反復回数に達した．	処理を打ち切る．配列 X には，そのときまでに得られている近似値を出力するが，精度は保証できない．
30000	K＜1 または，IWIDT＜0 または，N＜1,または N＞K,ITMAX≦0	処理を打ち切る．
30004	NDIRV＜1	

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ AMACH,URIPA,URITI,URITT,URMEG,URMVE,URMGE,URRCI,
URRAN,USSCP,URSTE,URSTI,USVAD,USVCN,USVCP,USVSC,USVSU,
USVUP,USVN1,USVN2,USVNM,URELT,MGSSL

b.　注意

① MGCR 法は，残差のユーグリッドノルムが最初の残差のユーグリッドノルムと EPS の積以下になったとき収束したと見なします．
正確な解と求められた近似解の誤差はほぼ行列 ***A*** の条件数と EPS の積に等しいです．

c.　使用例

領域［0,1］×［0,1］×［0,1］でディリクレ境界条件（境界で関数値が零）のもとで，“第 I 部 概説 3.2.2 偏微分演算子の離散化とその格納例”に記述された偏微分演算子を離散化して得られる係数行列を持つ連立 1 次方程式を解きます．
INIT_MAT_ELL は，“第 I 部 概説 3.2.2 偏微分演算子の離散化とその格納例”を参照してください．INIT_SOL は求めるべき解ベクトルを乱数で生成するルーチンです．

```
C     **EXAMPLE**
      PROGRAM TEST_ITER_SOLVERS
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (MACH = 0)
```

```
      PARAMETER (K = 10000,NDIRV = 50)
      PARAMETER (IWIDT = 7,NX=20,NY=20,NZ=20,N=NX*NY*NZ)
      PARAMETER (NVW=N*(NDIRV+5)+NDIRV*(NDIRV+1))
      REAL*8 A(K,IWIDT),X(N),B(N),VW(NVW),SOLEX(N)
      INTEGER ICOL(K,IWIDT)
C
      XL=1.0
      YL=1.0
      ZL=1.0
C
      CALL INIT_SOL(SOLEX,N,1D0,MACH)
      PRINT*,'EXPECTED SOLUTION'
      PRINT*,'X(1)= ',SOLEX(1),' X(N)= ',SOLEX(N)
      PRINT*,'    MGCR METHOD'
      PRINT*,'    ELLPACK FORMAT'
C
      VA1=3D0
      VA2=1D0/3D0
      VA3=5D0
      VC=5D0
C
      CALL INIT_MAT_ELL(VA1,VA2,VA3,VC,A,ICOL,NX,NY,NZ,
     &    XL,YL,ZL,IWIDT,N,K)
      CALL DVMVSE(A,K,IWIDT,N,ICOL,SOLEX,B,ICON)
      PRINT*,'DVMVSE ICON = ',ICON
      IGUSS =0
      EPS = 1D-10
      ITMAX=800
      CALL DVCRE(A,K,IWIDT,N,ICOL,B,ITMAX,EPS,IGUSS,NDIRV
     &     ,X,ITER,VW,ICON)
C
      PRINT*,'DVCRE ICON = ',ICON
      PRINT*,'COMPUTED VALUE'
      PRINT*,'X(1)= ',X(1),' X(N)= ',X(N)
      STOP
      END
```

(4)　手法概要

MGCR 法については“付録　参考文献一覧表”の [25] を参照してください．アルゴリズムは一般化共役残差法を修正したものです．本アルゴリズムは堅固ですが，常に GMRES 法（“付録　参考文献一覧表”の [35] 参照）より高速です．

B71-13-0101 VHEVP,DVHEVP

エルミート行列の固有値・固有ベクトル (三重対角化，マルチセクション法，逆反復法)
CALL　VHEVP(AR,AI,K,N,NF,NL,IVEC,ETOL,CTOL,NEV,E,MAXNE,M,EVR,EVI, VW,IW,ICON)

(1)　機能

n 次のエルミート行列の指定されたいくつかの固有値を求めます．指定に応じて対応する固有ベクトルを計算します．

$$\boldsymbol{Ax} = \lambda \boldsymbol{x} \tag{1.1}$$

(2)　パラメタ

AR.....入力．AR(1:N,1:N)に固有値・固有ベクトルを求めるエルミート行列 ***A*** の実部を格納します．
AR(K,N)なる 2 次元配列．

AI .....入力．AI(1:N,1:N)に固有値・固有ベクトルを求めるエルミート行列 ***A*** の虚部を格納します．
AI(K,N)なる 2 次元配列．

K.....入力．配列 AR 及び AI の整合寸法(≧N)．

N.....入力．エルミート行列 ***A*** の次数 *n*．

NF .....入力．固有値を小さい方から番号付けして（多重固有値には多重度分番号を割り当てる），求める最初の固有値の番号．

NL .....入力．固有値を小さい方から番号付けして（多重固有値には多重度分番号を割り当てる），求める最後の固有値の番号．

IVEC .....入力．制御情報．
1 のとき，固有値および対応する固有ベクトルの両方を求めます．
1 以外のとき，固有値のみ求めます．

ETOL .....入力．固有値が数値的に異なるか多重かを式(3.1)を利用して判定する判定値．倍精度では 3.0D－16（単精度では 2.0D－7）以下のときこの値が標準値として設定されます．
（(3)使用上の注意 b.注意①参照）．

CTOL .....入力．近接している固有値が近似的に多重かを式(3.1)を利用して判定する判定値．近似的多重固有値(cluster)に属する固有値の対応する固有ベクトルの 1 次独立性を保証するために使用されます．CTOL≧ETOL．
CTOL＜ETOL のときは，CTOL＝ETOL が標準値として設定されます．
（(3)使用上の注意 b.注意①参照）．

NEV .....出力．固有値の個数に関する情報．
NEV(1)は，異なる固有値の個数．
NEV(2)は，異なる近似的多重固有値(cluster)の個数．

NEV(3)は，多重度も含んだ全ての固有値の個数.
NEV(3)なる 1 次元配列.

E.....出力．固有値が格納されます.
求められた固有値は E(1:NEV(3))に格納されます.
E(MAXNE)なる 1 次元配列.

MAXNE.....入力．計算できる固有値の最大個数．配列 E の大きさ.
NEV(3)＞MAXNE となったとき，固有ベクトルの計算はできません.
（(3)使用上の注意 b.注意②参照）.

M.....出力．固有値の多重度に関する情報.
M(*i*,1)は *i* 番目の固有値λ_iの多重度を，M(*i*,2)は *i* 番目の固有値λ_iに関して，近接している固有値を近似的多重固有値(cluster)と見なしたときの多重度を示します.
（(3)使用上の注意 b.注意①参照）.
M(MAXNE,2)なる 2 次元配列.

EVR .....出力．IVEC＝1 のとき，おのおのの固有値に対応して固有ベクトルの実部が格納されます.
固有ベクトルは EVR(1:N,1:NEV(3))に格納されます.
EVR(K,MAXNE)なる 2 次元配列.

EVI.....出力．IVEC＝1 のとき，おのおのの固有値に対応して固有ベクトルの虚部が格納されます.
固有ベクトルは EVI(1:N, 1: NEV (3))に格納されます． EVI(K,MAXNE)なる 2 次元配列.

VW.....作業領域．大きさ１７×K なる 1 次元配列.

IW .....作業領域．大きさ９×MAXNE＋１２８なる 1 次元配列.

ICON.....出力．コンディションコード.
表 VHEVP-1 参照

表 VHEVP-1 コンディションコード

コード	意　　味	処理内容
0	エラーなし.	―
20000	密集固有値の計算中，固有値の総数が MAXNE を超えた.	処理を打ち切る．固有ベクトルを求めることはできないが，異なる固有値自体は求められている．（(3)使用上の注意 b.注意②参照）.
30000	NF＜1, NL＞N, NL＜NF, K＜N, N＜1, MAXNE＜NL-NF+1	処理を打ち切る.
30100	入力行列がエルミート行列でない可能性がある.	

(3) 使用上の注意

a. 使用する副プログラム

SSLII・・・ UHEVP, UIBBS, UIBFC, UIBFE, UIBSL, UITBS, UITFC,
UITFE, UITSL, UTDEX, UTDEY, UTMLS, UTRZB, UTRZV,
UZRDM, MGSSL, UMGSL, UMGSL2,

b. 注意

① 本ルーチンは，多重固有値(cluster)に対して特別の考慮を払っています．
ε＝ETOL としたとき，連続する固有値λ_j, $j=s-1,s,...,s+k,(k\geqq 0)$に対して，

$$\frac{\left|\lambda_i-\lambda_{i-1}\right|}{1+\max\left(\left|\lambda_{i-1}\right|,\left|\lambda_i\right|\right)}\leq\varepsilon \qquad (3.1)$$

$i=s,s+1,...,s+k$ となる i に対して(3.1)を満たし，$i=s-1$ および $i=s+k+1$ について(3.1)を満たさないとき，これらの固有値λ_j, j＝s－1,s,...,$s+k$,は数値的に多重であるとみなされます．
ETOL の標準値は倍精度で 3.0D-16（単精度では 2.0D-7）～丸め誤差の単位であり，このとき固有値は計算機で求め得る精度まで分離されます．
(3.1)がε＝ETOL に対して成立しないとき，λ_{i-1} およびλ_i は異なる固有値と考えます．
ε＝ETOL で異なる固有値とみなされた連続する固有値λ_m,$m=t-1,t,...,t+k$,(k≧0)に対して，ε＝CTOL としたとき，$i=t,t+1,...,t+k$ について(3.1)を満たし，$i=t-1$ および $i=t+k+1$ について(3.1)を満たさないとき，これらの異なる固有値λ_m,$m=t-1,t,...,t+k$ を近似的に多重(cluster)と見なします．これは，cluster に対応する不変部分空間, つまり 1 次独立な固有ベクトルを計算するために利用されます．つまり，対応する固有ベクトルは直交する初期ベクトルを用いて計算され，再直交化されます．もちろん，CTOL≧ETOL を満たさなければなりません．この条件を満たさないとき．CTOL は ETOL の値と等しく設定されます．

② r 個数の固有値を求めるとき，最初または最後の固有値が，1 より大きい多重度を持つとき，r 個以上の固有値が求まります．対応する固有ベクトルは増えた固有値も含んで, 対応する固有ベクトルの格納域が足りるときのみ計算されます．
MAXNE に，計算できる固有値の最大個数を指定できます．計算する固有値の総数が MAXNE を超えた場合，ICON＝20000 を返却します．このとき，固有ベクトルの計算を行うよう指定されていたら固有ベクトルの計算はできません．固有値は求められていますが, 多重度分だけ繰り返して格納はされてはいません．つまり，固有値に関しては，求められた異なる固有値が，E(1:NEV(1)) に，および対応する固有値の多重度が M(1:NEV(1),1)にそれぞれ格納されています．
固有値がすべて異なり，近似的多重な固有値もない場合，MAXNE は求める固有値の総数で十分です．計算する固有値の総数が MAXNE を超えた場合，計算を続けるために必要な MAXNE の値は NEV(3)に返却されます．

③ 本ルーチンは，HEIG2 より高速です．

c.　使用例

エルミート行列の指定された固有値・固有ベクトルを求めます.

```
C     ** EXAMPLE PROGRAM **
      IMPLICIT REAL*8(A-H,O-Z)
      PARAMETER (K=512,N=K,NF=1,NL=28,MAXNE=NL-NF+1)
      PARAMETER (NVW=19*K,NIW=9*MAXNE+128)
      REAL*8    AR(K,N),AI(K,N)
      REAL*8    E(MAXNE), EVR(K,MAXNE), EVI(K,MAXNE)
      INTEGER   NEV(3), M(MAXNE,2)
      REAL*8    VW(NVW)
      INTEGER   IW(NIW)
      REAL*8    ETOL, CTOL
      IVEC=1

      ETOL=1.0D-14
      CTOL=5.0D-12

      WRITE (*,*)' Number of data points = ',N
      WRITE (*,*)' Parameter k =',K
      WRITE (*,*)' Eigenvalue calculation tolerance = ',ETOL
      WRITE (*,*)' Cluster tolerance =',CTOL
      WRITE (*,*)' First eigenvalue to be found is ',NF
      WRITE (*,*)' Last eigenvalue to be found is ',NL

C Set up real and imaginary parts of matrix in AR and AI
      DO 100 J=1,N
         DO 98 I=1,N
            AR(I,J) = DBLE(I+J)/DBLE(N)
            IF (I.EQ.J) THEN
               AI(I,J) = 0.0D0
               AR(I,J) = DBLE(J)
             ELSE
                AI(I,J) = DBLE(I*J)/DBLE(N*N)
             ENDIF
 98      CONTINUE
 100  CONTINUE

      DO 99 J=1,N
          DO 99 I=1,N
              IF (I.GT.J) AI(I,J) = -AI(I,J)
 99   CONTINUE
```

```
C Call complex eigensolver

      NNF=NF
      NNL=NL
      CALL DVHEVP(AR,AI,K,N,NNF,NNL,IVEC,ETOL,CTOL,NEV,
     *             E,MAXNE,M,EVR,EVI,VW,IW,ICON)

      WRITE (*,*)' ************************************'
      WRITE (*,*)'      VHEVP OUTPUT'
      IF(ICON.NE.0) THEN
         WRITE (*,*)' Error parameter icon = ',ICON,
     *              ' VHEVP failed'
         GOTO 5000
      ENDIF
      WRITE (*,*)' Number of Hermitian eigenvalues'
      WRITE (*,*) NEV(3)
      WRITE (*,*)' Eigenvaluse of complex Hermitian matrix'
      WRITE (*,*)(E(I),I=1,NEV(3))
 5000 STOP
      END
```

(4)　手法概要

$n \times n$ のエルミート行列 A＝AR＋iAI は，AR＝AR^T,AI＝－AI^T を満たします.

Householder 法でエルミート行列をエルミート 3 重対角行列に変換して，対角ユニタリー変換で実三重対角行列に変換します．Householder 変換の詳細は“付録 参考文献一覧表” [45] または“富士通 SSLII 使用手引書”の TRIDH の記述を参照して下さい.

三重対角行列の固有値・固有ベクトルはマルチセクション法および逆反復法（VTDEV の記述および“付録 参考文献一覧表” [33] 参照）で計算されます.

この固有ベクトルを利用して，エルミート行列の固有ベクトルが計算されます.

B71-11-0101 VLAND,DVLAND

実対称スパース行列の固有値・固有ベクトル（Lanczos 法，対角形式格納法）
CALL VLAND(A,K,NDIAG,N,NOFST,IVEC,IX,EPS,NMIN,NMAX,NLMIN, NLMAX,KR,MAXC,E,INDX,NCMIN,NCMAX,EV,WV,IW, ICON)

(1)　機能

大規模実対称スパース行列 $\boldsymbol{A}$ の最大固有値から少数個または／および最小固有値から少数個の固有値と対応する固有ベクトルを Lanczos 法で求めます．

$$\boldsymbol{Ax} = \lambda \boldsymbol{x} \tag{1.1}$$

大規模実対称スパース行列の最大または最小固有値を求める有効な方法です．

(2)　パラメタ

A.....入力．実対称スパース行列の非零要素を格納します．
実対称係数スパース行列の非零要素を含む対角ベクトルを，一般スパース行列の対角形式格納法を利用して格納します．
A(1:N,1:NDIAG) に係数行列を格納します．
A(K,DIAG) なる 2 次元配列．
対角形式の格納方法については，“第 I 部 概説 3.2.1.1 一般スパース行列の格納方法 b.一般スパース行列の対角形式の格納方法”を参照してください．

K.....入力．配列 A の 1 次元目の大きさ（≧N）．

NDIAG.....入力．係数行列 $\boldsymbol{A}$ の非零要素を含む対角ベクトル列の総数．

N.....入力．行列 $\boldsymbol{A}$ の次数 n．

NOFST.....入力．配列 A に格納される対角ベクトルに対応した主対角ベクトルからの距離を格納します．上対角ベクトル列は正，下対角ベクトル列は負の値で表します．

IVEC.....入力．EV（1:N,1）に指定した初期ベクトルを利用するかどうかを示す制御情報．
1 のとき，EV（1:N,1）に格納されたベクトルを初期ベクトルとして利用します．
1 以外のとき，初期ベクトルを乱数列を生成して構成します．
（(3) 使用上の注意 b.注意①参照）．

IX.....入力．IVEC≠1 の場合，初期ベクトルを乱数列を生成して構成するとき，乱数列を生成するための初期値．1～100000 の任意の整数値を与えます．
（(3) 使用上の注意 b.注意①参照）．

EPS.....入力．計算の途中で得られた固有値・固有ベクトル（λ_i, V_i）が行列 $\boldsymbol{A}$ の固有値・固有ベクトルであるかを判定する判定値．EPS≦0 のとき倍精度は 10^{-6}（単精度は 10^{-3}）が標準値として設定されます．

（(3) 使用上の注意 b.注意③参照）.

NMIN.....入力．求める最小固有値および対応する固有ベクトルの個数（≧0）．小さな数で，NMAX≧1 なら 0 でもよい.

NMAX .....入力．求める最大固有値および対応する固有ベクトルの個数（≧0）．小さな数で，NMIN≧1 なら 0 でもよい.

NLMIN .....入力．最小固有値の計算で使われる数．(≧NMIN).
多くの場合は，2×NMIN で十分です.
（(3) 使用上の注意 b.注意⑤参照）.

NLMAX.....入力．最大固有値の計算で使われる数．(≧NMAX).
多くの場合は，2×NMAX で十分です.
（(3) 使用上の注意 b.注意⑤参照）.

KR.....入力．Lanczos 法で生成されるクリーロフ部分空間の次元数の上限値（≧NLMIN＋NLMAX）．（(3) 使用上の注意 b.注意④参照）.

MAXC.....入力．密集している（cluster）固有値の最大個数．例えば 10.
（(3) 使用上の注意 b.注意②参照）.

E.....出力．E（NEVL）なる 1 次元配列.
求められた最大及び最小固有値が間接指標リスト INDX を使って昇順に格納されています.
NEVL＝NLMIN＋NLMAX.
E（INDX (1:NCMIN)）に最小固有値が格納されています.
E（INDX (NEV－NCMAX:NEV)）に最大固有値が格納されています.
NEV＝NMIN＋NMAX.

INDX .....出力．INDX（NEV）なる 1 次元配列で，配列 E 及び配列 EV の間接指標が格納されます.
固有値 E（INDX(I)）に対応する固有ベクトルは EV（1:N,INDX(I)）に格納されています.
I＝1,..., NEV．ただし NEV＝NMIN＋NMAX.

NCMIN .....出力．求められた最小固有値及び対応する固有ベクトルの個数.

NCMAX.....出力．求められた最大固有値及び対応する固有ベクトルの個数.

EV .....入力．IVEC＝1 のとき，初期ベクトルを EV（1:N,1）に格納します.
出力．計算された固有ベクトルが格納されます．固有値に対して，同様に間接指標リスト INDX を使って引用できます.
EV(K,NEVL)，NEVL＝NLMIN＋NLMAX，なる 2 次元配列.

WV.....作業領域．大きさ (MAXC＋MNL)×(KR＋2)＋MD×(KR＋1)＋7×K＋14×(KR＋1)
の 1 次元配列.
ここで，MNL＝MAX(NLMIN,NLMAX)，MD＝NLMIN＋NLMAX

IW .....作業領域．大きさ 11×(MAXC＋MNL)＋MD＋128 の 1 次元配列.
ここで，MNL＝MAX(NLMIN,NLMAX)，MD＝NLMIN＋NLMAX

ICON.....出力．コンディションコード.
表 VLAND-1 参照.

表 VLAND-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	—
20000	密集固有値の計算中，固有値の総数が MAXC を超えた.	処理を打ち切る.
30000	N＜1，N＞K，NDIAG＜1, IX＜1，IX＞100000, NLMIN＜NMIN, NLMAX＜NMAX, NMIN＜0，NMAX＜0, NMIN＝NMAX＝0	処理を打ち切る.
30004	KR＜NLMIN＋NLMAX	
32001	｜NOFST(I)｜＞N－1	
39001	初期ベクトルが零かまたは零に近い.	
39006	入力行列が対称行列でない.	

(3)　使用上の注意.

a.　使用する副プログラム

SSL II･･･ UZBBM,UZGSD,UZGUD,UZGBD,UZISE,UZLCD,UZLPD,UZMLS,
UZSRZ,UZSTE,UZS3D,UZTDC,UZTDE,USMN1,USSPS,UIBBS,
UIBFC,UIBFE,UIBSL,UITBS,UITFC,UITEE,UITSL,AMACH,URIPA,
URMVD,URPER,URPRE,URPFP,URPIP,UZRDM,USSCP,URSTE,
USVAD,USVCN,USVCP,USVSC,USVSU,USVUP,USVN1,USVN2,
USVNM,MGSSL

b.　注意

① Lanczos 法は確定的な手法ではありません. Householder 変換による三重対角化をベースにした方法ほど堅固な方法ではありません.
Lanczos 法で得た結果は初期ベクトルの選択に大きく依存します．初期ベクトルが求める固有ベクトルの比較的大きな要素を含むとき，よい近似で望む固有値・固有ベクトルが計算されます．これらの要素が小さいか又は無いとき，望む固有値・固有ベクトルは得られないことがあります．しかし，計算された固有値・固有ベクトルは行列 ***A*** の望まれたものとは別の固有値・固有ベクトルの近似となっています.
ほとんどの場合，計算前に初期ベクトルは知られていないので，初期ベクトルは乱数を使って生成されます．初期値 IX を変えて異なる乱数列を生成して初期ベクトルとして本ルーチンを複数回呼び出すことを勧めます.

② 固有値が密集している（cluster）とは，隣り合う固有値の距離が丸め誤差単位のオーダである固有値の集まりのことです.

③ 固有値・固有ベクトルの対（$\lambda_{i,}V_i$）は $\| AV_i - \lambda_i V_i \| \leqq n\varepsilon |\lambda_i|$ を満たすとき行列 ***A*** の固有値・固有ベクトルとして受け入れます．満たないときは，この対は捨てられます.

ここでε=EPS，n=KR，KR はクリーロフ部分空間の次元数.
EPS の値に対する依存性は緩やかです．ただし，EPS が大き過ぎると得られた固有値・固有ベクトルが十分精度の良いものでない可能性があります.

④ KR を大きくすると近似の良い固有値・固有ベクトルが得られますが，より多くのメモリと計算量が必要となります. KRはできるだけ小さな値とするべきです．しかし，KR を N より小さくできない場合（例えば，1 次元ラプラシアンを離散化した行列）があります. 他方, KR を N より大きくすることは勧められません．KR と N が等しいときも本ルーチンは正しく動作しますが, 計算時間は長くなってしまいます.
計算された固有値・固有ベクトルの精度は，かなりクリーロフ部分空間の次元数 KR と初期ベクトルに依存します.

⑤ Lanczos 法では，元の行列 $\boldsymbol{A}$ の固有値・固有ベクトルではない偽り（spurious）の固有値・固有ベクトルが計算されることがあるため，これらを除外します．このため求める固有値・固有ベクトルの個数を多めに計算することが必要になります．NLMIN および NLMAX はこの個数を示します．この大きさは注意して決める必要があります.
ほとんどの場合，NLMIN＝NMIN，NLMAX＝NMAX では不十分です.
NLMIN＝2×NMIN, NLMAX＝2×NMAX程度の大きさで普通の場合は十分です.

c. 使用例

以下の楕円型演算子 $\boldsymbol{L}$ を 3 次元直方体領域で，有限差分法による近似で得られた行列 $\boldsymbol{A}$ に対する最小および最大固有値と対応する固有ベクトルをおのおの 3 個ずつ求めます.

$$Lu = -\Delta u + a\nabla u + u$$

境界では関数値は零．a=(a_1, a_2, a_3) で a_1,a_2 および a_3 は定数.
（行列 $\boldsymbol{A}$ は，mat_diag で生成され対角形式の格納法で格納されます．）

```
C     ** EXAMPLE PROGRAM **
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER REP
      PARAMETER (REP = 2)
      PARAMETER (NX = 20,NY = 20,NZ = 20)
      PARAMETER (K = NX*NY*NZ, N = K)
      PARAMETER (NMAX = 3, NMIN = 3)
      PARAMETER (IVEC=0,IX=123)
      PARAMETER (EPS1 = 1D-6)
      PARAMETER (NLMIN = 2*NMIN, NLMAX = 2*NMAX)
      PARAMETER (MD = NLMIN+NLMAX,NEVL=MD)
      PARAMETER (MNL = NLMIN)    ! MNL = MAX(NLMIN,NLMAX)
      PARAMETER (NEV = NMIN+NMAX)
      PARAMETER (KR = (NX*NY*NZ)/REP)
      PARAMETER (NDIAG = 7)
      PARAMETER (MAXC = 10)
```

```
      PARAMETER (NWV = (MAXC+MNL)*(KR+2)+MD*(KR+1)+
     &                 7*K+14*(KR+1))
      PARAMETER (NIW = 11*(MAXC+MNL)+MD+128)

      REAL*8 A(K,NDIAG),EV(K,NEVL),E(NEVL),VW(NWV)
      INTEGER NOFST(NDIAG),INDX(NEV),IW(NIW)

C     Initialize matrix A
      CALL MAT_DIAG(0D0,0D0,0D0,0D0,2D0,-1D0,A,NOFST,
     &           NX,NY,NZ,NDIAG,K)

      EPS = EPS1

      CALL DVLAND(A,K,NDIAG,N,NOFST,IVEC,IX,EPS,NMIN,
     &           NMAX,NLMIN,NLMAX,KR,MAXC,E,INDX,NCMIN,
     &           NCMAX,EV,VW,IW,ICON)

      IF (ICON.LT. 20000) THEN
         PRINT*,' Real eigenvalues (MIN:MAX)'
         WRITE (*,901) (E(INDX(I)),I=1,NCMIN)
         WRITE (*,901) (E(INDX(I)),I=NEV-NCMAX+1,NEV)
      ENDIF

  901 FORMAT(D23.16)
      STOP
      END
```

(4)　手法概要

Lanczos 法については“付録　参考文献一覧表”の [14]とその文献目録および [8] を参照してください．本ルーチンで使われているアルゴリズムは行列 $\boldsymbol{A}$ の次数より小さくて近いか等しい次数の三重対角行列を生成して，次にこの三重対角行列の固有値・固有ベクトル（Ritz 固有ベクトル）をマルチセクションによるスツルムの方法と逆反復法で求めます（VTDEV の記述参照）．最後に，行列 $\boldsymbol{A}$ の固有ベクトルを Lanczos 法で生成されたクリーロフ部分空間のベクトルと Ritz 固有ベクトルを用いて計算します．

A53-11-0301 VLBX,DVLBX

実バンド行列の連立 1 次方程式（ガウスの消去法）
CALL VLBX(A,N,NH1,NH2,B,EPSZ,ISW,IS,IP,VW,ICON)

(1)　機能

実係数連立 1 次方程式（1.1）をガウス消去で解きます.

$$\boldsymbol{Ax} = \boldsymbol{b} \tag{1.1}$$

ただし，$\boldsymbol{A}$ は $n \times n$，下バンド巾 h_1，上バンド巾 h_2 のバンド行列.

b は n 次元の実定数ベクトル，x は n 次元の解ベクトルです.

$n > h_1 \geqq 0, n > h_2 \geqq 0$ を満たす必要があります.

(2)　パラメタ

A.....入力．バンド係数行列 $\boldsymbol{A}$ を格納する，大きさ $(2 \times h_1 + h_2 + 1) \times n$ の 1 次元配列.

行列 $\boldsymbol{A}$ の格納法については，図 VLBX-1 参照.

出力．LU 分解された $\boldsymbol{L}$ および $\boldsymbol{U}$ が格納されます．格納方法は入力と同じです.

行列 $\boldsymbol{L}$ および $\boldsymbol{U}$ の格納法については，図 VLBX-2 参照.

N.....入力．行列 $\boldsymbol{A}$ の次数 n.

NH1.....入力．行列 $\boldsymbol{A}$ の下バンド巾 h_1.

NH2.....入力．行列 $\boldsymbol{A}$ の上バンド巾 h_2.

B.....入力．定数ベクトル b.

出力．解ベクトル x.

大きさ n の 1 次元配列.

EPSZ.....入力．ピボットの相対零判定値（≧0.0）．0.0 のときは，標準値が選定されます.

（(3) 使用上の注意 b.注意①参照）.

ISW.....入力．制御情報.

同一係数行列を k($k \geqq 1$) 組の方程式を解くとき，次のとおり指定してください.

ISW＝1 のとき　 1 組目の方程式を解きます.

ISW＝2 のとき　 2 組目以降の方程式を解きます.

ただし，このとき B の値だけを新しい定数ベクトル b の値に変え，それ以外のパラメタはそのまま使用してください.

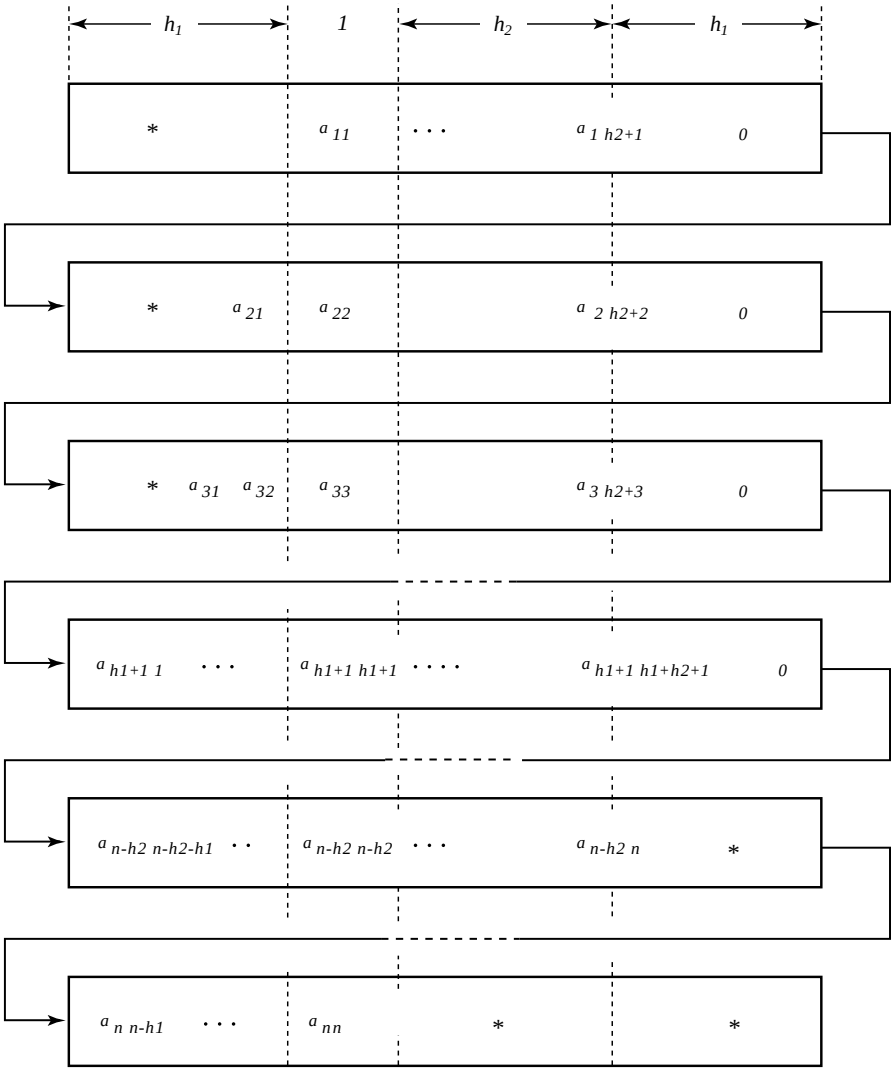
（(3) 使用上の注意 b.注意②参照）.

IS.....出力．行列 $\boldsymbol{A}$ の行列式を求めるための情報.

（(3) 使用上の注意 b.注意③参照）.

IP.....出力. 部分ピボッティングによる行の入替えの履歴を示すトランスポジションベクトル．大きさ n の 1 次元配列.

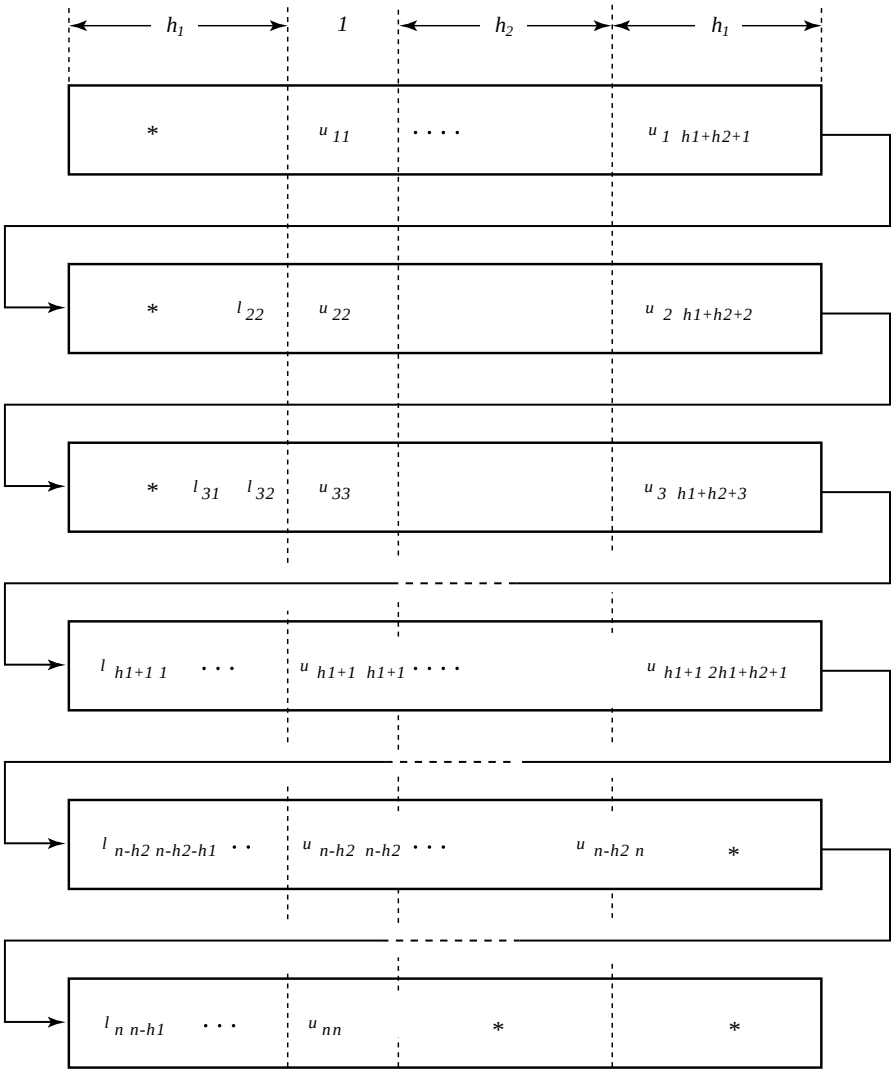
VW.....作業領域．大きさ *n* の 1 次元配列．
ICON.....出力．コンディションコード．
　　表 VLBX-1 参照．



＊ ： 不定値を示します。

図 VLBX-1　配列 A におけるバンド行列の格納方法

係数行列 ***A*** の *i* 番目の行ベクトルは，A((2×h_1＋h_2＋1)×(*i*－1)＋1：(2×h_1＋h_2＋1)×*i*) に連続に格納します．対角要素 a_{ii} が，A((2×h_1＋h_2＋1)×(*i*－1)＋h_1＋1) に格納されるようにします．バンド部分の外側の係数行列の要素には零を設定します．



＊： 不定値を示します。

図 VLBX-2　配列 A における ***L*** および ***U*** の格納方法

行列 ***L*** の対角要素を除いた i 行ベクトルが，A((2×h_1＋h_2＋1)×(i－1)＋1：(2×h_1＋h_2＋1)×(i－1)＋h_1) に格納します．また行列 ***U*** の i 行ベクトルを A((2×h_1＋h_2＋1)×(i－1)＋h_1＋1 : (2×h_1＋h_2＋1)×i) に対角要素から連続に格納します．

表 VLBX-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし．	—
20000	行列 A のある行の要素がすべて零であったか，またはピボットが相対的に零となった．行列 A は非正則の可能性が強い．	処理を打ち切る．
30000	N≦NH1,N≦NH2,NH1＜0,NH2＜0 または EPSZ＜0.0 であった．	処理を打ち切る．

(3)　使用上の注意

a.　使用する副プログラム

SSL II・・・ AMACH,VBLU,VBLUX,MGSSL

b.　注意

① 本サブルーチンでは, EPSZ よりピボットの値が小さいとき相対的に零と見なし, ICON＝20000 として処理を打ち切ります.
EPSZ の標準値は丸め誤差の単位を u としたとき，EPSZ＝16×u です.

② 同一係数行列を持つ連立 1 次方程式をいくつか続けて解く場合には, 2 回目以降 ISW＝2 として解くと，係数行列 $\boldsymbol{A}$ の LU 分解過程を省略するので計算時間が少なくなります．この場合 ISW＝1 のときの値が保障されます.

③ 行列 $\boldsymbol{U}$ の要素は配列 A 上に，図 VBLU-2 で示すとおり格納されます．よって行列式は，n 個の対角要素，即ち $A((2\times h_1+h_2+1)\times(i-1)+h_1+1)$, $i=1, ..., n$ の積と IS の値を掛け合わせて得られます.

④ 本サブルーチンでは，バンド行列の特性を考慮して，データ格納領域を節約できるように, バンド行列を格納しているが, バンド幅の大きさによっては, VALU よりデータ格納領域を多く必要とする場合があります．そのときは，VALU を用いた方がデータ格納領域を節約できます.
本サブルーチンの特性を生かせるのは，$n>2\times h_1+h_2+1$ のときです.

c.　使用例

バンド幅 $h_1=h_2=160$, $n=160\times160$ の非対称バンド行列の連立 1 次方程式を解きます.

```
C     **EXAMPLE**
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER(NH=80)
      PARAMETER(NH1=NH)
      PARAMETER(NH2=NH)
      PARAMETER(N=NH*NH)
      PARAMETER(ALPHA=0.5/(NH1+1),BETA=-ALPHA)
      DIMENSION A((2*NH1+NH2+1)*2*N),B(N)
      DIMENSION C(2*NH1+NH2+1,N),IP(N),VW(N)
      EQUIVALENCE(A,C)
C
C     Zero clear
C
      DO 10 I=1,N*(3*NH+1)
      A(I)=0.0
   10 CONTINUE
C
      DO 15 I=1,N
      B(I)=0.0
      IP(I)=0
   15 CONTINUE
```

```
C
C     Coefficient Matrix is built
C
      DO 20 I=1,N
      C(NH1+1,I)=4.0
      B(I)=B(I)+4.0
      IF(I.GT.NH)THEN
      C(1,I)=-1.0+ALPHA
      B(I)=B(I)-1.0+ALPHA
      ENDIF
      IF(I+NH.LE.N)THEN
      C(1+NH1+NH2,I)=-1.0+BETA
      B(I)=B(I)-1.0+BETA
      ENDIF
      IF(I.GT.1.AND.MOD(I-1,NH).NE.0)THEN
      C(NH1,I)=-1.0+ALPHA
      B(I)=B(I)-1.0+ALPHA
      ENDIF
      IF(I+1.LE.N.AND.MOD(I,NH).NE.0)THEN
      C(NH1+2,I)=-1.0+BETA
      B(I)=B(I)-1.0+BETA
      ENDIF
   20 CONTINUE
C
C     Solve Banded linear equation
C
      EPSZ=0.0D0
      ICON=0
      ISW=1
      CALL DVLBX(A,N,NH1,NH2,B,EPSZ,ISW,IS,IP,VW,ICON)
      PRINT*,'ICON= ',ICON
C
      PRINT*,'B(1)= ',B(1)
      PRINT*,'B(N)= ',B(N)
      STOP
      END
```

(4)　手法概要

外積形式の LU 分解（“付録　参考文献一覧表”の [14] 参照）を行った後，前進代入および後退代入により，

$$\boldsymbol{Ax} = \boldsymbol{b}$$

を解きます.

A22-61-0402 VLDIV, DVLDIV

LDL^T 分解された正値対称行列の逆行列
CALL VLDIV (FA, N, VW, ICON)

(1)　機能

LDL^T 分解された $n \times n$ の正値対称行列 $\boldsymbol{A}$ の逆行列 $\boldsymbol{A}^{-1}$ を求めます.

$$\boldsymbol{A}^{-1} = (\boldsymbol{L}^T)^{-1}\boldsymbol{D}^{-1}\boldsymbol{L}^{-1} \qquad (1.1)$$

ただし，$\boldsymbol{L}$，$\boldsymbol{D}$ はそれぞれ $n \times n$ の下三角行列と対角行列です.

(2)　パラメタ

FA ..... 入力．行列 $\boldsymbol{L}$ と行列 $\boldsymbol{D}^{-1}$.
格納方法については，“FUJITSU SSLII 拡張機能使用手引書”のサブルーチン VSLDL の図 VSLDL-1 参照.
出力．逆行列 $\boldsymbol{A}^{-1}$.
対称行列の格納方法については，“FUJITSU SSLII 拡張機能使用手引書”のサブルーチン VSLDL の図 VSLDL-1 参照. 対称行列の下三角部分が第 1 列から第 n 列まで列ごとに格納されます.
大きさ $n(n+1)/2$ の 1 次元配列.

N..... 入力．行列 $\boldsymbol{L}$，$\boldsymbol{D}$ の次数 n.

VW..... 作業領域．大きさ n の 1 次元配列.

ICON..... 出力．コンディションコード.
表 VLDIV-1 参照.

表 VLDIV-1 コンディションコード

コード	意　　味	処理内容
0	エラーなし.	ー
10000	行列が正値でなかった.	処理は続行する.
30000	N < 1 であった.	処理を打ち切る.

(3)　使用上の注意

a.　使用する副プログラム

SSL II・・・MGSSL

b.　注意

①　本サブルーチンは，LDL^T 分解された正値対称行列を入力して，分解された元の正値対称行列の逆行列を計算します．LDL^T 分解はサブルーチン VSLDL により行い，続けて本サブルーチンを呼び出すことにより逆行列を求めることができます（使用例参照）.

②　連立 1 次方程式を解く場合には，サブルーチン VLSX を用いてください．逆行列を求めて解くことは，VLSX を用いるときよりも演算回数が多くなります．本

サブルーチンは逆行列が必要なときだけ使用します.

c. 使用例

$n \times n$ の正値対称行列の逆行列を求めます.

```
C    **EXAMPLE**
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(55),VW(10)
      DIMENSION VW2(20)
      INTEGER IVW(10)
      N=10
      EPS=0.0D0
      L = 1
      DO J=1,N
         A(l)=N*(N+1)/2-J*(J-1)/2+10.0D0
         L=L+1
         DO I=J+1,N
            A(l)=N+1-I
            L=L+1
         ENDDO
      ENDDO
      WRITE(*,*) 'INPUT MATRIX'
      DO I=1,N
         WRITE(*,1000) (A(((2*N+1-J)*J)/2-N+I),J=1,I)
      ENDDO
      CALL DVSLDL(A,N,EPS,VW2,IVW,ICON)
      IF(ICON.GE.20000)STOP
      CALL DVLDIV(A,N,VW,ICON)
      WRITE(*,*) 'DVLDIV ICON = ',ICON
      WRITE(*,*) 'OUTPUT MATRIX'
      DO I=1,N
         WRITE(*,1000) (A(((2*N+1-J)*J)/2-N+I),J=1,I)
      ENDDO
 1000 FORMAT(X,10E11.3)
      END
```

(4) 手法概要

詳細については，“富士通 SSLII 使用手引書”の LDIV の手法概要，および，“付録参考文献一覧表”の [29] 参照．標準機能の LDIV と格納法は違いますがアルゴリズムは同様です.

A53-31-0301 VLSBX,DVLSBX

正値対称バンド行列の連立 1 次方程式（変形コレスキー分解）
CALL VLSBX(A,N,NH,B,EPSZ,ISW,ICON)

(1)　機能

実係数連立 1 次方程式（1.1）を変形コレスキー法で解きます.

$$\boldsymbol{Ax} = \boldsymbol{b} \tag{1.1}$$

ただし $\boldsymbol{A}$ は $n \times n$，上，下バンド幅 h の正値対称バンド行列．$\boldsymbol{b}$ は n 次元の実定数ベクトル，x は n 次元の解ベクトルです.

$n > h \geqq 0$ を満たさなければなりません.

本ルーチンでは，ベクトル計算機の性能を引き出すために列ベクトル順に格納する格納法を採用しています.

(2)　パラメタ

A.....入力．大きさ $(h+1) \times n$ の 1 次元配列.
係数行列 $\boldsymbol{A}$ の対角要素とバンド行列の下三角部分を格納します.
行列 $\boldsymbol{A}$ の格納法については，図 VLSBX-1 参照.
出力．LDL^T 分解された $\boldsymbol{D}$ および $\boldsymbol{L}$ が格納されます.
行列 $\boldsymbol{L}$ および $\boldsymbol{D}$ の格納法については，図 VLSBX-2 参照.

N.....入力．行列 A の次数 n.

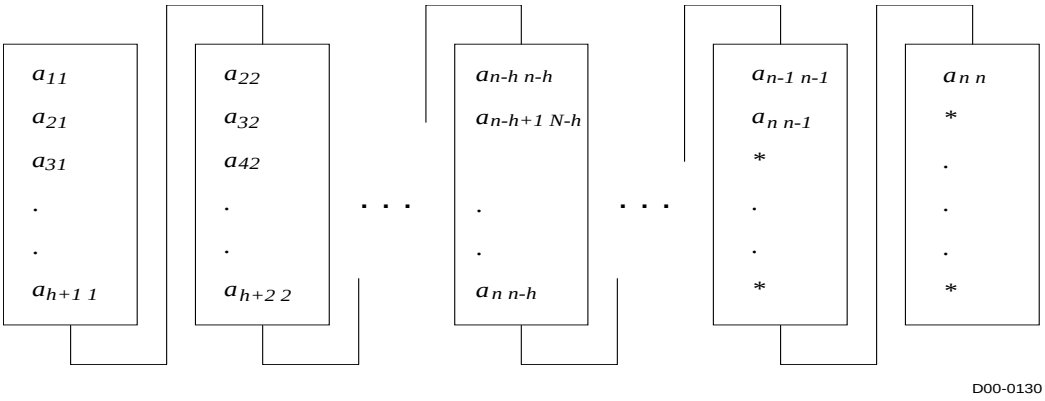
NH.....入力．下バンド幅 h.

B.....入力．定数ベクトル b.
出力．解ベクトル x.
大きさ n の 1 次元配列.

EPSZ .....入力．ピボットの相対零判定値（≧0.0）.
0.0 のときは標準値が採用されます.
（(3) 使用上の注意 b.注意①参照）.

ISW .....入力．制御情報.
同一の係数行列を持つ $k(\geqq 1)$ 組の方程式を解くとき，次のとおり指定してください.
ISW＝1 のとき　1 組目の方程式を解く.
ISW＝2 のとき　2 組目以降の方程式を解く.
ただし，このとき B の値だけを新しい定数ベクトル b の値に変え，それ以外のパラメタはそのまま使ってください.
（(3) 使用上の注意 b.注意②参照）.

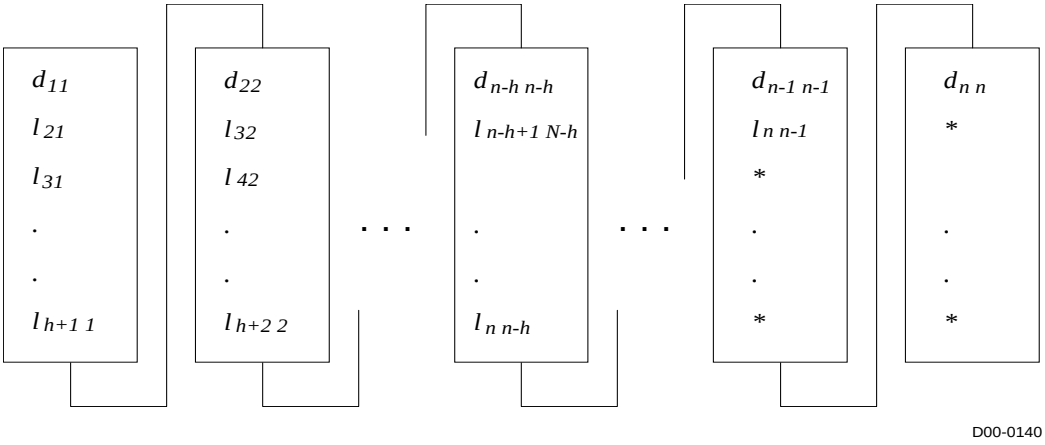
ICON.....出力．コンディションコード.
表 VLSBX-1 参照.



＊：不定値を示します.

図 VLSBX-1　配列 A への行列 *A* の格納方法

A の下バンド行列の *i* 列ベクトルを A((*h*＋1)×(*i*－1)＋*j*－*i*＋1)＝a_{ji}のように格納します. *j* ＝*i*, ..., *i*＋*h*, *i*＝1, ..., *n*



＊：不定値を示します.

図 VLSBX-2　配列 A への行列 ***L*** および行列 ***D*** の格納方法

d_{ii}が A((*h*＋1)×(*i*－1)＋1) に格納されます.
l_{ji}が A((*h*＋1)×(*i*－1)＋*j*－*i*＋1) に格納されます.
j＝*i*＋1, ..., *i*＋*h*, *i*＝1, ..., *n*

表 VLSBX-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	―
10000	ピボットが負となった. 行列 *A* は正値ではない.	処理は続行する.
20000	ピボットが相対的に零となった. 行列 *A* は非正則の可能性が強い.	処理を打ち切る.
30000	NH＜0,NH≧N　又は，EPSZ＜0.0 であった. ISW≠1,2 であった.	

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ AMACH,UBLTS,UBUTS,VBLDL,VBLDX,MGSSL

b.　注意

① 本サブルーチンでは, EPSZ よりピボットの値が小さいとき相対的に零と見なし, ICON＝20000 として処理を打ち切ります.
EPSZ の標準値は丸め誤差の単位を u としたとき，EPSZ＝16×u です.

② 同一係数行列を持つ連立 1 次方程式をいくつか続けて解く場合には，2 回目以降 ISW＝2 として解くと，係数行列 $\boldsymbol{A}$ の LDL^T 分解過程を省略するので計算時間が少なくなります.

③ 分解過程でピボットが負となった場合，係数行列は正値ではありません．本サブルーチンでは，このとき処理を続行されますが ICON＝10000 とします.

④ 行列 $\boldsymbol{L}$ の要素は配列 A 上に，図 VLSBX-2 で示すとおり格納されます．よって行列式は，n 個の対角要素，A((h＋1)×(i－1)＋1), i＝1, ..., n の積を計算することで得られます.

c.　使用例

バンド幅 256 の正値対称行列の連立 1 次方程式を解きます.

```
C     **EXAMPLE**
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER(NH=128)
      PARAMETER(N=128*128)
      DIMENSION A((NH+1)*N),B(N),C(NH+1,N)
      EQUIVALENCE(A,C)
C
C     Zero clear
C
      DO 10 I=1,N*(NH+1)
      A(I)=0.0
   10 CONTINUE
C
      DO 15 I=1,N
      B(I)=0.0
   15 CONTINUE
C
C     Coefficient Matrix is built
C     b = A*y , where y=(1,1,....,1)
C
      DO 20 I=1,N
      C(1,I)=4.0
      B(I)=B(I)+4.0

      IF(I+NH.LE.N)THEN
```

```
      C(NH+1,I)=-1.0
      B(I+NH)=B(I+NH)-1.0
      B(I)=B(I)-1.0
      ENDIF

      IF(I+1.LE.N.AND.MOD(I,NH).NE.0)THEN
      C(2,I)=-1.0
      B(I+1)=B(I+1)-1.0
      B(I)=B(I)-1.0
      ENDIF

   20 CONTINUE
C
C     Solve Symmetric Positive Definite linear equation
C
      EPSZ=0.0D0
      ISW=1
      CALL DVLSBX(A,N,NH,B,EPSZ,ISW,ICON)
      PRINT*,'ICON=',ICON
      IF(ICON.NE.0)STOP
C
      PRINT*,'B(1)= ',B(1)
      PRINT*,'B(N)= ',B(N)
      STOP
      END
```

(4)　手法概要

外積形式の LDL^T 分解（“付録 参考文献一覧表”の [32] 参照）の後，前進代入および後退代入で解きます.

A22-72-0101 VLSPX,DVLSPX

正値対称行列の連立 1 次方程式（ブロック化されたコレスキー分解法）
CALL VLSPX(A,K,N,B,EPSZ,ISW,ICON)

(1)　機能

実係数連立 1 次方程式（1.1）の係数行列 $\boldsymbol{A}$ を，ブロック化された外積形のコレスキー分解法で, (1.2)のように分解して方程式を解きます．ただし $\boldsymbol{A}$ は $n \times n$ の正値対称行列，$\boldsymbol{b}$ は n 次元の実定数ベクトル，$\boldsymbol{x}$ は n 次元の解ベクトル，$\boldsymbol{L}$ は下三角行列とします．また，$n \geqq 1$ とします．

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{1.1}$$

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}} \tag{1.2}$$

(2)　パラメタ

A.....入力．係数行列 A.
入力時には，A(1:N,1:N)の下三角部分$\{A(i, j) \mid i \geqq j\}$に，$\boldsymbol{A}$ の下三角部分$\{a_{ij} \mid i \geqq j\}$を格納します．
出力時には，A(1:N,1:N)の下三角部分$\{A(i, j) \mid i \geqq j\}$に下三角行列 $\boldsymbol{L}\{l_{ij} \mid i \geqq j\}$ が格納されます．(図 VLSPX-1 参照)
A(K,N)なる 2 次元配列.

K.....入力．配列 A の 1 次元目の大きさ．(≧N)

N.....入力．係数行列 $\boldsymbol{A}$ の次数 n.

B.....入力．定数ベクトル $\boldsymbol{b}$.
出力．解ベクトル $\boldsymbol{x}$.
B(N)なる 1 次元配列.

EPSZ .....入力．ピボットの相対零判定値（≧0.0）.
0.0 のときは標準値が採用されます.
（(3) 使用上の注意 b.注意①参照）.

ISW .....入力．制御情報.
同一の係数行列を持つ 複数組の方程式を解くとき, 次のように指定します.
ISW ＝1 のとき，1 組目の方程式を解きます.
ISW ＝2 のとき，2 組目以降の方程式を解きます.
ISW ＝2 のときは，配列 B の値だけを新しい定数ベクトル $\boldsymbol{b}$ の値に変え，それ以外の引数の内容は変更せずに使用して下さい.
（(3) 使用上の注意 b.注意②参照）.

ICON.....出力．コンディションコード.
表 VLSPX-1 参照.

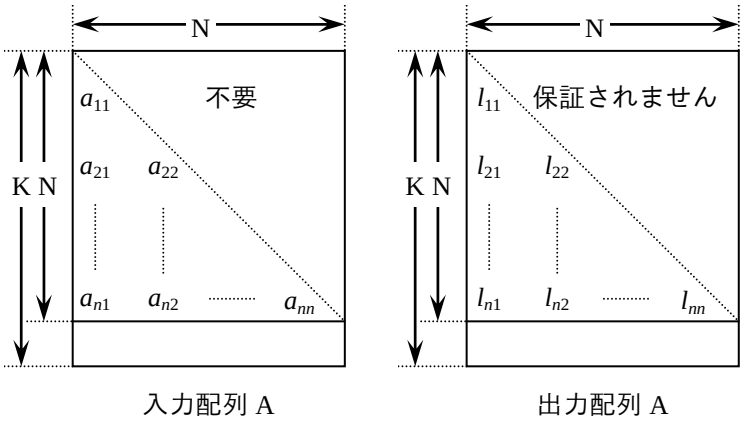


図 VLSPX-1　コレスキー分解法のデータの格納方法

LL^T分解を行う正値対称行列の対角要素および下三角部分 a_{ij} を配列 A(i, j), $i = j,...,n, j=1,...,n$ に格納します.

LL^T分解された結果は，下三角行列 $\boldsymbol{L}$ を下三角部分に格納されます．上三角部分は保証されません.

表 VLSPX-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	ー
20000	ピボットが相対的に零となった．係数行列は非正則の可能性が強い.	処理を打ち切る.
20100	ピボットが負となった．係数行列は正値でない.	
30000	N＜1, EPSZ＜0, K＜N, ISW≠1,2	

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ VSPLL,VSPLX

b.　注意

①　ピボットの相対零判定値にある値を設定したとすると, この値は次の意味を持っています．すなわち，コレスキー分解法による LL^T 分解の過程で，ピボットの値が正でありその値より小さくなった場合に，そのピボットの値を相対的に零と見なし，ICON＝20000 として処理を打ち切ります．EPSZ の標準値は丸め誤差の単位を u としたとき，EPSZ＝16u です.

なお，ピボットの値が小さくなっても，処理を続行させたい場合には，EPSZ に極小の値を設定すればよいのですが，結果の精度は保証されません.

②　同一係数行列を持つ連立1次方程式をいくつか続けて解く場合には, 2回目以降, ISW＝2 として解くと，係数行列 $\boldsymbol{A}$ の LL^T分解過程を省略するので，計算時間を節約できます.

③　分解の途中でピボットが負となった場合，係数行列は正値ではないことになります．このとき，ICON＝20100 として処理を打ち切ります.

④　係数行列の行列式の値を求めたいときは，演算後の配列 A の n 個の対角要素を掛け合わせ，その 2 乗をとって下さい.

c.　使用例

2000×2000 の係数行列の連立 1 次方程式を解きます.

```
C      **EXAMPLE**
       IMPLICIT REAL*8(A-H,O-Z)
       PARAMETER (N=2000,K=N+1)
       REAL*8 A(K,N),B(N)
C
       DO J=1,N
       DO I=J,N
       A(I,J)=MIN(I,J)
       ENDDO
       ENDDO
C
       DO I=1,N
       B(I)=I*(I+1)/2+I*(N-I)
       ENDDO
C
       ISW=1
       CALL DVLSPX(A,K,N,B,1.D-13,ISW,ICON)
       WRITE(6,610) ICON
       IF(ICON.GE.20000) GO TO 100
       WRITE(6,620) (B(I),I=1,10)
C
  100 STOP
  610 FORMAT(1H ,10X,5HICON=,I5)
  620 FORMAT(11X,15HSOLUTION VECTOR
     */(10X,3D23.16))
       END
```

(4)　手法概要

ブロック化された外積形のコレスキー分解法については，“付録 1 参考文献一覧表”の[14]を参照して下さい.

A53-41-0101 VLTQR,DVLTQR

実 3 重対角行列の連立 1 次方程式（QR 分解）
CALL VLTQR(SU,D,SL,N,B,VW,ICON)

(1)　機能

実 3 重対角行列の連立 1 次方程式を QR 分解で解きます.

$$\boldsymbol{Tx} = \boldsymbol{b} \tag{1.1}$$

T は $n \times n$ の正則な実 3 重対角行列．$\boldsymbol{b}$ は n 次元の実定数ベクトル，x は n 次元の解ベクトル．$n \geqq 1$ であること．行列 $\boldsymbol{T}$ の要素を $t_{i,j}$ とすると，対角要素は $d_i = t_{i,i}$ 下副対角要素は $l_i = t_{i,i-1}$，上副対角要素は $u_i = t_{i,i+1}$ です．ただし，$l_1 = u_n = 0$

(2)　パラメタ

SU .....入力．SU（N）なる 1 次元配列で上副対角行列 u_i を SU（1:N-1）に格納します．SU（N）＝0.

D.....入力．D（N）なる 1 次元配列で対角要素 d_i を格納します.

SL.....入力．SL（N）なる 1 次元配列で下副対角行列 l_i を SL（2:N）に格納します．SL（1）＝0.

N.....入力．3 重対角行列 T の次数 n.

B.....入力．定数ベクトル b.
出力．解ベクトル x.

VW.....作業領域．大きさ 7×N なる 1 次元配列.

ICON.....出力．コンディションコード.
表 VLTQR-1 参照.

表 VLTQR-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	―
10000	行列が特異に近い.	
20000	行列が特異である.	処理を打ち切る.
30000	N＜1	

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ UQBBS,UQBFC,UQBFE,UQBSL,UQTBS,UQTFC,UQTFE,UQTSL,
AMACH,MGSSL

b.　使用例

次の 3 重対角行列に関する連立 1 次方程式を解きます.

$$\boldsymbol{T} = \begin{bmatrix} 0 & 1 & & & & \\ -1 & 0 & 1 & & & \\ & -1 & 0 & 1 & & \\ & & \cdot & \cdot & \cdot & \\ & & & -1 & 0 & 1 \\ & & & & -1 & 0 \end{bmatrix}$$

この行列は，対角優位ではありません．ただし n は偶数でなければなりません．さもないと $\boldsymbol{T}$ は特異となります．以下の例では，解の精度をテストするために，解が次の値となるように右辺の定数ベクトルを決めます.

$x_i = (i - 1)/n,\ i = 1, ..., n$

本ルーチンは，行列が特異であるとき，ICON＝20000 を返し，解は求まりません．行列が特異に近いとき, ICON＝10000 を返しますが, 解は正しく求められています.

```
C     ** EXAMPLE PROGRAM **
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER(MAXN=300000)
      REAL*8 SU(MAXN),D(MAXN),SL(MAXN),B(MAXN),X(MAXN)
      REAL*8 VW(7*MAXN)
      REAL*8 ERR
      INTEGER I,ICON

C Data initialization
      N = 256*1024
      DO 9000 I=1,N
        SU(I)=1
        D(I)=0
        SL(I)=-1
        X(I) = (I-1.0)/N
 9000 CONTINUE
      SU(N)=0
      SL(1)=0

C Calculate the right hand side.
      B(1)=X(1)*D(1)+X(2)*SU(1)
      DO 9002 I=2,N-1
        B(I)=SL(I)*X(I-1)+D(I)*X(I)+SU(I)*X(I+1)
 9002 CONTINUE
      B(N)=SL(N)*X(N-1)+D(N)*X(N)
```

```
C Call subroutine
      CALL DVLTQR(SU,D,SL,N,B,VW,ICON)

C Calculate the relative error
      ERR=0.0D0
      DO 9004 I=1,N
        CONTINUE
      IF(X(I).NE.0.AND.B(I).NE.0)THEN
        ERR=MAX(ABS((X(I)-B(I))/X(I)),ERR)
      ELSE
        ERR=MAX(ABS(X(I)-B(I)),ERR)
      ENDIF
 9004 CONTINUE
      WRITE(*,*)'ERROR:',ERR

      END
```

(4)　手法概要

マルチフロンタル法を使って方程式の係数行列を, ブロック 2 重対角形式に簡約します.
方程式を，再帰的な巻き付け（wrap-around）分割アルゴリズムを用いて解きます．未知数の分割に関して，ブロック 2 重対角形式への簡約と再帰的な消去についての制限はありません.
また本方法は，メモリバンクコンフリクトによる性能劣化を起こしません.
本方法で用いている方法は，ハウスホルダーの QR 分解です.
詳細に関しては，“付録　参考文献一覧表”の [14] , [18] を参照してください.

A53-11-0101 VMBV, DVMBV

実バンド行列と実ベクトルの積
CALL VMBV (A, N, NH1, NH2, X, Y, ICON)

(1)　機能

$n \times n$ 下バンド幅 h_1，上バンド幅 h_2 の実バンド行列 $\boldsymbol{A}$ と実ベクトル $\boldsymbol{x}$ の積 $\boldsymbol{y}$ を求めます.

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \tag{1.1}$$

ここで，$\boldsymbol{x}$ と $\boldsymbol{y}$ は n 次元ベクトルです.

$n > h_1 \geq 0$, $n > h_2 \geq 0$ を満たす必要があります.

(2)　パラメタ

A..... 入力．バンド行列 $\boldsymbol{A}$ を格納する，大きさ $(2 \times h_1 + h_2 + 1) \times n$ の 1 次元配列.
格納法については，サブルーチン VLBX の図 VLBX-1 参照.

N ..... 入力．行列 $\boldsymbol{A}$ の次数 n.
（使用上の注意①参照）

NH1 ..... 入力．下バンド幅 h_1 .

NH2 ..... 入力．上バンド幅 h_2.

X ..... 入力．ベクトル $\boldsymbol{x}$.
大きさ の 1 次元配列.

Y ..... 出力．行列 $\boldsymbol{A}$ とベクトル $\boldsymbol{x}$ の積 $\boldsymbol{y}$.
大きさ の 1 次元配列.

ICON ..... 出力．コンディションコード.
表 VMBV-1 参照.

表 VMBV-1 コンディションコード

コード	意　　味	処理内容
0	エラーなし.	ー
30000	N=0, $\|N\| \leq$ NH1, $\|N\| \leq$ NH2, NH1 < 0 又は NH2 < 0 であった.	処理を打ち切る.

(3)　使用上の注意

a.　使用する副プログラム

SSLII ... MGSSL

b.　注意

①　本サブルーチンは通常

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \tag{3.1}$$

の計算をしますが，N= ーN として，任意のベクトル $\boldsymbol{y}'$ をパラメタ Y に与えると

$$\boldsymbol{y} = \boldsymbol{y}' - \boldsymbol{A}\boldsymbol{x}$$

になる計算ができます. 連立 1 次方程式の残差ベクトルの計算等を行う場合に利

用できます.

c. 使用例

行列 $\boldsymbol{A}$ とベクトル $\boldsymbol{x}$ を入力し，ベクトル $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ を求めます.

$n \leq 1000$, $h_1 \leq 100$, $h_2 \leq 100$ の場合.

```
C    **EXAMPLE**
      IMPLICIT  REAL*8  (A-H,O-Z)
      PARAMETER (K=1000,KH1=100,KH2=100)
      DIMENSION A((2*KH1+KH2+1)*K),X(K),Y(K)

      DO 10 I=1,(2*KH1+KH2+1)*K
         A(I)=0.0
 10   CONTINUE

      WRITE(*,*)'INPUT N,NH1,NH2'
      READ(*,*) N,NH1,NH2
      WRITE(*,*)'INPUT A'
      DO 20 I=1,N
         DO 30 J=1,NH1+NH2+1
            IF((J-NH1+(I-1).GE.1).AND.(J-NH1+(I-1).LE.N))THEN
               WRITE(*,*)'A(',I,',',J-NH1+(I-1),')= '
               READ(*,*) A(J+(2*NH1+NH2+1)*(I-1))
            ENDIF
 30      CONTINUE
 20   CONTINUE
      WRITE(*,*)'INPUT X'
      READ(*,*) (X(I),I=1,N)

      CALL DVMBV(A,N,NH1,NH2,X,Y,ICON)
      PRINT*,'ICON= ',ICON
      PRINT*,'Y(1)= ',Y(1)
      PRINT*,'Y(2)= ',Y(2)
      PRINT*,'...'
      PRINT*,'Y(N)= ',Y(N)
      END
```

(4)　手法概要

$n \times n$, 下バンド幅 h_1, 上バンド幅 h_2 のバンド行列 $\boldsymbol{A} = (a_{ij})$と 次元のベクトル $\boldsymbol{x} = (x_i)$ の積 $\boldsymbol{y} = (y_i)$ の要素を(4.1)により計算します.

$$y_i = \sum_{j=1}^{n} a_{ij} x_j, \quad i = 1,...,n \tag{4.1}$$

ただし，本サブルーチンでは，バンド行列であることを考慮して，(4.2)により計算します.

$$y_i = \sum_{j=\max(1,i-h_1)}^{\min(i+h_2,n)} a_{ij} x_j, \quad i = 1,...,n \tag{4.2}$$

F17-12-0101 VMCF2,DVMCF2

1 次元，多重，多次元離散型複素フーリエ変換（複素配列格納，混合基底）
CALL VMCF2(Z,N,M,ISN,ICON)

(1)　機能

1 次元，多重，多次元の離散型複素フーリエ変換を，複素配列を入出力として行います．各次元に対してフーリエ変換を行うかどうか，正変換か逆変換かを指定することができます．

各次元の次数は任意の数が指定可能ですが，2,3,5 の因子に分解される場合が高速です．

① 多次元フーリエ変換

m 次元データ $\{x_{j1j2....jm}\}$ を入力し，(1.1）で定義する変換を行い，$\{\alpha_{klk2....km}\}$ を求めます．

$$\alpha_{k1k2...km} = \sum_{j1=0}^{n1-1}\sum_{j2=0}^{n2-1}\cdots\sum_{jm=0}^{nm-1} x_{j1j2...jm}\cdot\omega_{n1}^{-j1k1r1}\ \ \omega_{n2}^{-j2k2r2}\cdots\omega_{nm}^{-jmkmrm}$$

$$,k1 = 0, 1,..., n1-1 \qquad (1.1)$$

$$,k2 = 0, 1,..., n2-1$$

$$....$$

$$,km = 0, 1,..., nm-1$$

$$,\omega_{n1} = \exp(2\pi i/n1)$$

$$,\omega_{n2} = \exp(2\pi i/n2)$$

$$....$$

$$,\omega_{nm} = \exp(2\pi i/nm)$$

ここで，$n1, n2,...,nm$ は各次元の大きさです．

ri＝1 のとき正変換，ri＝－1 のとき逆変換になります．

例えば r＝(1,1,1) のときは，

$$\alpha_{k1k2k3} = \sum_{j1=0}^{n1-1}\sum_{j2=0}^{n2-1}\sum_{j3=0}^{n3-1} x_{j1j2j3}\cdot\omega_{n1}^{-j1k1}\omega_{n2}^{-j2k2}\omega_{n3}^{-j3k3}$$

という 3 次元正変換になります．

② 多重変換

ri＝0 のとき $\sum_{ji=0}^{ni-1}$ の和はとらず，式 (1.1)の x の対応する添字 ji を ki とします．

例えば 1 次元多重変換では 1 つの和のみになります．3 次元問題の 2 次元目の変換のときは，r＝(0,1,0)と指定することで

$$\alpha_{k1k2k3} = \sum_{j2=0}^{n2-1} x_{k1j2k3}\cdot\omega_{n2}^{-j2k2}$$

が計算されます．

(2)　パラメタ

Z.....入力．$\{x_{j1j2...jm}\}$ ．

出力．$\{\alpha_{k1k2...km}\}$ ．

Z (*n*1, *n*2,･･･, *nm*) なる複素型 M 次元配列．

N.....入力．大きさ M の 1 次元配列．N(*i*)に *i* 次元目の大きさを格納します．

i＝1,...,M

M.....入力．多次元フーリエ変換の次元数 *m*．

ISN.....入力．大きさ M の 1 次元配列．

ISN(*i*)に各次元のフーリエ変換の方向 *ri* を示します．

1　のとき　正変換．

0　のとき　変換なし．

-1　のとき　逆変換．

ICON.....出力．コンディションコード．

表 VMCF2-1 参照．

表 VMCF2-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし．	―
30000	M≦0	処理を打ち切る．
30002	ISN(*i*)＞1 または ISN(*i*)＜-1	
30003	N(*i*)＜1 が指定された	
30004	ISN(*i*)がすべて 0 であった	

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･UZACM,UZCOM,UZFB2,UZFB3,UZFB4,UZFB5, UZFB8, UZFB6,
UZFBL,UZFBR,UZFBS,UZFCT,UZFF2,UZFF3,UZFF4,UZFF5, UZFF8,
UZFF6, UZFMR,UZFOC,UZUPB,UZFPF,UZFRC,UZFRP,UZFS,UZFT,
UZFT2,UZFT3,UZFT5,UZFTB,UZFTF,UZFUB,UZFUF,UZFUS,UZFUW,
UZSCL,UZTR2,UZTRN,UZUNI,UNXRD,UFCT,MGSSL

b.　注意

①　一般的なフーリエ変換の定義

多次元離散型複素フーリエ変換および，逆変換は一般的に（3.1），(3.2）で定義されます．

$$\alpha_{k1k2...km} = \frac{1}{n1n2\cdot\cdot nm} \times \sum_{j1=0}^{n1-1}\sum_{j2=0}^{n2-1}...\sum_{jm=0}^{nm-1} x_{j1j2...jm}\cdot\omega_{n1}^{-j1k1}\omega_{n2}^{-j2k2}...\omega_{nm}^{-jmkm} \qquad (3.1)$$

,*k*1 = 0, 1,..., *n*1－1

,*k*2 = 0, 1,..., *n*2－1

....

,*km* = 0, 1,..., *nm*－1

$$x_{j1j2\ldots jm} = \sum_{k1=0}^{n1-1}\sum_{k2=0}^{n2-1}\cdots\sum_{km=0}^{nm-1}\alpha_{k1k2\ldots km}\cdot\omega_{n1}^{j1k1}\omega_{n2}^{j2k2}\ldots\omega_{nm}^{jmkm} \qquad (3.2)$$

$, j1 = 0, 1,..., n1-1$

$, j2 = 0, 1,..., n2-1$

....

$, jm = 0, 1,..., nm-1$

ここで,

$\omega_{n1} = \exp(2\pi i/n1)$

$,\omega_{n2} = \exp(2\pi i/n2)$

....

$,\omega_{nm} = \exp(2\pi i/nm)$

本サブルーチンでは,（3.1）,（3.2）の左辺に対応して｛$n1\ n2\cdots nm\ \alpha_{k1k2\ldots km}$｝または｛$x_{j1j2\ldots jm}$｝を求めます．従って，結果の正規化は必要に応じて行ってください．

② スタックサイズについて

本サブルーチンでは，内部で使用する作業域を自動割付け配列としてスタックに確保しています．このため，スタックが不足すると異常終了する可能性があります．本サブルーチンが自動割付け配列でスタックに必要とするサイズを以下に示します．

ー *ni* が 2,3,5 の巾乗の積で表せるとき，単精度ルーチンでは

$8\times\max\{ni \mid i = 1, .., m, \text{かつ } \mathrm{ISN}(i)\neq 0\}$ byte

の大きさを必要とします．倍精度ルーチンでは，その倍になります．

ー *ni* の中に 2,3,5 の巾乗の積であらわせない数があるとき，単精度ルーチンで

$40\times\max\{ni \mid i = 1, .., m, \text{かつ } \mathrm{ISN}(i)\neq 0\}$ byte

程度が必要とする大きさの上限になります．倍精度ルーチンでは，その倍になります．

この他に，固定サイズの作業領域や，ユーザープログラムで必要とするスタック領域もあるので，limit コマンドまたは ulimit コマンドなどで十分な大きさのスタックサイズを指定することを薦めます．

c. 使用例

1 次元 FFT を計算します．

```
C     **EXAMPLE**
      INTEGER NMAX
      PARAMETER (NMAX=100000,NDIM=1)
      COMPLEX*16 Z(NMAX)
      REAL*8 ERR,PI,THETA
      INTEGER N(NDIM),ISN(NDIM),N1,L,M,NVAL(6),IN
      DATA NVAL/16199,16200,16201,16383,16384,16385/
```

```
      PI=4D0*ATAN(1D0)
      DO 40 IN=1,6
        N1=NVAL(IN)
        N(1)=N1
        L=79
        DO 10 I=1,N1
          Z(I)=(0D0,0D0)
10    CONTINUE
        Z(L+1)=(1D0,0D0)
        ISN(1)=1
        M=1
        CALL DVMCF2 (Z,N,M,ISN,ICON)
        IF (ICON.NE.0) WRITE (6,*) 'ICON=',ICON
        ERR=0D0
        DO 20 K=0,N1-1
          THETA=2*PI*L*K/DBLE(N1)
          ERR=MAX(ERR,ABS(Z(K+1)-
     &         DCMPLX(COS(THETA),-SIN(THETA))))
20    CONTINUE
        WRITE (6,30) N1,ERR
30    FORMAT (' N=',I6,' ERROR = ',D10.3)
40  CONTINUE
      STOP
      END
```

(4)　手法概要

本ルーチンは，複素フーリエ変換の多重変換，もしくは多次元複素フーリエ変換をスカラ CPU 上で高速に行います．多次元変換は多重 1 次元変換を各次元に対して順次行うことで計算されます．各次元の変換ではその大きさ *ni* に応じて適切な計算手法が選択されます．また，大きな *ni* に対しては，"付録　参考文献一覧表"の[17]の two-sided splitting の変形をブロック化に利用して高速化をはかっています．

F17-11-0101 VMCFT,DVMCFT

1 次元，多重，多次元離散型複素フーリエ変換（実虚別配列，混合基底）
CALL VMCFT(XR,XI,N,M,ISN,W,IW,ICON)

(1)　機能

1 次元，多重，多次元の離散型複素フーリエ変換を行います.

各次元に対してフーリエ変換を行うかどうか，正変換か逆変換かを指定することができます.

各次元の次数は任意の数が可能ですが，2,3,5 の因子に分解される場合が高速です.

①　多次元フーリエ変換

$\{x_{j1j2....jm}\}$ を入力し，(1.1）で定義する変換を行い，$\{\alpha_{klk2....km}\}$ を求めます.

$$\alpha_{k1k2...km} = \sum_{j1=0}^{n1-1}\sum_{j2=0}^{n2-1}\cdots\sum_{jm=0}^{nm-1} x_{j1j2...jm}\cdot\omega_{n1}^{-j1k1r1}\ \ \omega_{n2}^{-j2k2r2}\cdots\omega_{nm}^{-jmkmrm}$$

$$,k_1 = 0, 1,..., n_1-1 \qquad (1.1)$$

$$,k_2 = 0, 1,..., n_2-1$$

$$....$$

$$,k_m = 0, 1,..., n_m-1$$

$$,\omega_{n1} = \exp(2\pi i/n_1)$$

$$,\omega_{n2} = \exp(2\pi i/n_2)$$

$$....$$

$$,\omega_{nm} = \exp(2\pi i/n_m)$$

r_i＝1 のとき正変換，r_i＝－1 のとき逆変換.

r_i＝0 のとき $\sum_{ji=0}^{ni-1}$ の和はとらず，式 (1.1)x の対応する添字 j_i を k_i とします.

r＝(0,1,1) のときは,

$$\alpha_{k1k2k3} = \sum_{j2=0}^{n2-1}\sum_{j3=0}^{n3-1} x_{k1j2j3}\cdot\omega_{n2}^{-j2k2}\omega_{n3}^{j3k3}$$

となります.

②　多重変換

1 次元多重変換は 1 つの和のみを持ちます．2 次元目の変換のときは,

$$\alpha_{k1k2k3} = \sum_{j2=0}^{n2-1} x_{k1j2k3}\cdot\omega_{n2}^{-j2k2}$$

となります.

(2)　パラメタ

XR.....入力.　$\{x_{j1j2...jm}\}$ の実部.

出力.　$\{\alpha_{k1k2...km}\}$ の実部.

大きさ $n_1\times n_2\times\cdots\times n_m$ の 1 次元配列.

XI .....入力.　$\{x_{j1j2...jm}\}$ の虚部.

出力.　$\{\alpha_{k1k2...km}\}$ の虚部.

大きさ $n_1 \times n_2 \times \cdots \times n_m$ の 1 次元配列.

N.....入力. 大きさ M の 1 次元配列. N(I)に I 次元目の大きさ. I＝1,...,M

M.....入力. 多次元フーリエ変換の次元数 m.

ISN.....入力. 大きさ M の 1 次元配列.

ISN(I)に各次元のフーリエ変換の方向 r_i を示します.

1　のとき　正変換.

0　のとき　変換なし.

-1　のとき　逆変換.

W.....作業域.

大きさ IW の 1 次元配列.

IW .....入力. 作業域の大きさ.

n_i が 2,3,5 の巾乗の積で表せるとき，大きさは,

$2 \times MAX\ \{n_i \mid i = 1, .., M,$ かつ $ISN(i) \neq 0\}$

n_i の中に 2,3,5 の巾乗の積であらわせない数があるとき，作業域の大きさは $2 \times n_1 \times ... \times n_m$ を越えます.

このときは IW に 0 を設定して呼出し作業域の大きさを知ることができます.

作業域の大きさを決める手順は“（3）使用上の注意 b.注意②”を参照してください.

出力. 作業域の大きさが必要な量より小さいとき，必要な作業域の大きさを返します.

ICON.....出力. コンディションコード.

表 VMCFT-1 参照.

表 VMCFT-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	ー
30000	M≦0	処理を打ち切る.
30001	作業領域が不足.	
30002	ISN(I)＞1 または ISN(I)＜-1	
30003	N(I)＜1 が指定された	
30004	ISN(I)がすべて 0 であった	

(3)　使用上の注意

a.　使用する副プログラム

SSL II ･･･ UACOM,UCOMR,UFT,UFTBS,UCFS,UCF16,UCFT2,UCFT3,UCFT4,
UCFT5,UCFT8,UCFMR,UCRU,UCTRF,URUNI,USCAL,UTRAN,UTRTW,
UTWID,UGCD,UNXRD,UFCT,MGSSL

b. 注意

① 一般的なフーリエ変換の定義

多次元離散型複素フーリエ変換および，逆変換は一般的に（3.1），（3.2）で定義されます.

$$\alpha_{k1k2\ldots km} = \frac{1}{n_1 n_2 \cdot\cdot n_m} \tag{3.1}$$

$$\times \sum_{j1=0}^{n1-1} \sum_{j2=0}^{n2-1} \cdots \sum_{jm=0}^{nm-1} x_{j1j2\ldots jm} \cdot \omega_{n1}^{-j1k1} \omega_{n2}^{-j2k2} \ldots \omega_{nm}^{-jmkm}$$

$$,k_1 = 0, 1,..., n_1 - 1$$
$$,k_2 = 0, 1,..., n_2 - 1$$
$$....$$
$$,k_m = 0, 1,..., n_m - 1$$

ここで，

$$x_{j1j2\ldots jm} = \tag{3.2}$$
$$\sum_{k1=0}^{n1-1} \sum_{k2=0}^{n2-1} \cdots \sum_{km=0}^{nm-1} \alpha_{k1k2\ldots km} \cdot \omega_{n1}^{j1k1} \omega_{n2}^{j2k2} \ldots \omega_{nm}^{jmkm}$$
$$, j_1 = 0, 1,..., n_1 - 1$$
$$, j_2 = 0, 1,..., n_2 - 1$$
$$....$$
$$, j_m = 0, 1,..., n_m - 1$$
$$\omega_{n1} = \exp(2\pi i/n_1)$$
$$,\omega_{n2} = \exp(2\pi i/n_2)$$
$$....$$
$$,\omega_{nm} = \exp(2\pi i/n_m)$$

本サブルーチンでは，（3.1），（3.2）の左辺に対応して $\{n_1 n_2 \cdot\cdot n_m \alpha_{k1k2\ldots km}\}$ または $\{x_{j1j2\ldots jm}\}$ を求めます．従って，結果の正規化は必要に応じて行ってください.

② 作業域の大きさ

以下の説明で，使う記号を定義します.

RADIX＝ {*n* : 正の整数で 2,3,5 の巾乗の積の形に表せる数}

NORAD＝自然数－*RADIX*

minrad(*n*) は $n < m$ かつ $m \in RADIX$ なる最小の自然数 *m*.

relfac(*n*) は $n = p \times q$ で $p \in RADIX$，$q \in NORAD$ なる最小の自然数 *q*.

$NP = n1 \times n2 \times \ldots \times nm$

このとき作業域の大きさは以下の手順で決まります.

$\{I \mid 1,...,M$ かつ $ISN(I) \neq 0\}$

$\max_{i \in I}(SIZE_i)$ が作業域の大きさです.

$SIZE_i$ は以下のように決まります.

－　$n_i \in$ *RADIX* のとき　$SIZE_i = 2 \times n_i$

－　$relfac(n_i)$が n_i に等しいとき

$$SIZE_i = 2 \times NP \times minrad(n_i)/n_i + 4 \times minrad(n_i)$$

－　その他のとき

$$SIZE_i = 2 \times NP \times minrad(relfac(n_i))/relfac(n_i) + max(4 \times minrad(relfac(n_i)), 2 \times n_i)$$

c.　使用例

1 次元 FFT を計算します.

```
C     **EXAMPLE**
      INTEGER NMAX,NW
      PARAMETER (NMAX=100000,NW=200000)
      REAL*8 XR(NMAX),XI(NMAX),W(NW),PI
      REAL*4 ERR
      INTEGER N(3),ISN(3),IW,N1,L,M,NVAL(6),IN
      DATA NVAL/16199,16200,16201,16383,16384,16385/
      PI=4D0*ATAN(1D0)
      DO 40 IN=1,6
        N1=NVAL(IN)
        N(1)=N1
        L=79
        DO 10 I=1,N1
          XR(I)=0D0
   10     XI(I)=0D0
        XR(L+1)=1D0
        ISN(1)=1
        M=1
        IW=NW
        CALL DVMCFT (XR,XI,N,M,ISN,W,IW,ICON)
        IF (ICON.NE.0) WRITE (6,*) 'ICON=',ICON
        ERR=0D0
        DO 20 K=0,N1-1
          ERR=MAX(ERR,XR(K+1)-COS(2*PI*L*K/DBLE(N1)))
   20     ERR=MAX(ERR,XI(K+1)+SIN(2*PI*L*K/DBLE(N1)))
        WRITE (6,30) N1,ERR
   30   FORMAT (' N=',I6,' ERROR = ',E10.3)
   40 CONTINUE
      STOP
      END
```

(4)　手法概要

本ルーチンは，1 変数の複素フーリエ変換の多重変換，もしくは多次元複素フーリエ変換を行います.

1 変数の変換は次の手順で行われます.

① 変数の次数を $n=pq$ と分解します．ここで p は 2,3 および 5 の巾乗の積の形にあらわせる数で，q は 2,3 および 5 と互いに素な数です．（以下 2,3 および 5 を基数集合と呼ぶ．）

② 次数が $n=pq$ と分解できたら，以下の 4-step algorithm を実行します.

$$z^{(1)}_{j1+k0q} = \sum_{j0=0}^{p-1} \omega_p^{k0\,j0} x_{j1+j0q} \qquad j_1 = 0, ..., q-1, k_0 = 0, ..., p-1 \tag{4.1}$$

$$z^{(2)}_{j1+k0q} = \omega_n^{k0\,j1} z^{(1)}_{j1+k0q} \qquad k_0 = 0, ..., p-1, j_1 = 0, ..., q-1 \tag{4.2}$$

$$z^{(3)}_{k0+j1p} = z^{(2)}_{j1+k0q} \qquad k_0 = 0, ..., p-1, j_1 = 0, ..., q-1 \tag{4.3}$$

$$y_{k0+k1p} = \sum_{j1=0}^{q-1} \omega_q^{k1\,j1} z^{(3)}_{k0+j1p} \qquad k_0 = 0, ..., p-1, k_1 = 0, ..., q-1 \tag{4.4}$$

第 1 および第 4 ステップはそれぞれ次数 p および q の多重フーリエ変換です. 因子 p は基数の巾乗の積の形をしており，混合基底の高速フーリエ変換を使って第 1 ステップを計算します.

このアルゴリズムの詳細は，“付録 参考文献一覧表”の [17] , [19] を参照してください.

この混合基底のアルゴリズムは，より小さい次数の変換，ユニタリースケーリングおよび転置の組み合わせです.

第 2 と第 3 ステップはかなり基本的で，直接的な方法で行われます.

因子 q は基数集合と互いに素であるので, 第 4 ステップは Bluestein's algorithm の変形（“付録 参考文献一覧表”の [41] 参照）を用いて行われます．多次元変換は以上の多重 1 次元変換を各次元に対して順次行うことで計算されます．この処理の中で，ベクトル長が大きくかつ連続アクセスになるように，データは並び替えられますが，最終的には正しい並びで結果を格納します.

F17-13-0101 VMCST,DVMCST

離散型 cosine 変換
CALL VMCST(X,K,N,M,ISW,TAB,ICON)

(1)　機能

1 次元，多重の離散型 cosine 変換を行います.

周期 2π の偶関数 $x(t)$ の半周期を n 等分し，両端を含む n+1 個の標本 $\{x_j\}$

$$x_j = x\left(\frac{\pi}{n} j\right),\ j = 0,1,...,n$$

が与えられたとき，以下で定義する離散型 cosine 変換を配列の各列に対して行います.

$$a_k = x_0 + (-1)^k x_n + 2\sum_{j=1}^{n-1} x_j \cos\frac{\pi}{n} kj,\quad k = 0,1,...,n \tag{1.1}$$

(2)　パラメタ

X.....入力．m 本の $\{x_j\}$, j=0,1,...,n を X(1:N+1, 1:M)に格納します.
　　出力．m 本の $\{a_k\}$, k=0,1,...,n が X(1:N+1, 1:M)に格納されます.
　　X(K,M)なる 2 次元配列.

K.....入力．配列 X の整合寸法．K ≥ N+1

N.....入力．区間の分割数 n．n は偶数．（(3)使用上の注意 b.注意①参照）

M.....入力．変換の多重度 m.

ISW .....入力．制御情報．（(3)使用上の注意 b.注意②参照）
　　0 のとき TAB 設定および cosine 変換.
　　1 のとき TAB 設定のみ.
　　2 のとき設定済みの TAB を使用して cosine 変換.

TAB .....作業域．三角関数表を格納します．TAB(2×N)の 1 次元配列.
　　（(3)使用上の注意 b.注意②参照）

ICON.....出力．コンディションコード.
　　表 VMCST-1 参照.

表 VMCST-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	ー
30000	N≦0, K<N+1, M≦0, ISW ≠ 0,1,2 または N が偶数でなかった.	処理を打ち切る.

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ UMRF2,UZFB2,UZFB3,UZFB4,UZFB5, UZFB8, UZFB6,

UZFF2,UZFF3,UZFF4,UZFF5, UZFF8,UZFF6, UZFPB,UZFPF,
UZFTB,UZFTF, UZUNI,MGSSL

b. 注意

① *n* の値について

区間の分割数 *n* は偶数であれば任意の数が指定可能ですが，2,3,5 の因子に分解される場合が高速です.

② 配列 TAB の再利用について

同じ N で本ルーチンを繰り返し呼び出す場合，初回に ISW=0 または 1 で呼び出した後，2 回目以降は ISW=2 で呼び出すことにより，配列 TAB の作成が省略されて処理が高速化できます.

③ 正規化について

(1.1)式で定義された cosine 変換は，それ自身が逆変換になっています．VMCST ルーチンで二回変換を行うと, 入力データが 2N 倍されたデータが出力されます. 必要に応じて正規化を行ってください.

④ スタックサイズについて

本サブルーチンでは，内部で使用する作業域を自動割付け配列としてスタックに確保しています．このため，スタックが不足すると異常終了する可能性があります．本サブルーチンが自動割付け配列でスタックに必要とするサイズは，単精度ルーチンでは 4×N byte．倍精度ルーチンでは，8×N byte になります. この他に，固定サイズの作業領域や，ユーザープログラムで必要とするスタック領域もあるので，limit コマンドまたは ulimit コマンドなどで十分な大きさのスタックサイズを指定することを薦めます.

c. 使用例

n=1024, 標本点数 1025 の cosine 変換を 5 本計算します.

```
C     **EXAMPLE**
      IMPLICIT REAL*8(A-H,O-Z)
      PARAMETER(N=1024,M=5)
      DIMENSION X(N+1,M),TAB(N*2)

      DO 100 J=1,M
      DO 100 I=1,N+1
      X(I,J)=FLOAT(MAX(I-1,(N-I+1)/J))
  100 CONTINUE

C     FORWARD TRANSFORM
      ISW=0
      CALL DVMCST(X,N+1,N,M,ISW,TAB,ICON)
      PRINT*,'ICON=',ICON

C     BACKWARD TRANSFORM
      ISW=2
```

```
      CALL DVMCST(X,N+1,N,M,ISW,TAB,ICON)
      PRINT*,'ICON=',ICON

      DO 200 J=1,M
      ERROR=0.0D0
      VNRM=0.0D0
      DO 210 I=1,N+1
      ERROR=ERROR+(X(I,J)/(N*2)-
     &    FLOAT(MAX(I-1,(N-I+1)/J)))**2
      VNRM=VNRM+(X(I,J)/(N*2))**2
  210 CONTINUE
      PRINT*,'ERROR=',SQRT(ERROR/VNRM)
  200 CONTINUE
      STOP
      END
```

(4)　手法概要

本ルーチンは，離散型 cosine 変換を実フーリエ変換の問題に帰着させてスカラ CPU 上で高速に計算します．偶関数の対称性を利用することで，効率よく変換が可能であることが知られています．

詳細については“付録 参考文献一覧表”の[26]を参照してください．

F17-12-0201 VMRF2,DVMRF2

1 次元，多重，多次元離散型実フーリエ変換（混合基底）
CALL VMRF2(X,N,M,ISIN,ISN,ICON)

(1)　機能

1 次元，多重，多次元の離散型実フーリエ変換を行います.

各次元に対してフーリエ変換を行わないか，変換を行う場合にその向きを指定することができます.

1 次元目に関しては変換なしを指定できません．また，1 次元目の大きさは偶数でなければなりません．その他の次元の大きさは任意ですが，2，3，5 の巾の積として表される数の場合が高速です.

多重，多次元の離散型実フーリエ変換の結果には複素共役の関係があるため，1 次元目に関して最初の $n1/2+1$ 個の複素数要素を格納します.

a.　多次元実フーリエ変換

①　変換

m 次元実データ $\{x_{j1j2...jm}\}$ を入力し，(1.1）で定義する変換を行い，$\{\alpha_{k1k2....km}\}$ を求めます.

$$\alpha_{k1k2...km} = \sum_{j1=0}^{n1-1}\sum_{j2=0}^{n2-1}\cdots\sum_{jm=0}^{nm-1} x_{j1j2...jm}\cdot\omega_{n1}^{-j1k1r1}\omega_{n2}^{-j2k2r2}\cdots\omega_{nm}^{-jmkmrm} \qquad (1.1)$$

$,k1 = 0, 1,..., n1-1$

$,k2 = 0, 1,..., n2-1$

....

$,km = 0, 1,..., nm-1$

$,\omega_{n1} = \exp(2\pi i/n1)$

$,\omega_{n2} = \exp(2\pi i/n2)$

....

$,\omega_{nm} = \exp(2\pi i/nm)$

ここで，$n1, n2,...,nm$ は各次元の大きさです.

変換の方向は $ri=1$ または $ri=-1$ を指定できます.

例えば，r=（1,1,1）のときの変換は,

$$\alpha_{k1k2k3} = \sum_{j1=0}^{n1-1}\sum_{j2=0}^{n2-1}\sum_{j3=0}^{n3-1} x_{j1j2j3}\cdot\omega_{n1}^{-j1k1}\omega_{n2}^{-j2k2}\omega_{n3}^{-j3k3}$$

という 3 次元離散型実フーリエ正変換になります.

②　逆変換

m 次元データ $\{\alpha_{k1k2....km}\}$ を入力し，(1.2）で定義する変換を行い，$\{x_{j1j2...jm}\}$ を求めます.

$$x_{j1j2\ldots jm} = \sum_{k1=0}^{n1-1}\sum_{k2=0}^{n2-1}\cdots\sum_{km=0}^{nm-1}\alpha_{k1k2\ldots km}\cdot\omega_{n1}^{-j1k1r1}\omega_{n2}^{-j2k2r2}\cdots\omega_{nm}^{-jmkmrm} \quad (1.2)$$

, $j1 = 0, 1, \ldots, n1-1$

, $j2 = 0, 1, \ldots, n2-1$

....

, $jm = 0, 1, \ldots, nm-1$

, $\omega_{n1} = \exp(2\pi i/n1)$

, $\omega_{n2} = \exp(2\pi i/n2)$

....

, $\omega_{nm} = \exp(2\pi i/nm)$

ここで，$n1, n2, \ldots, nm$ は各次元の大きさです．

逆変換では，変換で指定した方向と逆向きを指定します．

$ri=-1$ または $ri=1$

b.　多重変換

$ri=0$ を指定すると，$\sum_{ji=0}^{ni-1}$ の和はとられません．

このとき，正変換では，式（1.1）の x の対応する添字 ji を ki とします．逆変換では，式（1.2）の α の対応する添字 ki を ji とします．

例えば 3 次元配列の 1 次元目の多重変換は r=（1,0,0）と指定することで，

$$\alpha_{k1k2k3} = \sum_{j1=0}^{n1-1} x_{j1k2k3}\cdot\omega_{n1}^{-j1k1}$$

が計算されます．

(2)　パラメタ

X.....X($n1+2, n2, \ldots, nm$) なる M 次元配列．

【ISN＝1（実数から複素数への変換）のとき】

入力．X($1 : n1, 1 : n2, \ldots, 1 : nm$) に実データ $\{x_{j1j2\ldots jm}\}$ を格納します．

出力．X($1 : n1+2, 1 : n2, \ldots, 1 : nm$) に $\{\alpha_{k1k2\ldots km}\}$ の実数成分と虚数成分が交互に格納されます．

$k1 = 0, 1, \ldots, n1/2$

$k2 = 0, 1, \ldots, n2-1$

...

$km = 0, 1, \ldots, nm$

【ISN＝－1（複素数から実数への変換）のとき】

入力．X($1 : n1+2, 1 : n2, \ldots, 1 : nm$) に $\{\alpha_{k1k2\ldots km}\}$ の実数成分と虚数成分を交互に格納します．

$k1 = 0, 1, \ldots, n1/2,$

$k2 = 0, 1, \ldots, n2-1,$

...

$km=0, 1, \ldots, nm-1$

出力．X($1 : n1, 1 : n2, \ldots, 1 : nm$)に実データ $\{x_{j1j2\ldots jm}\}$ が格納されます．

N.....入力．大きさ M の 1 次元配列．N(*i*)(*i*＝1,...,M) に *i* 次元目の大きさ *ni* を格納します． *n*1 は，偶数でなければなりません．

M.....入力．多次元フーリエ変換の次元数 *m*.

ISIN.....入力．大きさ M の 1 次元配列.

ISIN(*i*) に各次元のフーリエ変換の方向 r_i を示します.

ただし，ISIN(1)≠0

1　のとき　*ri*＝1

0　のとき　変換なし

-1　のとき　*ri*＝-1

ISN.....入力.

1 のとき　変換（実数から複素数）

-1 のとき　逆変換（複素数から実数）

ICON.....出力．コンディションコード.

表 VMRF2-1 参照.

表 VMRF2-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	ー
30001	N(*i*)≦0 または M≦0	処理を打ち切る.
30016	ISIN(*i*)＜-1 または ISIN(*i*)＞1 または ISIN(1)＝0	
30032	ISN≠1 かつ ISN≠-1	
30512	N(1)が奇数である.	

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ UMRFF,UMRFB,VMCF2,UZACM,UZCOM,UZFB2,UZFB3,UZFB4,
UZFB5,UZFB8,UZFB6,UZFBL,UZFBR,UZFBS,UZFCT,UZFF2,UZFF3,
UZFF4,UZFF5,UZFF8,UZFF6,UZFMR,UZFOC,UZUPB,UZFPF,UZFRC,
UZFRP,UZFS,UZFT,UZFT2,UZFT3,UZFT5,UZFTB,UZFTF,UZFUB,UZFUF,
UZFUS,UZFUW,UZSCL,UZTR2,UZTRN,UZUNI,UNXRD,UFCT,MGSSL

b.　注意

①　一般的なフーリエ変換の定義

多次元離散型フーリエ変換および，逆変換は一般的に（3.1），(3.2）で定義されます.

$$\alpha_{k1k2\ldots km} = \frac{1}{n1\,n2\ldots nm} \times \sum_{j1=0}^{n1-1} \sum_{j2=0}^{n2-1} \cdots \sum_{jm=0}^{nm-1} x_{j1j2\ldots jm} \cdot \omega_{n1}^{-j1k1} \omega_{n2}^{-j2k2} \ldots \omega_{nm}^{-jmkm} \qquad (3.1)$$

$$,k1 = 0,\ 1,\ldots,\ n1-1$$

$$,k2 = 0,\ 1,\ldots,\ n2-1$$

....

,km = 0, 1,..., nm－1

$$x_{j1j2...jm} = \sum_{k1=0}^{n1-1} \sum_{k2=0}^{n2-1} ... \sum_{km=0}^{nm-1} \alpha_{k1k2...km} \cdot \omega_{n1}^{j1k1} \omega_{n2}^{j2k2} ... \omega_{nm}^{jmkm} \tag{3.2}$$

, $j1$ = 0, 1,..., $n1$－1

, $j2$ = 0, 1,..., $n2$－1

....

, jm = 0, 1,..., nm－1

ここで,　$\omega_{n1} = \exp(2\pi i/n1)$

,$\omega_{n2} = \exp(2\pi i/n2)$

....

,$\omega_{nm} = \exp(2\pi i/nm)$

本サブルーチンでは,(1.1), (1.2)の左辺に対応して$\{n1\ n2 \cdot \cdot nm\ \alpha_{k1k2...km}\}$または$\{x_{j1j2...jm}\}$を求めます．ただし ISIN($i$)＝0 となる i に対しては，ni は 1 とします．結果の正規化は必要に応じて行ってください．

② 多次元離散型実フーリエ変換の結果には，以下の複素共役の関係があります．

$$\alpha_{k1\ k2\ ...\ km} = \overline{\alpha_{n1-k1\ n2-k2\ ...nm-km}}$$

,$k1$ = 0,..., $n1$/2

,$k2$ = 0,..., $n2$－1

....

,km = 1,..., nm－1

ただし，ni－ki は ki＝0 のとき 0 と見なします．

ISIN(h)＝0 となる h に対して，右辺の h 番目の添字は kh のままです．

この関係を使って残りの部分を計算することができます．

③ スタックサイズについて

本サブルーチンでは，内部で使用する作業域を自動割付け配列としてスタックに確保しています．このため，スタックが不足すると異常終了する可能性があります．本サブルーチンが自動割付け配列でスタックに必要とするサイズを以下に示します．

－ ni が 2,3,5 の巾乗の積で表せるとき，単精度ルーチンでは

12×max｛ni｜i = 1, .., M, かつ ISN(i)≠0｝byte

の大きさを必要とします．倍精度ルーチンでは，その倍になります．

－ ni の中に 2,3,5 の巾乗の積であらわせない数があるとき，単精度ルーチンで

40×max｛ni｜i = 1, .., M, かつ ISN(i)≠0｝byte

程度が必要とする大きさの上限になります．倍精度ルーチンでは，その倍になります．

この他に，固定サイズの作業領域や，ユーザープログラムで必要とするスタック領域もあるので，limit コマンドまたは ulimit コマンドなどで十分な大きさのスタックサイズを指定することを薦めます.

c. 使用例

2 次元実フーリエ変換を計算します.

```
C     **EXAMPLE**
      IMPLICIT REAL*8(A-H,O-Z)
      PARAMETER(N1=1024,N2=1024,M=2)
      DIMENSION X(N1+2,N2),N(M),ISIN(M)
C
      DO 100 I=1,N2
      DO 100 J=1,N1
      X(J,I)=FLOAT(J)+FLOAT(N1)*(I-1)
  100 CONTINUE
C
      N(1)=N1
      N(2)=N2
      ISIN(1)=1
      ISIN(2)=1
      ISN= 1
C
C     REAL TO COMPLEX TRANSFORM
C
      CALL DVMRF2(X,N,M,ISIN,ISN,ICON)
      PRINT*,'ICON=',ICON
C
      N(1)=N1
      N(2)=N2
      ISIN(1)=-1
      ISIN(2)=-1
      ISN=-1
C
C     COMPLEX TO REAL TRANSFORM
C
      CALL DVMRF2(X,N,M,ISIN,ISN,ICON)
      PRINT*,'ICON=',ICON

      ERROR=0.0D0
      DO 200 I=1,N2
      DO 200 J=1,N1
      ERROR=MAX(ABS(X(J,I)/(N1*N2)-
```

```
     &    (FLOAT(J)+FLOAT(N1)*(I-1))),ERROR)
  200 CONTINUE
C
      PRINT*,'ERROR=',ERROR
      STOP
      END
```

(4)　手法概要

本ルーチンは，多重および多次元の離散型実フーリエ正逆変換をスカラ CPU 上で高速に行います．

1 次元目の実フーリエ変換に関しては，複素フーリエ変換の問題に帰着させて効率よく計算します．その方法の詳細については，“富士通 SSLII 使用手引書”の RFT ルーチンの手法概要を参照してください．2 次元目以降の変換では，複素型データを対象として，離散型複素フーリエ変換ルーチン VMCF2 が利用されます．

F17-11-0201 VMRFT,DVMRFT

多重，多次元離散型実フーリエ変換（2，3 および 5 の混合基底）
CALL VMRFT(X,N,M,ISIN,ISN,W,ICON)

(1)　機能

多重，多次元の離散型実フーリエ変換を行います.

各次元に対してフーリエ変換を行わないか，変換を行う場合にその向きを指定することができます．変換を行う次元の大きさは 2，3，5 の巾の積として表される数でなければなりません.

1～*m*-1 次元の大きさのうち少なくとも 1 つは偶数でなければなりません．また *m* 次元目に関しては変換なしを指定することはできません.

多重，多次元の離散型実フーリエ変換の結果には複素共役の関係があるため，*m* 番目の次元に関しては最初の $n_m/2+1$ 個を格納します.

a.　多次元実フーリエ変換

①　変換

$\{x_{j1j2...jm}\}$ を入力し,（1.1）で定義する変換を行い，$\{n_1n_2...n_m\alpha_{k1k2....km}\}$ を求めます.

$$n_1n_2...n_m\alpha_{k1k2...km} = \sum_{j1=0}^{n1-1}\sum_{j2=0}^{n2-1}\cdots\sum_{jm=0}^{nm-1} x_{j1j2...jm}\cdot\omega_{n1}^{-j1k1r1}\omega_{n2}^{-j2k2r2}\cdots\omega_{nm}^{-jmkmrm} \tag{1.1}$$

$,k_1 = 0, 1,..., n_1-1$

$,k_2 = 0, 1,..., n_2-1$

....

$,k_m = 0, 1,..., n_m-1$

$,\omega_{n1} = \exp(2\pi i/n_1)$

$,\omega_{n2} = \exp(2\pi i/n_2)$

....

$,\omega_{nm} = \exp(2\pi i/n_m)$

変換の方向は $r_i=1$ または $r_i=-1$ を指定できます.

$r_i=0$ のとき $\sum_{ji=0}^{ni-1}$ の和はとらず，式（1.1）x の対応する添字 j_i を k_i とします.

このとき，式（1.1）の左辺の n_i は 1 に置き換わります.

r=（0,1,1）のときの変換は,

$$n_2n_3\alpha_{k1k2k3} = \sum_{j2=0}^{n2-1}\sum_{j3=0}^{n3-1} x_{k1j2k3}\cdot\omega_{n2}^{-j2k2}\omega_{n3}^{-j3k3}$$

となります.

② 逆変換

$\{\alpha_{k1k2....km}\}$ を入力し，(1.2）で定義する変換を行い，$\{x_{j1j2...jm}\}$ を求めます.

$$x_{j1j2...jm} = \sum_{k1=0}^{n1-1}\sum_{k2=0}^{n2-1}\cdots\sum_{km=0}^{nm-1}\alpha_{k1k2...km}\cdot\omega_{n1}^{-j1k1r1}\omega_{n2}^{-j2k2r2}\cdots\omega_{nm}^{-jmkmrm} \quad (1.2)$$

$, j_1 = 0, 1,..., n_1-1$

$, j_2 = 0, 1,..., n_2-1$

....

$, j_m = 0, 1,..., n_m-1$

$,\omega_{n1} = \exp(2\pi i/n_1)$

$,\omega_{n2} = \exp(2\pi i/n_2)$

....

$,\omega_{nm} = \exp(2\pi i/n_m)$

逆変換では，変換で指定した方向と逆向きを指定しなければなりません.

r_i＝－1 または r_i＝1

r_i＝0 のとき $\sum_{ji=0}^{ni-1}$ の和はとらず，式（1.2）αの対応する添字 k_i を j_i とします.

b. 多重変換

多重変換は 1 つの和のみを持ちます．3 次元の変換のときは,

$$n_3\alpha_{k1k2k3} = \sum_{j3=0}^{n3-1} x_{k1k2j3}\cdot\omega_{n3}^{-j3k3r3}$$

となります.

(2) パラメタ

X.....ISN＝1（実数から複素数への変換）のとき

入力．X(1 : n_1, 1 : n_2, ..., 1 : n_m) に実データ $\{x_{j1j2...jm}\}$ を格納します.

出力．X(1 : n_1, 1 : n_2, ..., 1 : n_m/2＋1) に $\{n_1n_2...n_m\alpha_{k1k2...km}\}$ の実部が，X(1 : n_1, 1 : n_2, ..., n_m/2＋2 : 2×(n_m/2＋1) に $\{n_1n_2...n_m\alpha_{k1k2...km}\}$ の虚部が格納されます.

$k_1 = 0, 1, ..., n_1-1$

$k_2 = 0, 1, ..., n_2-1$

...

$k_m = 0, 1, ..., n_m/2$

ただし，ISIN(i)＝0 の $\{n_1n_2...n_m\alpha_{k1k2...km}\}$ の n_1 は，1 と置き換わります.

ISN＝-1（複素数から実数への変換）のとき

入力．X(1 : n_1, 1 : n_2, ..., 1 : n_m/2＋1) に $\{\alpha_{k1k2...km}\}$ の実部を，X(1 : n_1, 1 : n_2, ..., n_m/2＋2 : 2×(n_m/2＋1)) に $\{\alpha_{k1k2...km}\}$ の虚部を格納します.

$k_1 = 0, 1, ..., n_1-1,$

k_2 = 0, 1, ..., n_2－1,

...

k_m＝0, 1, ..., n_m/2

出力．X(1 : n_1, 1 : n_2, ..., 1 : n_m)に実データ $\{x_{j1j2...jm}\}$ が格納されます．

大きさ　n_1×n_2×...×(2×(n_m/2＋1)) の 1 次元配列．

または　X(n_1,n_2, ...,(2×(n_m/2＋1))) なる m 次元配列．

N.....入力．大きさ M の 1 次元配列．N(I)(I＝1,...,M) に I 次元目の大きさ n_i を格納します．n_i は 2，3，5 の巾乗の積で表せる数．n_i(i＝1,...,M-1) の少なくとも 1 つは，偶数でなければなりません．

M.....入力．多次元フーリエ変換の次元数 m.

ISIN.....入力．大きさ M の 1 次元配列．

ISIN(I) に各次元のフーリエ変換の方向 r_i を示します．

ただし，ISIN(M)≠0

1　のとき　r_i＝1

0　のとき　変換なし

-1　のとき　r_i＝-1

ISN.....入力．

1 のとき　変換（実数から複素数）

-1 のとき　逆変換（複素数から実数）

W.....作業域．

大きさ 2×max(n_1,n_2, ..., n_m)＋n_1×n_2×...×n_{m-1}×(2×(n_m/2＋1)) なる 1 次元配列．

ICON.....出力．コンディションコード．

表 VMRFT-1 参照．

表 VMRFT-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし．	―
30001	N(I)≦0 または M＜2	処理を打ち切る．
30008	ISIN(I)≠0 なる N(I)が 2，3，5 の巾の積であらわされる整数でない．	
30016	ISIN(I)＜-1 または ISIN(I)＞1 または ISIN(M)＝0	
30032	ISN≠1 かつ ISN≠-1	
30512	N(I)(I＝1,...,M-1)がすべて奇数である．	

(3)　使用上の注意

a.　使用する副プログラム

SSL II・・・ UASSM,USEPR,UJOIN,USPLT,UCTRV,UCFS,UCF16,UCFT2,
UCFT3,UCFT4,UCFT5,UCFT8,UCFMRW,UCRU,UCTRF,MGSSL

b.　注意

①　一般的なフーリエ変換の定義

多次元離散型フーリエ変換および，逆変換は一般的に（3.1），(3.2）で定義されます.

$$\alpha_{k1k2\ldots km} = \frac{1}{n_1 n_2 \ldots n_m} \tag{3.1}$$

$$\times \sum_{j1=0}^{n1-1} \sum_{j2=0}^{n2-1} \cdots \sum_{jm=0}^{nm-1} x_{j1j2\ldots jm} \cdot \omega_{n1}^{-j1k1} \omega_{n2}^{-j2k2} \ldots \omega_{nm}^{-jmkm}$$

$$,k_1 = 0, 1,..., n_1 - 1$$
$$,k_2 = 0, 1,..., n_2 - 1$$
$$....$$
$$,k_m = 0, 1,..., n_m - 1$$

$$x_{j1j2\ldots jm} = \tag{3.2}$$

$$\sum_{k1=0}^{n1-1} \sum_{k2=0}^{n2-1} \cdots \sum_{km=0}^{nm-1} \alpha_{k1k2\ldots km} \cdot \omega_{n1}^{j1k1} \omega_{n2}^{j2k2} \ldots \omega_{nm}^{jmkm}$$

$$, j_1 = 0, 1,..., n_1 - 1$$
$$, j_2 = 0, 1,..., n_2 - 1$$
$$....$$
$$, j_m = 0, 1,..., n_m - 1$$

ここで，　$\omega_{n1} = \exp(2\pi i/n_1)$
$$,\omega_{n2} = \exp(2\pi i/n_2)$$
$$....$$
$$,\omega_{nm} = \exp(2\pi i/n_m)$$

本サブルーチンでは,(1.1), (1.2)の左辺に対応して $\{n_1 n_2 \cdots n_m \alpha_{k1k2\ldots km}\}$ または $\{x_{j1j2\ldots jm}\}$ を求めます．ただし ISIN(i)＝0 となる i に対して，n_i を 1 と置き換えます．従って，結果の正規化は必要に応じて行ってください.

②　多次元離散型実フーリエ変換の結果には以下の複素共役の関係があります.

$$\alpha_{k1\ k2\ \ldots\ km} = \overline{\alpha_{n1-k1\ n2-k2\ \ldots nm-km}}$$

$$,k_1 = 0,..., n_1 - 1$$
$$,k_2 = 0,..., n_2 - 1$$

....
,k_m = 1,..., n_m/2

ただし，$n_i - k_i$ は $k_i = 0$ のとき 0 と見なします．

ISIN(h)＝0 となる h に対して，右辺の h 番目の添字は k_h のままです．

この関係を使って残りの部分を計算することができます．

c. 使用例 2 次元実フーリエ変換を計算します．

```
C     **EXAMPLE**
      IMPLICIT REAL*8(A-H,O-Z)
      PARAMETER(N1=1024,N2=1024,M=2)
      PARAMETER(NS=2*(N2/2+1))
      DIMENSION X(N1,NS),N(M),W(2*N1+N1*NS),ISIN(M)
C
      DO 100 I=1,N2
      DO 100 J=1,N1
      X(J,I)=FLOAT(J)+FLOAT(N1)*(I-1)
  100 CONTINUE
C
      N(1)=N1
      N(2)=N2
      ISIN(1)=1
      ISIN(2)=1
      ISN= 1
C
C     REAL TO COMPLEX TRANSFORM
C
      CALL DVMRFT(X,N,M,ISIN,ISN,W,ICON)
      PRINT*,'ICON=',ICON
C
      N(1)=N1
      N(2)=N2
      ISIN(1)=-1
      ISIN(2)=-1
      ISN=-1
C
C     COMPLEX TO REAL TRANSFORM
C
      CALL DVMRFT(X,N,M,ISIN,ISN,W,ICON)
      PRINT*,'ICON=',ICON

      ERROR=0.0D0
      DO 200 I=1,N2
```

```
      DO 200 J=1,N1
      ERROR=MAX(ABS(X(J,I)/(N1*N2)-
     &    (FLOAT(J)+FLOAT(N1)*(I-1))),ERROR)
  200 CONTINUE
C
      PRINT*,'ERROR=',ERROR
      STOP
      END
```

F17-13-0201 VMSNT,DVMSNT

離散型 sine 変換
CALL VMSNT(X,K,N,M,ISW,TAB,ICON)

(1)　機能

1 次元，多重の離散型サイン変換を行います.

周期 2π の奇関数 $x(t)$ の半周期を n 等分し，零とみなされる両端の値を含まない，

n–1 個の標本 $\{x_j\}$

$$x_j = x\left(\frac{\pi}{n} j\right), \quad j = 1,2,...,n-1$$

が与えられたとき，以下で定義する離散型 sine 変換を配列の各列に対して行います.

$$a_k = 2\sum_{j=1}^{n-1} x_j \sin\frac{\pi}{n} kj, \quad k = 1,2,...,n-1 \tag{1.1}$$

(2)　パラメタ

X.....入力．m 本の $\{x_j\}$ を X(1:N－1,1:M)に格納します.

出力．m 本の $\{a_k\}$ が X(1:N－1,1:M)に格納されます.

X(K, M)なる 2 次元配列.

K.....入力．配列 X の整合寸法．K ≥ N－1

N.....入力．区間の分割数 n．n は偶数．（(3)使用上の注意 b.注意①参照）

M.....入力．変換の多重度 m.

ISW .....入力．制御情報．（(3)使用上の注意 b.注意②参照）

0 のとき TAB 設定および sine 変換.

1 のとき TAB 設定のみ.

2 のとき設定済みの TAB を使用して sine 変換.

TAB .....出力．三角関数表を格納します．TAB(2×N)の 1 次元配列.

（(3)使用上の注意 b.注意②参照）

ICON.....出力．コンディションコード.

表 VMSNT-1 参照.

表 VMSNT-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	ー
30000	N≦0, K<N－1, M≦0, ISW ≠ 0,1,2 または N が偶数でなかった.	処理を打ち切る.

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ UMRF2,UZFB2,UZFB3,UZFB4,UZFB5, UZFB8, UZFB6,
UZFF2,UZFF3,UZFF4,UZFF5, UZFF8,UZFF6, UZFPB,UZFPF,
UZFTB,UZFTF, UZUNI,MGSSL

b.　注意

①　*n* の値について
区間の分割数 *n* は偶数であれば任意の数が指定可能ですが，2,3,5 の因子に分解される場合が高速です.

②　TAB の再利用について
同じ N で本ルーチンを繰り返し呼び出す場合，初回に ISW=0 または 1 で呼び出した後，2 回目以降は ISW=2 で呼び出すことにより，配列 TAB の作成が省略されて処理が高速化できます.

③　正規化について
(1.1)式で定義された sine 変換は, それ自身が逆変換になっています. VMSNT ルーチンで二回変換を行うと，入力データが 2N 倍されたデータが出力されます．必要に応じて正規化を行ってください.

④　スタックサイズについて
本サブルーチンでは，内部で使用する作業域を自動割付け配列としてスタックに確保しています．このため，スタックが不足すると異常終了する可能性があります．本サブルーチンが自動割付け配列でスタックに必要とするサイズは，単精度ルーチンでは 8×N byte ．倍精度ルーチンでは，16×N byte になります．この他に，固定サイズの作業領域や，ユーザープログラムで必要とするスタック領域もあるので，limit コマンドまたは ulimit コマンドなどで十分な大きさのスタックサイズを指定することを薦めます.

c.　使用例

n=1024, 標本点数 1023 の sine 変換を 5 本計算します.

```
C      **EXAMPLE**
       IMPLICIT REAL*8(A-H,O-Z)
       PARAMETER(N=1024,M=5)
       DIMENSION X(N-1,M),TAB(N*2)

       DO 100 J=1,M
       DO 100 I=1,N-1
       X(I,J)=FLOAT(MIN(I,(N-I)/J))
  100 CONTINUE

C      FORWARD TRANSFORM
       ISW=0
       CALL DVMSNT(X,N-1,N,M,ISW,TAB,ICON)
       PRINT*,'ICON=',ICON

C      BACKWARD TRANSFORM
```

```
      ISW=2
      CALL DVMSNT(X,N-1,N,M,ISW,TAB,ICON)
      PRINT*,'ICON=',ICON

      DO 200 J=1,M
      ERROR=0.0D0
      VNRM=0.0D0
      DO 210 I=1,N-1

      ERROR=ERROR+(X(I,J)/(N*2)-
     &    FLOAT(MIN(I,(N-I)/J)))**2
      VNRM=VNRM+(X(I,J)/(N*2))**2
  210 CONTINUE
      PRINT*,'ERROR=',SQRT(ERROR/VNRM)
  200 CONTINUE

      STOP
      END
```

(4)　手法概要

本ルーチンは，離散型 sine 変換を実フーリエ変換の問題に帰着させてスカラ CPU 上で高速に計算します．奇関数の対称性を利用することで，効率よく変換が可能であることが知られています．

詳細については“付録 参考文献一覧表”の[26]を参照してください．

A71-01-0101 VMVSD,DVMVSD

スパース実行列と実ベクトルの積（対角形式格納法）
CALL VMVSD(A,K,NDIAG,N,NOFST,NLB,X,Y,ICON)

(1)　機能

$n \times n$ のスパース行列とベクトルの積を計算します.

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$$

スパース行列 $\boldsymbol{A}$ は，対角形式の格納法で格納します.
ベクトル $\boldsymbol{x}$ および $\boldsymbol{y}$ は n 次元ベクトルです.

(2)　パラメタ

A.....入力．係数行列の非零要素を格納します.
A(K,NDIAG) なる実数型の 2 次元配列．A(1:N,NDIAG) にスパース行列の非零要素を格納します. 対角形式の格納方法については，“第 I 部 概説 3.2.1.1 一般スパース行列の格納方法　b. 一般スパース行列の対角形式の格納方法”を参照してください.

K.....入力．配列 A の整合寸法（≧n）.

NDIAG.....入力. 配列 A に格納する係数行列の非零要素を含む対角ベクトル列の総数.
A の 2 次元目の大きさ.

N.....入力．行列 $\boldsymbol{A}$ の次数 n.

NOFST.....入力．NOFST(NDIAG) なる 1 次元配列．配列 A に格納した対角ベクトルに対応した主対角ベクトルからの距離を格納します. 上対角ベクトル列は正，下対角ベクトル列は負の値で表します.

NLB .....入力．行列 $\boldsymbol{A}$ 下バンド幅.

X.....入力．ベクトル x を X(NLB＋1:NLB＋N) に格納します.
大きさ n＋nlb＋nub の 1 次元配列.
nlb：下バンド幅，nub：上バンド幅.

Y.....出力. 行列とベクトルの積の結果が格納されます. 大きさ n の 1 次元配列.

ICON.....出力．コンディションコード.
表 VMVSD-1 参照.

表 VMVSD-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	ー
30000	K＜1，N＜1，N＞K，NDIAG＜1 または， NLB≠MAX(-NOFST(I))または， ｜NOFST(I)｜＞N-1	処理を打ち切る.

(3) 使用上の注意

a. 使用する副プログラム

SSL II･･･ MGSSL

b. 注意

① 対角形式を使う上での注意

係数行列 *A* の外側の対角ベクトルの要素は零を設定する必要があります.

対角ベクトル列を配列 A に格納する順序に制限は特にありません.

この方法の利点は，行列ベクトル積が間接指標を使わずに計算できる点です.

対角構造を持たない行列を効率よく格納できないことが欠点です.

c. 使用例

DVCGD を利用するため，対角要素が 1 で，対角要素以外を配列 A に格納し，Ax を求めます．$b=(A-E)x+x$．SET は“VCGD,DVCGD（3）使用上の注意 c.使用例”を参照してください.

```
C     **EXAMPLE**
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (N=51200,K=N+1)
      PARAMETER (NW=2,IWKS=4,N2=K+1)
      PARAMETER (NVW=K*(NW+6)+10)
      REAL*8 B(N),EPS,OMEGA,RZ,VW(NVW),X(N)
      INTEGER NDLT(NW)
      REAL*8 A(K,NW)
      INTEGER IVW(N2,IWKS)
C
C INITIALISE A
      CALL SET(A,NDLT,K,NW,N)
      ISHIFT=0
      DO 10 J=1,NW
      ISHIFT=MAX(ISHIFT,ABS(NDLT(J)))
   10 CONTINUE
C COMPUTE RHS SO AX=B SO WE KNOW SOLUTION X (X(I)=I)
      DO 30 I=1,N
   30 VW(I+ISHIFT)=I
C
C     B=(A-E)*X+X
      CALL DVMVSD(A,K,NW,N,NDLT,ISHIFT,VW,B,ICON)
      DO 70 I=1,N
      B(I)=B(I)+VW(I+ISHIFT)
   70 CONTINUE
C
      ITMAX=8*SQRT(N+0.1)
      EPS=1D-10
      OMEGA=0D0
```

```
      ISW=1
      IGUSS=0
      DO 100 IPC=1,3
      IF(IPC.EQ.3) OMEGA=0.98
      CALL DVCGD(A,K,NW,N,NDLT,B,IPC,ITMAX,ISW,OMEGA,
     &           EPS,IGUSS,X,ITER,RZ,VW,IVW,ICON)
      IF(ICON.NE.0) WRITE(6,*)'ICON=',ICON
      IF(RZ.LE.EPS) WRITE(6,41)'CONVERGED. ACCURACY=',RZ
      IF(RZ.GT.EPS) WRITE(6,41)'FAILED. ACCURACY=',RZ
      WRITE(6,*)'X'
      DO 60 I=1,MIN(N,16),4
   60 WRITE(6,42) I,(X(M),M=I,I+3)
  100 CONTINUE
   42 FORMAT(1X,I3,4(1X,F20.10))
   41 FORMAT(A,2X,E10.3)
      STOP
      END
```

A71-02-0101 VMVSE,DVMVSE

スパース実行列と実ベクトルの積（ELLPACK 形式格納法）
CALL VMVSE(A,K,NW,N,ICOL,X,Y,ICON)

(1) 機能

$n \times n$ のスパース行列とベクトルの積を計算します.

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$$

$n \times n$ の係数行列は，2 つの配列を使用する，ELLPACK 形式の格納法で格納します.
y および x は n 次元ベクトルです.

(2) パラメタ

A.....入力．係数行列の非零要素を格納します.
A(K,NW) なる 2 次元配列.
ELLPACK 形式の格納方法については，“第 I 部 概説 3.2.1.1 一般スパース行列の格納方法”を参照してください.

K.....入力．配列 A および ICOL の整合寸法（≧n）.

NW.....入力. 配列 A に格納される行列 $\boldsymbol{A}$ の非零要素の行ベクトル方向の最大個数.
ICOL および A の 2 次元目の大きさ.

N.....入力．配列 A に格納される行列 $\boldsymbol{A}$ の次数.

ICOL .....入力．ELLPACK 形式で使用される列指標で，A の対応する要素がいずれの列ベクトルに属すかを示します.
ICOL(K,NW)なる 2 次元配列.

X.....入力．ベクトル x を格納します．大きさ n の 1 次元配列.

Y.....出力. 行列とベクトルの積の結果が格納されます. 大きさ n の 1 次元配列.

ICON.....出力．コンディションコード
表 VMVSE-1 参照.

表 VMVSE-1 コンディションコード

コード	意　味	処理内容
0	エラーなし.	ー
30000	K＜1，N≦0，NW＜1 または N＞K であった.	処理を打ち切る.

(3) 使用上の注意

a. 使用する副プログラム
SSL II･･･MGSSL

b.　注意

①　ELLPACK 格納形式を使う上での注意

ELLPACK 形式でデータを格納する前に，配列 A および ICOL をおのおの零，行ベクトルの番号で初期化することを勧めます.

c.　使用例

DVCGE を使用するとき, 行列 $\boldsymbol{A}$ の対角要素は 1 で, 対角要素以外を配列 A に格納し, $\boldsymbol{Ax}$ を $\boldsymbol{b=(A-E)x+x}$ で求めます．サブルーチン SET は“VCGE,DVCGE (3) 使用上の注意 c.使用例”を参照してください.

```
C     **EXAMPLE**
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (NW=2,N=51200,K=N+1)
      REAL*8 B(N),X(N),EPS,OMEGA,RZ,
     &      A(K,NW),VW(K*NW+4*N)
      INTEGER ICOL(K,NW),IVW(K*NW+4*N)
      WRITE(6,*) ' EXAMPLE DVCGE '
C INITIALISE A,ICOL
      CALL SET(A,ICOL,K,NW,N)
C GENERATE RHS B
      DO 10 I=1,N
   10 VW(I)=I
C COMPUTE RHS SO AX=B SO WE KNOW SOLUTION X (X(I)=I)
C
C B = (A-E)*X + E*X
      CALL DVMVSE(A,K,NW,N,ICOL,VW,B,ICON)
      PRINT*,'ERROR CODE =',ICON
      DO 20 I=1,N
      B(I)=B(I)+VW(I)
   20 CONTINUE
C
      ITMAX=4000
      EPS=1D-10
      ISW=1
      IGUSS=0
      DO 30 IPC=1,3
      IF(IPC.EQ.3)OMEGA=0.98
      CALL DVCGE(A,K,NW,N,ICOL,B,IPC,ITMAX,ISW,OMEGA
     &          ,EPS,IGUSS,X,ITER,RZ,VW,IVW,ICON)
C
      PRINT*,'ERROR CODE= ',ICON
      IF(RZ.LE.EPS) WRITE(6,41)'CONVERGED. ACCURACY=',RZ
      IF(RZ.GT.EPS) WRITE(6,41)'FAILED. ACCURACY=',RZ
      WRITE(6,*)'X'
```

```
   DO 60 I=1,MIN(N,16),4
60 WRITE(6,42) I,(X(M),M=I,I+3)
30 CONTINUE
42 FORMAT(I3,4(F12.4))
41 FORMAT(A,2X,E10.3)
   STOP
   END
```

A72-23-0101 VQMRD,DVQMRD

非対称または不定値のスパース実行列の連立 1 次方程式 （QMR 法，対角形式格納法）
CALL VQMRD(A,K,NDIAG,N,NOFST,AT,NTOFST,B,ITMAX,EPS,IGUSS, X,ITER,VW,ICON)

(1)　機能

$n \times n$ の非対称／不定値なスパース行列を係数行列とする連立 1 次方程式を準最小残差法（QMR:Quasi-Minimal Residual method）で解きます．

$$\boldsymbol{Ax = b}$$

$n \times n$ の係数行列 $\boldsymbol{A}$ 及び $\boldsymbol{A^T}$ の 2 つの行列を使用します．対角形式の格納法で格納します．ベクトクト b および x は n 次元ベクトルです．

反復計算が係数行列や初期ベクトルの特性のため続けられない場合（break down）があります．これは再帰的計算公式で，非零を期待される計算のある中間結果が零になるためです．この場合，break down を起こさない MGCR 法をお使いください．

反復解法の収束および利用指針に関しては，“第 I 部 概説 第 4 章 連立一次方程式の反復解法と収束性”を参照してください．

(2)　パラメタ

A.....入力．係数行列の非零要素を格納します．
A（K,NDIAG）なる 2 次元配列．A（1:N,NDIAG）に係数行列 $\boldsymbol{A}$ を対角形式で格納します．
対角形式の格納方法については，“第 I 部 概説 3.2.1.1 一般スパース行列の格納方法 b.一般スパース行列の対角形式の格納方法”を参照してください．

K.....入力．配列 A の整合寸法．

NDIAG.....入力．係数行列 $\boldsymbol{A}$ の非零要素を含む対角ベクトル列の総数．
配列 A の 2 次元目の大きさ．

N.....入力．行列 $\boldsymbol{A}$ の次数 n．

NOFST.....入力．A に格納される対角ベクトルに対応した主対角ベクトルからの距離を格納します．上対角ベクトル列は正，下対角ベクトル列は負の値で表します．
NOFST（NDIAG）なる 1 次元配列．

AT.....入力．係数行列 $\boldsymbol{A}$ を転置した $\boldsymbol{A^T}$ の非零要素を格納します．
AT（K,NDIAG）なる 2 次元配列．AT（1:N,NDIAG）に係数行列 A^T を格納します．
対角形式の格納方法については，“第 I 部 概説 3.2.1.1 一般スパース行列の格納方法 b.一般スパース行列の対角形式の格納方法”を参照してください．

NTOFST .....入力. 配列 AT に格納される対角ベクトルに対応した主対角ベクトルからの距離を格納します．上対角ベクトル列は正，下対角ベクトル列は負の値で表します.
NTOFST（NDIAG）なる 1 次元配列.

B.....入力. 大きさ *n* の 1 次元配列. 連立 1 次方程式の右辺の定数ベクトルを格納します.

ITMAX .....入力．QMR 法の反復回数の上限値（＞0）．2000 程度で十分です.

EPS.....入力．収束判定に用いられる判定値.
EPS が 0.0 以下のとき，EPS は倍精度ルーチンでは 10^{-6} が，単精度ルーチンでは 10^{-4} が設定されます.
（(3) 使用上の注意 b.注意①参照）.

IGUSS.....入力．配列 X に指定された解ベクトルの近似値から反復計算を行うかを指定する制御情報.
0 のとき　解ベクトルの近似値を指定しません.
0 以外のとき 配列 X に指定された解ベクトルの近似値から反復計算を開始します.

X.....入力．大きさ *n* の 1 次元配列．解ベクトルの近似値を指定することができます.
出力．解ベクトルが格納されます.

ITER.....出力．QMR 法での実際の反復回数.

VW.....作業域．大きさ，K×9＋N＋NBANDL＋NBANDR なる 1 次元配列.
NBANDL は下バンド幅，NBANDR は上バンド幅の大きさ.

ICON.....出力．コンディションコード.
表 VQMRD-1 参照.

表 VQMRD-1 コンディションコード

コード	意　味	処理内容
0	エラーなし.	ー
20000	Break down を起こした.	処理を打切る.
20001	反復回数の上限に達した.	処理を打ち切る．配列 X には，そのときまでに得られている近似値を出力するが，精度は保証できない.
30000	N＜1，K＜1，K＜N, NDIAG＜1，K＜NDIAG, または ITMAX ≦ 0	処理を打ち切る.
32001	｜NOFST (I)｜＞N-1, ｜NTOFST (I)｜＞N-1	

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ AMACH,URGWD,URIPA,URITI,URITT,URMVD,USSCP,URSTE,
USVCN,USVCP,USVSC,USVSU,USVUP,USVN2,URELT,MGSSL,

UQMRR,UQMRD,UQBBM,UQITB

b.　注意

①　QMR 法は, 残差のユーグリッドノルムが最初の残差のユーグリッドノルムと EPS の積以下になったとき収束したと見なします．正確な解と求められた近似解の誤差はほぼ行列 ***A*** の条件数と EPS の積に等しくなります.

②　対角形式を使う上での注意

係数行列 ***A*** の外側の対角ベクトルの要素は零を設定する必要があります.

対角ベクトル列を配列 A に格納する順序に制限は特にありません.

この方法の利点は，行列ベクトル積が間接指標を使わずに計算できる点です.

対角構造を持たない行列は効率よく格納できないという点が欠点です.

c.　使用例

領域［0,1］×［0,1］×［0,1］でディリクレ境界条件（境界で関数値は零）のもとで,“第 I 部　概説　3.2.2　偏微分演算子の離散化とその格納例”に記述された偏微分演算子を離散化して得られる係数行列を持つ連立 1 次方程式を解きます.

INIT_MAT_DIAG は，“第 I 部 概説 3.2.2 偏微分演算子の離散化とその格納例”を参照してください．INIT_MAT_TR_DIAG その転置行列を生成するルーチン，GET_BANDWIDTH_DIAG はバンド幅見積もりのルーチン，INIT_SOL は，求めるべき解ベクトルを乱数で生成するルーチンです.

```
C     **EXAMPLE**
      PROGRAM TEST_ITER_SOLVERS
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER MACH
      PARAMETER (MACH = 0)
      PARAMETER (K = 10000)
      PARAMETER (NX = 20,NY = 20,NZ = 20,N = NX*NY*NZ)
      PARAMETER (NDIAG = 7,NVW = 9*K+N+400+400)
      REAL*8 A(K,NDIAG),AT(K,NDIAG),X(N),B(N),SOLEX(N)
     &       ,VW(NVW)
      INTEGER NOFST(NDIAG),NTOFST(NDIAG)
C
      CALL INIT_SOL(SOLEX,N,1D0,MACH)
      PRINT*,'EXPECTED SOLUSIONS'
      PRINT*,'X(1) = ',SOLEX(1),'X(N) = ',SOLEX(N)
C
      PRINT *
      PRINT *,'    QMR METHOD'
      PRINT *,'    DIAGONAL FORMAT'
C
      VA1 = 3D0
      VA2 = 1D0/3D0
      VA3 = 5D0
      VC = 1.0
```

```
      XL = 1.0
      YL = 1.0
      ZL = 1.0

      CALL INIT_MAT_DIAG(VA1,VA2,VA3,VC,A,NOFST
     &          ,NX,NY,NZ,XL,YL,ZL,NDIAG,N,K)
      CALL INIT_MAT_TR_DIAG(VA1,VA2,VA3,VC,AT,NTOFST
     &          ,NX,NY,NZ,XL,YL,ZL,NDIAG,N,K)
      CALL GET_BANDWIDTH_DIAG(NOFST,NDIAG,NBANDL,NBANDR)
         DO 110 I = 1,N
            VW(I+NBANDL) = SOLEX(I)
 110     CONTINUE
      CALL DVMVSD(A,K,NDIAG,N,NOFST,NBANDL,VW,B,ICON)
      PRINT*,'DVMVSD ICON = ',ICON
C
      EPS = 1D-10
      IGUSS = 0
      ITMAX = 2000
      CALL DVQMRD(A,K,NDIAG,N,NOFST,AT,NTOFST,B,ITMAX
     &           ,EPS,IGUSS,X,ITER,VW,ICON)
C
      PRINT* ,'ITER = ',ITER
      PRINT* ,'DVQMRD ICON = ',ICON
      PRINT*, 'COMPUTED VALUES'
      PRINT*, 'X(1) = ',X(1),'X(N) = ',X(N)
      STOP
      END
```

(4)　手法概要

QMR 法は“付録 参考文献一覧表”の [13] を参照してください.

A72-24-0101 VQMRE,DVQMRE

非対称または不定値のスパース実行列の連立 1 次方程式 （QMR 法，ELLPACK 形式格納法）
CALL VQMRE(A,K,IWIDT,N,ICOL,AT,IWIDTT,ICOLT,B,ITMAX,EPS, IGUSS,X,ITER,VW,ICON)

(1)　機能

$n \times n$ の非対称／不定値なスパース行列を係数行列とする連立 1 次方程式を準最小残差法（QMR:Quasi-Minimal Residual method）で解きます．

$$\boldsymbol{Ax} = \boldsymbol{b}$$

$n \times n$ の係数行列 $\boldsymbol{A}$ 及び $\boldsymbol{A}^T$ の 2 つの行列を使用します．ELLPACK 形式の格納法で格納します．ベクトル $\boldsymbol{b}$ および $\boldsymbol{x}$ は n 次元ベクトルです．

反復計算が係数行列や初期ベクトルの特性のため続けられない場合（break down）があります．これは再帰的計算公式で，非零を期待される計算のある中間結果が零になるためです．この場合 break down を起こさない MGCR 法をお使いください．

反復解法の収束および利用指針に関しては，“第 I 部 概説 第 4 章 連立一次方程式の反復解法と収束性”を参照してください．

(2)　パラメタ

A.....入力．係数行列 A の非零要素を格納します．
A（K,IWIDT）なる 2 次元配列．
ELLPACK 形式の格納方法については，“第 I 部 概説 3.2.1.1 一般スパース行列の格納方法”を参照してください．

K.....入力．A および ICOL の整合寸法（≧ n）．

IWIDT.....入力．係数行列 $\boldsymbol{A}$ の非零要素の行ベクトル方向の最大個数．
A および ICOL の 2 次元目の大きさ．

N.....入力．行列 A の次数 n．

ICOL .....入力．ELLPACK 形式で使用される列指標で，A の対応する要素がいずれの列ベクトルに属すかを示します．
ICOL（K,IWIDT）なる 2 次元配列．

AT .....入力．転置した係数行列 $\boldsymbol{A}^T$ の非零要素を AT（1:N,IWIDTT）に格納します．
AT（K,IWIDTT）なる 2 次元配列．
ELLPACK 形式の格納方法については，“第 I 部 概説 3.2.1.1 一般スパース行列の格納方法”を参照してください．

IWIDTT .....入力．転置した係数行列 $\boldsymbol{A}^T$ の非零要素の行ベクトル方向の最大個数．

ICOLT.....入力．ELLPACK 形式で使用される列指標で，AT の対応する要素がいずれの列ベクトルに属すかを示します．
ICOLT（K,IWIDTT）なる 2 次元配列．

B.....入力．大きさ n の 1 次元配列．連立 1 次方程式の右辺の定数ベクトルを B に格納します．

ITMAX .....入力．QMR 法の反復回数の上限値（＞0）．2000 程度で十分です．

EPS.....入力．収束判定に用いられる判定値．

EPS が 0.0 以下のとき，EPS は倍精度ルーチンでは 10^{-6} が，単精度ルーチンでは 10^{-4} が設定されます．

（(3) 使用上の注意 b.注意①参照）．

IGUSS.....入力．配列 X に指定された解ベクトルの近似値から反復計算を開始するかを示す制御情報．

0 のとき　解ベクトルの近似値を指定しません．

0 以外のとき 配列 X に指定された解ベクトルの近似値から反復計算を開始します．

X.....入力．大きさ *n* の 1 次元配列．解ベクトルの近似値を指定することができます．

出力．解ベクトルが格納されます．

ITER.....出力．QMR 法での実際の反復回数．

VW.....作業域．大きさ，K×12 なる 1 次元配列．

ICON.....出力．コンディションコード．

表 VQMRE-1 参照．

表 VQMRE-1 コンディションコード

コード	意　味	処理内容
0	エラーなし．	ー
20000	break down を起こした．	処理を打切る．
20001	最大反復回数に達した．	処理を打ち切る．配列 X には，そのときまでに得られている近似値を出力するが，精度は保証できない．
30000	K＜1，N＜1，K＜N，IWIDT＜1，IWIDTT＜1，K＜IWIDT，K＜IWIDTT，または ITMAX≦0	処理を打切る．

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ AMACH,URIPA,URITI,URITT,URMVE,USSCP,URSTE,USVCN,
USVCP,USVSC,USVSU,USVUP,USVN2,URELT,MGSSL,UQMRR,
UQMRE,UQBBM,UQITB

b.　注意

①　QMR法は, 残差のユーグリッドノルムが最初の残差のユーグリッドノルムとEPSの積以下になったとき収束したと見なします．

正確な解と求められた近似解の誤差はほぼ行列 *A* の条件数と EPS の積に等しくなります．

c.　使用例

領域［0,1］×［0,1］×［0,1］でディリクレ境界条件（境界で関数値は零）のもとで，“第 I 部 概説 3.2.2 偏微分演算子の離散化とその格納例”に記述された偏微分演算子を離散化して得られる係数行列を持つ連立 1 次方程式を解きます. INIT_MAT_ELL は，“第 I 部 概説 3.2.2 偏微分演算子の離散化とその格納例”を参照してください. INIT_MAT_TR_DIAG はその転置行列を生成するルーチン，INIT_SOL は，求めるべき解ベクトルを乱数で生成するルーチンです.

```
C     **EXAMPLE**
      PROGRAM TEST_ITER_SOLVERS
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (MACH = 0)
      PARAMETER (K = 10000)
      PARAMETER (NX = 20,NY = 20,NZ = 20,N = NX*NY*NZ)
      PARAMETER (IWIDT = 7,IWIDTT = IWIDT,NVW = K*12)
      REAL*8 A(K,IWIDT),AT(K,IWIDTT),X(N),B(N),SOLEX(N)
     &       ,VW(NVW)
      INTEGER ICOL(K,IWIDT),ICOLT(K,IWIDTT)
C
      CALL INIT_SOL(SOLEX,N,1D0,MACH)
      PRINT*,'EAPECTED SOLUTION'
      PRINT*,'X(1) = ',SOLEX(1),' X(N) = ',SOLEX(N)
C
      PRINT *
      PRINT *,'    QMR METHOD'
      PRINT *,'    ELLPACK FORMAT'
C
      VA1 = 3D0
      VA2 = 1D0/3D0
      VA3 = 5D0
      VC = 5D0
      XL = 1.0
      YL = 1.0
      ZL = 1.0
C
      CALL INIT_MAT_ELL(VA1,VA2,VA3,VC,A,ICOL
     &         ,NX,NY,NZ,XL,YL,ZL,IWIDT,N,K)
      CALL INIT_MAT_TR_ELL(VA1,VA2,VA3,VC,AT,ICOLT
     &         ,NX,NY,NZ,XL,YL,ZL,IWIDT,N,K)
      CALL DVMVSE(A,K,IWIDT,N,ICOL,SOLEX,B,ICON)
      PRINT*,'DVMVSE ICON = ',ICON
C
      EPS = 1D-10
```

```
      IGUSS = 0
      ITMAX = 800
      CALL DVQMRE(A,K,IWIDT,N,ICOL,AT,IWIDTT,ICOLT,B,ITMAX
     &           ,EPS,IGUSS,X,ITER,VW,ICON)
C
      PRINT*,'DVQMRE ICON = ',ICON
      PRINT*,'COMPUTED VALUE'
      PRINT*,'X(1) = ',X(1),' X(N) = ',X(N)
      STOP
      END
```

(4)　手法概要

QMR 法は“付録 参考文献一覧表”の [13] を参照してください.

F17-13-0401 VRCVF,DVRCVF

実データの離散畳み込み および 離散相関
CALL VRCVF(X,K,N,M,Y,IVR,ISW,TAB,ICON)

(1)　機能

実データの 1 次元，多重の離散畳み込み または 離散相関をフーリエ変換を利用して計算します．

①　畳み込み

フィルターと，1 本のデータをそれぞれ $\{y_i\}$ と $\{x_j\}$ としたとき，以下で定義される畳み込みをデータ配列の各列に対して行い，$\{z_k\}$ を計算します．

$$z_k = \sum_{i=0}^{n-1} x_{k-i} y_i, \qquad k = 0,...,n-1$$

②　相関

フィルターと 1 本のデータをそれぞれ $\{y_i\}$ と $\{x_j\}$ としたとき，以下で定義される相関データ配列の各列に対して行い，$\{z_k\}$ を計算します．

$$z_k = \sum_{i=0}^{n-1} x_{k+i} y_i, \qquad k = 0,...,n-1$$

ここで，x_j は周期 n の周期的データです．（(3)使用上の注意 b.注意①参照）

(2)　パラメタ

X.....入力．m 本の $\{x_j\}$ データ, j=0,...,n–1 を X(1:N,1:M)に格納します．
出力．m 本の $\{z_k\}$ k=0,...,n–1 が X(1:N,1:M)に格納されます．
X(K,M)なる 2 次元配列．

K.....入力．配列 X の整合寸法．K ≥ N

N.....入力．1 本のデータの要素数 n．n は偶数．（(3)使用上の注意 b.注意②参照）

M.....入力．多重度 m.

Y.....入力．フィルター $\{y_i\}$，i=0,...,n–1 を Y(1:N)に格納します．
Y(N)なる 1 次元配列．（(3)使用上の注意 b.注意③,④参照）
ISW=0 または 2 のとき，値は保存されません．

IVR.....入力．畳み込みと相関の処理を選択します．
0　のとき　　畳み込みを計算します．
1　のとき　　相関を計算します．

ISW .....入力．制御情報．
0　のとき　1 回の呼び出しで処理を完了します．
以下の ISW 指定で順次呼び出すことで，処理のステップをわけて利用できます．（(3)使用上の注意 b.注意③参照）
1　のとき　TAB 設定のみ．
2　のとき　Y のフーリエ変換を行います．

3　のとき　設定済みの TAB と変換済みの Y を使用して畳み込みまたは相関を計算します.

TAB .....作業域．三角関数表を格納します．TAB(2×N)の 1 次元配列.

ICON.....出力．コンディションコード.

表 VRCVF-1 参照.

表 VRCVF-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	―
30000	N≦0, K<N, M≦0, ISW≠0,1,2,3, IVR≠0,1 または N が偶数でなかった.	処理を打ち切る.

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ UMRF2,UZFB2,UZFB3,UZFB4,UZFB5, UZFB8, UZFB6,
UZFF2,UZFF3,UZFF4,UZFF5, UZFF8,UZFF6, UZFPB,UZFPF,
UZFTB,UZFTF,UZUNI,MGSSL

b.　注意

①　非周期的データに対する畳み込みおよび相関について

データ x の長さが n_x，y の長さが n_y であるとき，$n \geq n_x + n_y - 1$ として X(n_x+1：n,＊)および Y(n_y+1：n)には 0 を設定することで，非周期データに対する畳み込みおよび相関が計算できます．（c.使用例の例２を参照）

このとき，相関の z_k ($k = -n_y+1,...,-1$)に相当する値は，X($n-n_y$+2：n,＊)に格納されることになります.

②　n の値について

1 本のデータの要素数 n は偶数であれば任意の数が指定可能ですが, 2,3,5 の因子に分解される場合が高速です.

③　TAB および Y の再利用について

同じ N で本ルーチンを繰り返し呼び出す場合，初回に ISW=0 または 1 で呼び出した後，2 回目以降は ISW=2 および 3 で呼び出すことにより，TAB 作成を省略して処理の高速化ができます．さらに，同じフィルターy を使用する場合には，ISW=0 または 2 で変換された Y をそのまま ISW=3 の呼び出しで繰り返し利用することで処理が効率化できます.

このように本ルーチンを繰り返し呼び出す場合，Y に対して 2 回以上の変換が行われないように注意が必要です.

④　自己相関計算について

引数 X と Y に同一の実引数を指定することで，自己相関の計算が可能です．このとき，ISW=2 の指定は無視されます．（c.使用例の例３を参照）

⑤　スタックサイズについて

本サブルーチンでは，内部で使用する作業域を自動割付け配列としてスタックに確保しています．このため，スタックが不足すると異常終了する可能性があ

ります．本サブルーチンが自動割付け配列でスタックに必要とするサイズは，単精度ルーチンでは 4×N byte．倍精度ルーチンでは，8×N byte になります．この他に，固定サイズの作業領域や，ユーザープログラムで必要とするスタック領域もあるので，limit コマンドまたは ulimit コマンドなどで十分な大きさのスタックサイズを指定することを薦めます．

c.　使用例

例１：周期的データに対する畳み込みを計算します．

```
C     ** PERIODIC CONVOLUTION EXAMPLE **
      IMPLICIT REAL*8(A-H,O-Z)
      PARAMETER(K=8,M=3)
      DIMENSION X(K,M),Y(K),TAB(K*2)

      N=8

C     --SET SAMPLE DATA--
      DO 100 J=1,M
      DO 100 I=1,N
      X(I,J)=FLOAT(I+J-1)
  100 CONTINUE
      DO 110 I=1,N
      Y(I)=FLOAT(I+10)
  110 CONTINUE

      WRITE(*,*)'--INPUT DATA--'
      DO 120 J=1,M
      WRITE(*,900)J,(X(I,J),I=1,N)
  120 CONTINUE
      WRITE(*,910)(Y(I),I=1,N)

C     --CALL DVRCVF--
      IVR=0
      ISW=0
      CALL DVRCVF(X,K,N,M,Y,IVR,ISW,TAB,ICON)

      WRITE(*,*)'--OUTPUT DATA--'
      DO 130 J=1,M
      WRITE(*,900)J,(X(I,J),I=1,N)
  130 CONTINUE

  900 FORMAT('X(*,',I2,') :',3X,8F8.2)
  910 FORMAT('Filter Y:',3X,8F8.2)
```

```
      STOP
      END
```

例２：非周期的データに対する畳み込みを計算します.

```
C     ** NONPERIODIC CONVOLUTION EXAMPLE **
      IMPLICIT REAL*8(A-H,O-Z)
      PARAMETER(K=16,M=3)
      DIMENSION X(K,M),Y(K),TAB(K*2)

      NX=7
      NY=9
      N=NX+NY-1
      IF(MOD(N,2).NE.0)N=N+1

C     --SET SAMPLE DATA--
      DO 100 J=1,M
      DO 110 I=1,NX
      X(I,J)=FLOAT(I+J-1)
  110 CONTINUE
      DO 120 I=NX+1,N
      X(I,J)=0.0D0
  120 CONTINUE
  100 CONTINUE
      DO 130 I=1,NY
      Y(I)=FLOAT(I+10)
  130 CONTINUE
      DO 140 I=NY+1,N
      Y(I)=0.0D0
  140 CONTINUE

      WRITE(*,*)'--INPUT DATA--'
      DO 150 J=1,M
      WRITE(*,900)J,(X(I,J),I=1,N)
  150 CONTINUE
      WRITE(*,910)(Y(I),I=1,N)

C     --CALL DVRCVF--
      IVR=0
      ISW=0
      CALL DVRCVF(X,K,N,M,Y,IVR,ISW,TAB,ICON)

      WRITE(*,*)'--OUTPUT DATA--'
```

```
      DO 160 J=1,M
      WRITE(*,900)J,(X(I,J),I=1,N)
  160 CONTINUE

  900 FORMAT('X(*,',I2,') :'/(12X,8F8.2))
  910 FORMAT('Filter Y:'/(12X,8F8.2))
      STOP
      END
```

例３：自己相関を計算します.

```
C     ** AUTOCORRELATION EXAMPLE **
      IMPLICIT REAL*8(A-H,O-Z)
      PARAMETER(K=8,M=3)
      DIMENSION X(K,M),TAB(K*2)

      NX=4
      N=NX*2

C     --SET SAMPLE DATA--
      DO 100 J=1,M
      DO 110 I=1,NX
      X(I,J)=FLOAT(I+J-1)
  110 CONTINUE
      DO 120 I=NX+1,N
      X(I,J)=0.0D0
  120 CONTINUE
  100 CONTINUE

      WRITE(*,*)'--INPUT DATA--'
      DO 130 J=1,M
      WRITE(*,900)J,(X(I,J),I=1,N)
  130 CONTINUE

C     --CALL DVRCVF--
      IVR=1
      ISW=1
      CALL DVRCVF(X,K,N,M,X,IVR,ISW,TAB,ICON)
      ISW=3
      CALL DVRCVF(X,K,N,M,X,IVR,ISW,TAB,ICON)

      WRITE(*,*)'--OUTPUT DATA--'
      DO 140 J=1,M
```

```
      WRITE(*,900)J,(X(I,J),I=1,N)
  140 CONTINUE

  900 FORMAT('X(*,',I2,') :',3X,8F8.2)
      STOP
      END
```

(4)　手法概要

離散畳み込み，および離散相関はフーリエ変換を利用することで効率的に計算できることが知られています．離散畳み込みではデータとフィルターそれぞれの離散フーリエ変換を計算した後，それらの要素毎の積を計算し，逆変換することで結果が得られます．離散相関でもほぼ同様に，データの離散フーリエ変換と，フィルターの離散フーリエ変換の共役を計算して，それらの要素毎の積を逆フーリエ変換することで計算されます．詳細については“付録 参考文献一覧表”の[26]を参照してください．

F15-31-0401 VRPF3,DVRPF3

3 次元素因子離散型実フーリエ変換
CALL VRPF3(A,L,M,N,ISN,VW,ICON)

(1)　機能

3 次元（各元の項数 $N1$, $N2$, $N3$）の実時系列データ $\{x_{J1,\,J2,\,J3}\}$ が与えられたとき，離散型実フーリエ変換，またはその逆変換を素因子フーリエ変換（a prime factor FFT）により行います．各元の項数は，次の条件を満たす数です．

- 次の数の中で互いに素な因子の積で表せること．
  因子 p($p \in \{2,3,4,5,7,8,9,16\}$)
- 1 次元目の項数は偶数（$2\times I$）であり，I は上の条件を満たす．

このルーチンで，パラメタ N に 1 を設定して呼び出すと 2 次元実素因子高速フーリエ変換になります．さらに，パラメタ N に 1 をそしてパラメタ M に 1 を設定して呼び出すと 1 次元実素因子高速フーリエ変換になります．

① 3 次元実フーリエ変換

3 次元フーリエ変換は $\{x_{J1,\,J2,\,J3}\}$ を入力し（1.1）で定義する変換を行い，$\{N1\times N2\times N3\times \alpha_{K1,\,K2,\,K3}\}$ を求めます．

$$N1\times N2\times N3\times \alpha_{K1,\,K2,\,K3} = \sum_{J1=0}^{N1-1}\sum_{J2=0}^{N2-1}\sum_{J3=0}^{N3-1} x_{J1,J2,J3}\omega_1^{-J1.K1}\omega_2^{-J2.K2}\omega_3^{-J3.K3} \qquad (1.1)$$

$,K1 = 0,1,..., N1-1$

$,K2 = 0,1,..., N2-1$

$,K3 = 0,1,..., N3-1$

$,\omega_i = \exp(2\pi i/Nj),\ \ j = 1,2,3$

3次元実フーリエ変換の場合，$\{x_{J1,\,J2,\,J3}\}$が実数なので約半分の計算を行います．つまり 1 次元についての $K1$ は 0 から $N1/2$ までの計算を行います．

② 3 次元実フーリエ逆変換

3 次元フーリエ逆変換は $\{\alpha_{K1,\,K2,\,K3}\}$ を入力し，（1.2）で定義する変換を行い，$\{x_{J1,\,J2,\,J3}\}$ を求めます．

$$x_{J1,\,J2,\,J3} = \sum_{K1=0}^{N1-1}\sum_{K2=0}^{N2-1}\sum_{K3=0}^{N3-1}\alpha_{K1,K2,K3}\omega_1^{J1.K1}\omega_2^{J2.K2}\omega_3^{J3.K3} \qquad (1.2)$$

$,J1 = 0,1,..., N1-1$

$,J2 = 0,1,..., N2-1$

$,J3 = 0,1,..., N3-1$

$,\omega_i = \exp(2\pi i/Nj),\ \ j = 1,2,3$

3 次元実フーリエ逆変換では，$\alpha_{K1,K2,K3}$は $K1$＝0,1,...,$N1$/2 しか求めていませんが，2 次元，3 次元の$\sum$を先に計算して，1 次元目の要素に関する共役関係を利用して（1.2）を計算します.

(2)　パラメタ

A.....入力．実数$\{x_{J1,J2,J3}\}$または，フーリエ変換された複素数$\{\alpha_{K1,K2,K3}\}$．
出力. フーリエ変換した複素数$\{\alpha_{K1,K2,K3}\}$または, 逆変換された実数$\{x_{J1,J2,J3}\}$.
A(L,M,N)なる 3 次元配列.
実数の場合（変換の入力と逆変換の出力）は A(L,M,N) にデータが格納されます.
1 次元目のデータ数は L-2 であり，L-2 は偶数です.
2 次元目，3 次元目のデータ数はそれぞれ M,N です.
複素数の場合（変換の出力と逆変換の入力）は，おなじ配列の前半と後半にそれぞれ実部と虚部が格納されます.
PARAMETER (L＝ ,M＝ ,N＝ ,LH＝L/2)
DIMENSION A(L,M,N),B(LH,M,N,2)
EQUIVALENCE (A,B)
実部：B(L/2,M,N,1)
虚部：B(L/2,M,N,2)
に連続領域に別々に格納されています．((3) 使用上の注意 b.注意③参照）

L.....入力．1 次元のデータ数＋2．1 次元のデータ数は偶数でなければなりません.
(L-2) / 2 ≦ 5040

M.....入力．2 次元のデータ数.
M≦5040

N.....入力．3 次元のデータ数.
N≦5040

ISN.....入力．変換か逆変換かを指定します.
ISN≧0（非負の整数）なら変換,
ISN＜0（負の整数）なら逆変換.

VW.....作業域．A と同じ大きさの 3 次元配列.

ICON.....出力．コンディションコード.
表 VRPF3-1 参照.

表 VRPF3-1　コンディションコード

コード	意　味	処理内容
0	エラーなし.	ー
20000	(L-2)/2，M，N のいずれかが 5040 を超えるか，{2,3,4,5,7,8,9,16} の中の互いに素な因子の積に分解できなかった.	処理を打ち切る.
30000	L-2 が偶数でなかったか，L，M，N のいずれかが 0 か負であった.	

(3)　使用上の注意

a.　使用する副プログラム

SSL II・・・ UINI1,UINI2,UTER1,UTER2,UTRSP,UPFT1,UPFT2,UTRR1,
UTRR2,MGSSL

b.　注意

①　一般的な 3 次元フーリエ変換の定義について

3 次元フーリエ変換及び，逆変換は一般的に (3.1)，(3.2) で定義されます.

$$\alpha_{K1,K2,K3} = \frac{1}{N1\times N2\times N3}\sum_{J1=0}^{N1-1}\sum_{J2=0}^{N2-1}\sum_{J3-0}^{N3-1} x_{J1,J2,J3}\omega_1^{-J1.K1}\omega_2^{-J2.K2}\omega_3^{-J3.K3} \tag{3.1}$$

$$x_{J1,J2,J3} = \sum_{K1=0}^{N1-1}\sum_{K2=0}^{N2-1}\sum_{K3-0}^{N3-1}\alpha_{K1,K2,K3}\omega_1^{J1.K1}\omega_2^{J2.K2}\omega_3^{J3.K3} \tag{3.2}$$

本サブルーチンでは，(3.1)，(3.2)の左辺に対応して $\{N1\times N2\times N3\times\alpha_{K1,K2,K3}\}$ 又は $\{x_{J1,J2,J3}\}$ を求めるので結果の正規化は必要に応じて行ってください．ちなみに，本サブルーチンにより変換・逆変換を正規化せずに連続して実行すると，入力データの各要素は $N1\cdot N2\cdot N3$ 倍されて出力されます.

②　項数について

項数は次の数の中で互いに素な因子の積で表せる数です.

最大値は 5×7×9×16＝5040 です.

因子 p($p\in$ {2,3,4,5,7,8,9,16})

1 次の項数は以上の数の倍数まで許されます.

③　データの格納方法について

実データ $\{x_{J1,J2,J3}\}$ は 3 次元配列 A に格納します．ただし 1 次元目の項数 $N1$ は，L-2 と等しく，1 から L-2 までに入ります.

複素データ $\{\alpha_{K1,K2,K3}\}$ は, 配列 A を連続した 2 つの領域に分けて配列とみなし，実部と虚部を格納します. 1 次元目の添字 $K1$ は, 0 から $N1/2$ までの $N1/2+1$(L/2) 個です.

LH = L/2

の関係があるとき,

```
PARAMETER   (L= ,M= ,N= ,LH=L/2)
DIMENSION    A(L,M,N),B(LH,M,N,2)
EQUIVALENCE (A,B)
```

とすれば，B(LH,M,N,1) に実部が，B(LH,M,N,2) に虚部が格納されます.

c. 使用例

$N1, N2, N3$ 項の実時系列データ $\{x_{J1,\,J2,\,J3}\}$ を入力し，フーリエ変換し，その結果をフーリエ逆変換し $\{x_{J1,\,J2,\,J3}\}$ を求めます.

ここでは，$N1$＝12，$N2$＝12，$N3$＝12，とします.

```
C     **EXAMPLE**
      DIMENSION A(12+2,12,12),B(6+1,12,12,2),NI(3)
      DIMENSION VW(12+2,12,12)
      DATA      NI/12,12,12/,L,M,N/14,12,12/
      EQUIVALENCE (A,B)
      READ(5,500) (((A(I,J,K),I=1,NI(1)),
     *             J=1,NI(2)),K=1,NI(3))
      WRITE(6,600) (NI(I),I=1,3),
     *            (((I,J,K,A(I,J,K),I=1,NI(1)),
     *             J=1,NI(2)),K=1,NI(3))
C     NORMAL TRANSFORM
      CALL VRPF3(A,L,M,N,1,VW,ICON)
      WRITE(6,610) ICON
      IF(ICON.NE.0) STOP
C     INVERSE TRANSFORM
      CALL VRPF3(A,L,M,N,-1,VW,ICON)
      NT=NI(1)*NI(2)*NI(3)
      DO 10 K=1,NI(3)
      DO 10 J=1,NI(2)
      DO 10 I=1,NI(1)
      A(I,J,K)=A(I,J,K)/FLOAT(NT)
   10 CONTINUE
      WRITE(6,620) (((I,J,K,A(I,J,K),I=1,NI(1)),
     *              J=1,NI(2)),K=1,NI(3))
      STOP
  500 FORMAT(E20.7)
  600 FORMAT('0',10X,'INPUT DATA',5X,
     *       '(',I3,',',I3,',',I3,')'/
     *      (15X,'(',I3,',',I3,',',I3,')',
     *       E20.7))
  610 FORMAT('0',10X,'RESULT ICON=',I5)
  620 FORMAT('0',10X,'OUTPUT DATA'/
     *      (15X,'(',I3,',',I3,',',I3,')',
     *       E20.7))
      END
```

(4)　手法概要

3 次元実フーリエ変換を，基底が互いに素な因子に分解される数を基底とする高速変換手法（aprime factor FFT）により行います.

①　3 次元変換

（1.1）で定義される 3 次元変換は，共通項を整理することにより（4.1）のような順序に変えることができます. *J*1,*J*2,*J*3 に関する和をとる順序を入替えることも可能です.

$$N1\times N2\times N3\times \alpha_{K1,K2,K3}$$

$$=\sum_{J1=0}^{N1-1}\omega_1^{-J1.K1}\times\sum_{J2=0}^{N2-1}\omega_2^{-J2.K2}\sum_{J3-0}^{N3-1}x_{J1,J2,J3}\omega_3^{-J3.K3} \tag{4.1}$$

（4.1）の $\sum_{j3}$ は *N*3 項の 1 次変換を *N*1×*N*2 組行い，$\sum_{j2}$ は *N*2 項の 1 次変換を *N*1×*N*3 組行い，$\sum_{j1}$ は *N*1 項の 1 変換を *N*2×*N*3 組行います.

本ルーチンでは各次元に対する 1 次元変換を行うために，互いに素な因子を基底に持つ高速フーリエ変換を適用しています.

②　実変換

これは 1 次元目の実フーリエ変換でデータ数が偶数であるので，*K*1 に関して 0 から *N*1/2 までの複素フーリエ変換を計算します．複素共役関係から，残りのフーリエ変換の計算は不要です.

項数 N の 1 次元の離散型実フーリエ変換を考えます.

$$\alpha_K=\sum_{J=0}^{N-1}x_J\exp(-2\pi iKJ/N) \tag{4.2}$$

$$K=0,\cdots,N-1$$

$$\alpha_{N-K}=\alpha^*_K \tag{4.3}$$

（4.1）の計算の $\sum$ をとる順序を 1 次元目の計算から行っても結果は同じですから，2 次元目，3 次元目の項を固定した 1 次元変換の（4.2）の計算を K＝0,1,...,*N*1/2 に対して行えばよいわけです.

データ数が偶数の実変換の場合（4.2）の 1 次変換は，複素変換を利用することができます．詳細は“富士通 SSL II 使用手引書”の RFT の手法概要を参照してください．さらに 3 次元実フーリエ変換には次の関係があるため，これを利用して他の係数は求めることができます.

$$\alpha_{K1,K2,K3}=\alpha^*_{N1-K1,N2-K2,N3-K3} \tag{4.4}$$

③　素因子高速フーリエ変換

3 次元実フーリエ変換は，1 次元のフーリエ変換を複数組 3 回行うことで計算できます. このとき 1 次元のフーリエ変換を素因子高速フーリエ変換（a prime factor FFT）で行います．素因子高速フーリエ変換の説明については，VCPF3 の手法概要の②素因子高速フーリエ変換を参照してください．なお詳細については“付録　参考文献一覧表”の [6]，[46] を参照してください.

B71-14-0101 VSEVP,DVSEVP

実対称行列の固有値・固有ベクトル（三重対角化，マルチセクション法，逆反復法）
CALL VSEVP(A,K,N,NF,NL,IVEC,ETOL,CTOL,NEV,E, MAXNE,M, EV, VW,IW,ICON)

(1)　機能

実対称行列 $\boldsymbol{A}$ の指定された幾つかの固有値を求めます．指定に応じて，固有ベクトルを求めます．

実対称行列 $\boldsymbol{A}$ を Householder 変換で三重対角化し，マルチセクション法により固有値を求めます．固有ベクトルは逆反復法で求めます．

$$\boldsymbol{Ax}=\lambda\boldsymbol{x}$$

$\boldsymbol{A}$ は $n\times n$ 実対称行列です．

(2)　パラメタ

A.....入力．実対称行列 $\boldsymbol{A}$ を A(1:N,1:N)に格納します．
演算後の内容は保存されません．
A(K,N)なる 2 次元配列．

K.....入力．配列 A の整合寸法（≧ N）．

N.....入力．実対称行列 $\boldsymbol{A}$ の次数 n．

NF .....入力．固有値を小さい方から番号付けして（多重固有値には多重度分番号を割り当てる），求める最初の固有値の番号．

NL .....入力．固有値を小さい方から番号付けして（多重固有値には多重度分番号を割り当てる），求める最後の固有値の番号．

IVEC .....入力．制御情報．
1 のとき，固有値および対応する固有ベクトルの両方を求めます．
1 以外のとき，固有値のみ求めます．

ETOL .....入力．固有値が数値的に異なるか多重かを判定する判定値．
倍精度では 3.0D−16（単精度では 2.0D−7）以下のときこの値が標準値として設定されます．

CTOL .....入力．近接している固有値が近似的に多重かを式（3.1）を利用して判定する判定値．近似的多重固有値（cluster）に属する固有値の対応する固有ベクトルの 1 次独立性を保証するために使用されます．CTOL ≧ ETOL.
CTOL ＜ ETOL のときは，CTOL＝ETOL が標準値として設定されます．
（(3)使用上の注意 b. 注意①参照）．

NEV .....出力．固有値の個数に関する情報．
NEV(1)は，異なる固有値の個数．
NEV(2)は，異なる近似的多重固有値（cluster）の個数．
NEV(3)は，多重度も含んだ全ての固有値の個数．
NEV(3)なる 1 次元配列．

E.....出力．固有値が格納されます．
求められた固有値は E(1:NEV(3))に格納されます．

E(MAXNE)なる 1 次元配列.

MAXNE.....入力．計算できる固有値の最大個数．配列 E の大きさ.
NEV(3) ＞ MAXNE となったとき，固有ベクトルの計算はできません.
（(3)使用上の注意 b. 注意②参照）.

M.....出力．求められた固有値の多重度に関する情報.
M(*i*,1)は *i* 番目の固有値λ_iの多重度を，M(*i*,2)は *i* 番目の固有値λ_iに関して，近接している固有値を近似的多重固有値（cluster）と見なしたときの多重度を示します.
M(MAXNE,2)なる 2 次元配列.
（(3)使用上の注意 b. 注意①参照）.

EV.....出力．IVEC＝1 のとき，固有値に対応して固有ベクトルが格納されます.
求められた固有ベクトルは，EV(1:N,1:NEV(3))に格納されます.
EV(K, MAXNE)なる 2 次元配列.

VW.....作業領域．大きさ 17 × K なる 1 次元配列.

IW.....作業領域．9× K＋128 なる 1 次元配列.

ICON.....出力．コンディションコード.
表 VSEVP-1 参照

表 VSEVP-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし	―
20000	密集固有値の計算中，固有値の総数が MAXNE を超えた.	処理を打ち切る．固有ベクトルを求めることはできないが，異なる固有値自体は求められている．（(3)使用上の注意 b. 注意②参照）.
30000	NF ＜ 1,NL ＞ N,NL ＜ NF,N ＜1, K ＜ N，MAXNE ＜ NL-NF+1	処理を打ち切る.
30100	入力行列が対称行列でない可能性がある.	

(3)　使用上の注意

a.　使用する副プログラム

SSLII・・・UIBBS, UIBFC, UIBFE, UIBSL, UITBS, UITFC, UITFE, UITSL,
USEVP, UTDEX, UTDEY, UTMLS, UTRBK, UTRVM, UZRDM,
MGSSL, UMGSL, UMGSL2

b.　注意

①　本ルーチンは，多重固有値(cluster)に対して特別の考慮を払っています.
ε＝ETOL としたとき，連続する固有値λ_j, $j=s-1,s,...,s+k$,(k≧0)に対して，

$$\frac{\left|\lambda_i - \lambda_{i-1}\right|}{1 + \max\left(\left|\lambda_{i-1}\right|, \left|\lambda_i\right|\right)} \le \varepsilon \qquad (3.1)$$

$i=s,s+1,...,s+k$ となる i に対して(3.1)を満たし，$i=s-1$ および $i=s+k+1$ について(3.1)を満たさないとき，これらの固有値λ_j, $j=s-1,s,...,s+k$ は数値的に多重であるとみなされます．

ETOL の標準値は倍精度で 3.0D-16（単精度では 2.0D-7）～丸め誤差の単位であり，このとき固有値は計算機で求め得る精度まで分離されます．

(3.1)が ε ＝ETOL に対して成立しないとき，λ_{i-1} およびλ_i は異なる固有値と考えます．

ε ＝ETOL で異なる固有値とみなされた連続する固有値λ_m,$m=t-1,t,...,t+k$,(k≧0)に対して，ε ＝CTOL としたとき，$i=t,t+1,...,t+k$ について(3.1)を満たし，$i=t-1$ および $i=t+k+1$ について(3.1)を満たさないとき, これらの異なる固有値 $\lambda_m m=t-1,t,...,t+k$ を近似的に多重（cluster）と見なします．これは，cluster に対応する不変部分空間，つまり 1 次元独立な固有ベクトルを計算するために利用されます．つまり，対応する固有ベクトルは直交する初期ベクトルを用いて計算され，再直交化されます．もちろん，CTOL≧ETOL を満たさなければなりません．この条件を満たさないとき．CTOL は ETOL の値と等しく設定されます．

② r 個数の固有値を求めるとき，最初または最後の固有値が，1 より大きい多重度を持つとき，r 個以上の固有値が求まります．対応する固有ベクトルは増えた固有値も含んで, 対応する固有ベクトルの格納域が足りるときのみ計算されます．

MAXNE に，計算できる固有値の最大個数を指定できます．計算する固有値の総数が MAXNE を超えた場合，ICON＝20000 を返却します．このとき，固有ベクトルの計算を行うよう指定されていたら固有ベクトルの計算はできません．固有値は求められていますが, 多重度分だけ繰り返して格納はされてはいません．つまり，固有値に関しては，求められた異なる固有値が，E(1:NEV(1))に，および対応する固有値の多重度が M(1:NEV(1),1)にそれぞれ格納されています．

固有値がすべて異なり，近似的多重な固有値もない場合，MAXNE は求める固有値の数で十分です．計算する固有値の総数が MAXNE を超えた場合，計算を続けるために必要な MAXNE の値は NEV(3)に返却されます．

③ 本ルーチンは SEIG1, SEIG2, VSEG2 より高速です．

c. 使用例

固有値・固有ベクトルが分かっている実対称行列の指定された固有値・固有ベクトルを求めます．

```
C     ** EXAMPLE PROGRAM **
      IMPLICIT REAL*8(A-H,O-Z)
      PARAMETER (K=500,N=K,NF=1,NL=100,MAXNE=NL-NF+1)
      PARAMETER (NVW=15*K,NIW=9*MAXNE+128)
      REAL*8    A(K,N),AB(K,N)
      REAL*8    E(K),EV(K,MAXNE),VW(NIW)
      REAL*8    VV(K,N)
      INTEGER   IW(NVW),M(MAXNE,2),NEV(3)
C
      ETOL=3.0D-16
```

```
      CTOL=5.0D-12
      NNF=NF
      NNL=NL
      IVEC=1
C Generate real symmetric matrix with known eigenvalues
C Initialization
      PI = 4.0D0*DATAN(1.0D0)
      DO 1 J=1,N
             DO 11 I=1,N
             VV(I,J)=DSQRT(2.0D0/DBLE(N+1))*SIN(DBLE(I)*PI
     +                  *DBLE(J)/ DBLE(N+1))
               A(I,J)=0.0D0
 11          CONTINUE
  1   CONTINUE
      DO 22 J=1,N
         A(J,J) = DBLE(-N/2+J)
 22   CONTINUE
      WRITE (6,*)' Input matrix size is ',N
      WRITE (6,*)' Matrix calculations use k =',K
      WRITE (6,*)' Desired eigenvalues are nf to nl ',NF,NL
      WRITE (6,*)' That is, request ',NL-NF+1,' eigenvalues.'
      WRITE (6,*)' True eigenvalues are as follows'
      WRITE (6,*)(A(J,J),J=NF,NL)
      CALL DVMGGM(A,K,VV,K,AB,K,N,N,N,ICON)
      CALL DVMGGM(VV,K,AB,K,A,K,N,N,N,ICON)
C Calculate the eigendecomposition of A
      CALL DVSEVP(A,K,N,NNF,NNL,IVEC,ETOL,CTOL,NEV,
     +                  E,MAXNE,M,EV,VW,IW,ICON)
      IF (ICON.GT.0) THEN
          WRITE (*,*)' VSEVP failed with parameter icon=',ICON
          STOP
      ENDIF
      WRITE (*,*)' Number of eigenvalues ',
     +                  NEV(3)
      WRITE (*,*)' Number of distinct eigenvalues ',
     +                  NEV(1)
      WRITE (*,*)' Solution to eigenvalues '
      WRITE (*,*)' E  ',(E(I),I=1,NEV(3))
 299  CONTINUE
      STOP
      END
```

(4)　手法概要

本ルーチンは，実対称行列を三重対角化して三重対角行列の固有値問題を解く方法に基づいています.

三重対行列の固有値問題は固有値をマルチセクション法で計算し，固有ベクトルを逆反復法で求めます．詳細は，VTDEV の記述および“付録 参考文献一覧表”の [33] を参照してください.

元の行列の固有ベクトルは，三重対角行列の固有ベクトルからなる行列に三重対角化を行う変換行列を掛けて求めます.

A22-72-0101 VSPLL,DVSPLL

正値対称行列の LL^T 分解（ブロック化されたコレスキー分解法）
CALL VSPLL(A,K,N,EPSZ,ICON)

(1)　機能

$n \times n$ の正値対称行列 $\boldsymbol{A}$ を，ブロック化された外積形のコレスキー分解法により LL^T 分解します．

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^T$$

ただし，$\boldsymbol{L}$ は下三角行列です．

(2)　パラメタ

A.....入力．係数行列 $\boldsymbol{A}$.
　　出力．行列 $\boldsymbol{L}$.
　　入力時には，A(1:N,1:N)の下三角部分{A(i, j) | $i \geqq j$}に，$\boldsymbol{A}$ の下三角部分{a_{ij} | $i \geqq j$}を格納します．
　　出力時には，A(1:N,1:N)の下三角部分{A(i, j) | $i \geqq j$}に下三角行列 $\boldsymbol{L}${l_{ij} | $i \geqq j$}が格納されます．(図 VSPLL-1 参照)
　　A(K,N)なる 2 次元配列.
K.....入力．配列 A の 1 次元目の大きさ．(≧N)
N.....入力．係数行列 $\boldsymbol{A}$ の次数 n.
EPSZ .....入力．ピボットの相対零判定値（≧0.0）．
　　0.0 のときは標準値が採用されます．
　　（(3) 使用上の注意 b.注意①参照）．
ICON.....出力．コンディションコード．
　　表 VSPLL-1 参照.

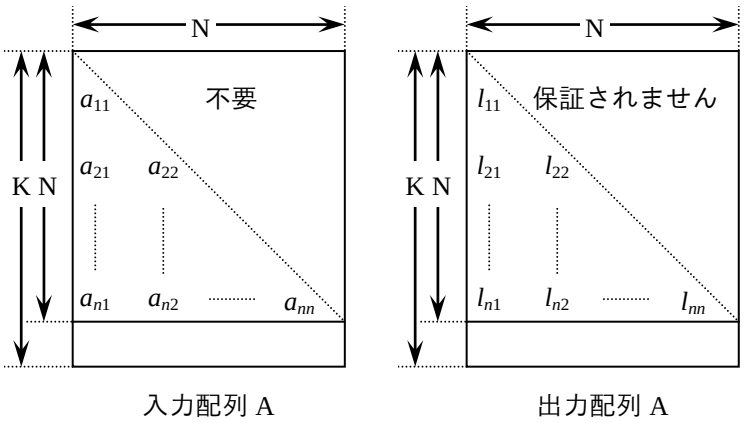


図 VSPLL-1　コレスキー分解法のデータの格納方法

LL^T 分解を行う正値対称行列の対角要素および下三角部分 a_{ij} を配列 A(i, j), $i = j,...,n, j=1,...,n$

に格納します.
LL^T分解された結果は，下三角行列 ***L*** を下三角部分に格納されます.

表 VSPLL-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	ー
20000	ピボットが相対的に零となった．係数行列は非正則の可能性が強い.	処理を打ち切る.
20100	ピボットが負となった．係数行列は正値でない.	
30000	N＜1, EPSZ＜0, K＜N	

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ AMACH,UBLL,UBLL2,UBLLX,UBLLX2,UBLLX3

b.　注意

① ピボットの相対零判定値にある値を設定したとすると, この値は次の意味を持っています．すなわち，コレスキー分解法による LL^T 分解の過程で，ピボットの値が正でありその値より小さくなった場合に，そのピボットの値を相対的に零と見なし，ICON＝20000 として処理を打ち切ります．EPSZ の標準値は丸め誤差の単位を *u* としたとき，EPSZ＝16*u* です.
なお，ピボットの値が正であり小さくなっても，処理を続行させたい場合は，EPSZ に極小の値を与えれば良いのですが，結果の精度は保証されません.

② 分解の途中でピボットが負となった場合，係数行列は正値ではありません．このとき，ICON＝20100 として処理を打ち切ります.

③ 係数行列の行列式の値を求めるときは，演算後の配列 A の *n* 個の対角要素を掛け合わせ，その 2 乗をとって下さい.

c.　使用例

2000×2000 の行列を LL^T 分解します.

```
C      **EXAMPLE**
       IMPLICIT REAL*8(A-H,O-Z)
       PARAMETER (N=2000,K=N+1)
       REAL*8    A(K,N)
C
       DO J=1,N
       DO I=J,N
       A(I,J)=MIN(I,J)
       ENDDO
       ENDDO

C
```

```
      CALL DVSPLL(A,K,N,0.0d0,ICON)
      WRITE(6,610) ICON
      IF(ICON.GE.20000) GO TO 10
C
      S=1.D0
      DO I=1,N
      S=S*A(I,I)
      ENDDO
C
      DET=S*S
      WRITE(6,620) DET
      WRITE(6,640)
      DO J=1,5
      WRITE(6,600) J,(A(I,J),I=J,5)
      ENDDO
  600 FORMAT(/10X,I5/(10X,3D23.16))
  610 FORMAT(/10X,5HICON=,I5)
  620 FORMAT(/10X
     *,22HDETERMINANT OF MATRIX=,D23.16)
  640 FORMAT(/10X,17HDECOMPOSED MATRIX)
   10 STOP
      END
```

(4)　手法概要

ブロック化された外積形のコレスキー分解法については，“付録 1 参考文献一覧表”の[14]を参照して下さい．

A22-72-0301 VSPLX,DVSPLX

LL^T分解された正値対称行列の連立 1 次方程式
CALL VSPLX(B,FA,KFA,N,ICON)

(1)　機能

LL^T分解された正値対称な係数行列を持つ連立 1 次方程式,

$$\boldsymbol{LL}^T = \boldsymbol{b}$$

を解きます．ただし $\boldsymbol{L}$ は $n \times n$ の下三角行列，$\boldsymbol{b}$ は n 次元の実定数ベクトル，$\boldsymbol{x}$ は n 次元の解ベクトルとします．また，$n \geqq 1$ とします．

本サブルーチンは，サブルーチン VSPLL により，LL^T分解された行列を受けて，求解処理を行います．

(2)　パラメタ

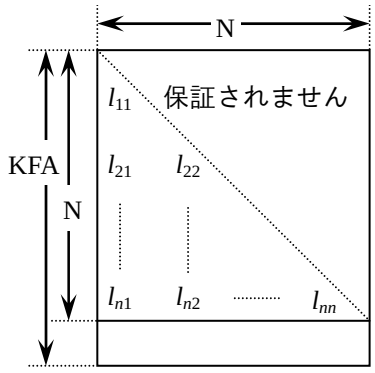
B.....入力．定数ベクトル $\boldsymbol{b}$.
　　出力．解ベクトル $\boldsymbol{x}$.
　　B(N)なる 1 次元配列.

FA .....入力．LL^T分解された行列 $\boldsymbol{L}$ を格納します.
　　FA(1:N,1:N)の下三角部分{FA(i, j) | $i \geqq j$}に，LL^T分解された下三角行列 $L\{l_{ij} \mid i \geqq j\}$を格納します．(図 VSPLX-1 参照)
　　FA(KFA,N)なる 2 次元配列.

KFA .....入力．配列 FA の整合寸法．(≧N)

N.....入力．行列 $\boldsymbol{L}$ の次数 n.

ICON.....出力．コンディションコード.
　　表 VSPLX-1 参照.



配列 FA

図 VSPLX-1　配列 FA への行列 $\boldsymbol{L}$ を格納する方法

LL^T分解された結果は，下三角行列 $\boldsymbol{L}$ を下三角部分に格納します.

表 VSPLX-1　コンディションコード

コード	意　　味	処理内容
0	エラーなし.	ー
20000	係数行列が非正則であった.	処理を打ち切る.
30000	N＜1, KFA＜N	

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ なし.

b.　注意

①　本サブルーチンを，サブルーチン VSPLL に続けて呼び出すことにより，正値対称な係数行列を持つ連立 1 次方程式を解くことができます．しかし，通常は，サブルーチン VLSPX を呼び出せば，一度に解が求められます.

c.　使用例

2000×2000 の行列を，LL^T 分解して連立 1 次方程式を解きます.

```
C      **EXAMPLE**
       IMPLICIT REAL*8(A-H,O-Z)
       PARAMETER (N=2000,KFA=N+1)
       REAL*8 A(KFA,N)
       REAL*8 B(N)
C
       DO J=1,N
       DO I=J,N
       A(I,J)=MIN(I,J)
       ENDDO
       ENDDO

       DO I=1,N
       B(I)=I*(I+1)/2+I*(N-I)
       ENDDO
C
       CALL DVSPLL(A,KFA,N,0.0D0,ICON)
       WRITE(6,610) ICON
       IF(ICON.GE.20000) GO TO 10

       CALL DVSPLX(B,A,KFA,N,ICON)
       WRITE(6,630) (B(I),I=1,10)

       S=1.0D0
       DO I=1,N
       S=S*A(I,I)
```

```
      ENDDO

      DET=S*S
      WRITE(6,620) DET
      GO TO 10
  610 FORMAT(/10X,5HICON=,I5)
  620 FORMAT(/10X
     *,34HDETERMINANT OF COEFFICIENT MATRIX=
     *,D23.16)
  630 FORMAT(/10X,15HSOLUTION VECTOR
     *//(10X,3D23.16))
   10 STOP
      END
```

(4)　手法概要

LL^T分解された係数行列の連立 1 次方程式を，前進代入および後進代入で解きます.

(“付録 1 参考文献一覧表”の[14]を参照)

F17-11-0301 VSRFT,DVSRFT

1 次元，多重離散型実フーリエ変換（2，3 および 5 の混合基底）
CALL VSRFT(X,M,N,ISIN,ISN,W,ICON)

(1)　機能

1 次元離散型実フーリエ変換を多重（m 組）に行います.

変換するデータの大きさ n は 2，3，5 巾の積として表される数でなければなりません.

①　変換

$\{x_{k1j2}\}$ を入力し，(1.1) で定義する変換を行い，$\{n\alpha_{k1k2}\}$ を求めます.

$$n\alpha_{k1k2} = \sum_{j2=0}^{n-1} x_{k1j2} \times \omega_n^{-j2k2r} \qquad (1.1)$$

$$\omega_n = \exp(2\pi i/n),$$

変換の方向は，r＝1 または r＝－1 を指定できます.

$k1$＝0,1,..., m－1,

$k2$＝0,1,..., n－1

②　逆変換

$\{\alpha_{k1k2}\}$ を入力し，(1.2) で定義する変換を行い，$\{x_{k1j2}\}$ を求めます.

$$x_{k1j2} = \sum_{k2=0}^{n-1} \alpha_{k1k2} \times \omega_n^{-j2k2r} \qquad (1.2)$$

$$\omega_n = \exp(2\pi i/n),$$

逆変換では，変換で指定した方向と逆向きを指定しなければなりません.

r = –1 または r = 1

k_1 = 0,1,..., m –1,

j_2 =0,1,..., n –1

実フーリエ変換の結果には複素共役の関係があるため $\{n\alpha_{k1k2}\}$ の k_2 の最初の n/2 + 1 個を格納します．また m，n の一方は偶数でなければなりません.

(2)　パラメタ

X.....ISN＝1（実数から複素数への変換）のとき

入力．X(1：m,1：n) に実データ $\{x_{k1j2}\}$ を格納します.

出力．X(1：m,1：n/2＋1)に $\{n\alpha_{k1k2}\}$ の実部が X(1：m，n/2＋2：2×(n/2＋1)) に $\{n\alpha_{k1k2}\}$ の虚部が格納されます.

$k_1 =$ 0,1,..., m－1,

$k_2 =$ 0,1,..., n/2

ISN＝ –1（複素数から実数への変換）のとき

入力．X(1：m,1：n/2＋1)に $\{\alpha_{k1k2}\}$ の実部を X(1：m,n/2＋2：2×(n/2＋1)) に $\{\alpha_{k1k2}\}$ の虚部を格納します．

$k_1 =$ 0,1,..., m－1,

$k_2 =$ 0,1,..., n/2

出力．X(1:m, 1:n)に実データ $\{x_{k1j2}\}$ が格納されます．

X(m,(n＋4×int ($\sqrt{n/2}$)))なる 2 次元配列．

X(m,2×(n/2＋1)＋1：n＋4×int ($\sqrt{n/2}$))は，内部で使われるため演算後結果は保証されません．

M.....入力．1 次元離散型実フーリエ変換を行う多重度（データの本数）m．m，n の一方は偶数でなければなりません．

N.....入力．1 次元離散型実フーリエ変換を行うデータの大きさ n．n は 2，3，5 の巾乗の積で表せる数．m，n の一方は偶数でなければなりません．

ISIN.....入力．フーリエ変換の方向 r を示します．

1 のとき　r＝1

-1 のとき　r＝-1

ISN.....入力．

1 のとき　変換（実数から複素数）

-1 のとき　逆変換（複素数から実数）

W.....作業域．

大きさ 2×n＋m×(n＋4×int ($\sqrt{n/2}$))なる 1 次元配列．

ICON.....出力．コンディションコード．

表 VSRFT-1 参照．

表 VSRFT-1 コンディションコード

コード	意　味	処理内容
0	エラーなし．	ー
30001	M≦0 または N≦0	処理を打ち切る．
30008	N が 2，3，5 の巾の積であらわされる整数ではない．	
30016	ISIN≠1 かつ ISIN≠-1	
30032	ISN≠1 かつ ISN≠-1	
30512	M，N がともに奇数である．	

(3)　使用上の注意

a.　使用する副プログラム

SSL II . . . UASSM,UCTSV,USEPR,UFIX,UJOIN,USPLT,UUFIX,USTUP,
UCFS,UCF16,UCFT2,UCFT3,UCFT4,UCFT5,UCFT8,UCFMRW,UCRU,
UCTRF,MGSSL

b.　注意

①　一般的なフーリエ変換の定義

多重離散型フーリエ変換および, 逆変換は一般的に (3.1) , (3.2) で定義されます.

$$\alpha_{k1k2} = \frac{1}{n}\sum_{j2=0}^{n-1} x_{k1j2}\omega_n^{-j2k2} \qquad (3.1)$$

$$, k_1 = 0,1,..., m-1$$

$$, k_2 = 0,1,..., n-1$$

$$x_{j1j2} = \sum_{k2=0}^{n-1} \alpha_{j1k2}\omega_n^{j2k2} \qquad (3.2)$$

$$, j_1 = 0,1,..., m-1$$

$$, \mathrm{j}_2 = 0,1,..., n-1$$

ここで,

$$\omega_n = \exp(2\pi \mathrm{i}/\mathrm{n})$$

本サブルーチンでは, (1.1), (1.2) の左辺に対応して $n\alpha_{k1k2}$ または x_{j1j2} を求めます. 従って，結果の正規化は必要に応じて行ってください.

②　多重離散型実フーリエ変換には以下の複素共役の関係があります.

$$\alpha_{k1\ \ k2} = \overline{\alpha_{k1\ n-k2}}$$

$$k_1 = 0,..., m-1$$

$$, k_2 = 1,..., n/2$$

この関係を使って残りの部分を計算することができます.

③　n が偶数のときと m が偶数のときとで，別の方法を適用しています．n が偶数のとき，ベクトル長は約 $m\sqrt{n}$ です．m が偶数であるときのベクトル長は $m/2$ ですが，データ転送量はより少ないです.

m が十分大きくかつ偶数のとき本ルーチンは最も高速です.

c. 使用例

多重度 m＝500 の 1 次元実 FFT を計算します.

```
C     **EXAMPLE**
      IMPLICIT REAL*8(A-H,O-Z)
      PARAMETER(M=500,N=2**10)
      PARAMETER(N2=N+4*22)
      DIMENSION X(M,N2),W(2*N+M*N2)
C
      DO 100 J=1,M
      DO 100 I=1,N
      X(J,I)=FLOAT(I)+FLOAT(N)*(J-1)
  100 CONTINUE
C
      ISIN=1
      ISN= 1
C
C     REAL TO COMPLEX TRANSFORM
C
      CALL DVSRFT(X,M,N,ISIN,ISN,W,ICON)
      PRINT*,'ICON=',ICON
C
      ISIN=-1
      ISN=-1
C
C     COMPLEX TO REAL TRANSFORM
C
      CALL DVSRFT(X,M,N,ISIN,ISN,W,ICON)
      PRINT*,'ICON=',ICON
C
      ERROR=0.0D0
      DO 200 J=1,M
      DO 200 I=1,N
      ERROR=MAX(ABS(X(J,I)/N-
     &      (FLOAT(I)+FLOAT(N)*(J-1))),ERROR)
  200 CONTINUE
C
      PRINT*,'ERROR=',ERROR
      STOP
      END
```

B71-12-0101 VTDEV,DVTDEV

実 3 重対角行列の固有値・固有ベクトル
CALL VTDEV(D,SL,SU,N,NF,IVEC,ETOL,CTOL,NEV,E,MAXNE,EV,K, M,VW,IVW,ICON)

(1)　機能

実 3 重対角行列の指定された固有値を求めます．指定に応じて対応する固有ベクトルを求めます．

$$\boldsymbol{Tx} = \lambda \boldsymbol{x} \tag{1.1}$$

3 重対角行列 $\boldsymbol{T}$ は，次の条件を満たすものとします．

$$l_i u_{i-1} > 0,\ i = 2,..,\ n \tag{1.2}$$

ここで，3 重対角行列 $\boldsymbol{T}$ の要素 t_{ij} としたとき，d_i は対角要素を，
$l_i = t_{i,\,i-1}$，$u_i = t_{i,\,i+1}$，は副対角要素を表します．ただし $l_1 = u_n = 0$.

$$(Tv)_i = l_i v_{i-1} + d_i v_i + u_i v_{i+1},\quad i = 1,2,..,\ n \tag{1.3}$$

T は n 次元の実 3 重対角行列です．

(2)　パラメタ

D.....入力．D（N）なる 1 次元配列で対角要素 d_i を格納します．

SL.....入力．SL（N）なる 1 次元配列で下副対角要素 l_i を SL（2:N）に格納します．
SL（1）＝0.

SU .....入力．SU（N）なる 1 次元配列で上副対角要素 u_i を SU（1:N－1）に格納します．SU（N）＝0.

N.....入力．3 重対角行列の次数 n.

NF .....入力．固有値を小さい方から番号付けして (多重固有値には多重度分番号を割り当てる)，求める固有値の最初の番号．固有値は NF から NF＋NEV(1)-1 まで求めます．
出力．求める最初の固有値が多重固有値である場合も考慮した，求められた最初の固有値の番号．

IVEC .....入力．制御情報．
1 のとき　固有値および固有ベクトルの両方を求めます．
1 以外のとき 固有値のみ求めます．

ETOL .....入力．固有値が数値的に異なるか多重かを式（3.4）を利用して判定する判定値．
倍精度では 3.0D－16（単精度では 2.0D－7）以下のときこの値が標準値として設定されます．
（(3)使用上の注意 b.注意②参照）．

CTOL.....入力．近接している固有値が近似的に多重かを式（3.4）を利用して判定する判定値．CTOL≧ETOL．
CTOL＜ETOL のときは，CTOL＝ETOL が標準値として設定されます．
（(3)使用上の注意 b.注意②参照）．

NEV .....入力．求める固有値の個数に関する情報．
NEV（1）は，求める固有値の個数．
出力．求められた固有値の個数に関する情報．
NEV（3）なる 1 次元配列．
NEV（1）は，異なる固有値の個数．
NEV（2）は，異なる近似的多重固有値（cluster）の個数．
NEV（3）は，多重度も含んだ全ての固有値の個数．

E.....出力．固有値が格納されます．
求められた固有値は E（1:NEV (3)）に格納されます．
E（MAXNE）なる 1 次元配列．

MAXNE.....入力．計算できる固有値の最大個数．配列 E の大きさ．
NEV (3) ＞MAXNE のとき，固有ベクトルの計算はできません．
（(3)使用上の注意 b.注意③参照）．

EV .....出力．IVEC＝1 のとき，おのおのの固有値に対応して固有ベクトルが格納されます．求められた固有ベクトルは EV（1:N,1:NEV (3)）に格納されます．
EV（K,MAXNE）なる 2 次元配列．

K.....入力．EV の 1 次元目の大きさ(≧N).

M.....出力．求められた固有値の多重度に関する情報．
M（*i*,1）は *i* 番目の固有値λ_iの多重度を，M（*i*,2）は *i* 番目の固有値λ_iに関して，近接している固有値を近似的多重固有値（cluster）と見なしたときの多重度を示します．
（(3)使用上の注意 b.注意③参照）．
M（MAXNE,2）なる 2 次元配列．

VW.....作業領域．大きさ 12×N なる 1 次元配列．

IVW .....作業領域．大きさ 9×MAXNE＋128 なる 1 次元配列．

ICON.....出力．コンディションコード．
表 VTDEV-1 参照．

表 VTDEV-1　コンディションコード

コード	意　味	処理内容
0	エラーなし.	ー
20000	多重固有値の計算中, 固有値の総数が MAXNE を超えた.	処理を打ち切る．固有ベクトルを求めることはできないが，異なる固有値自体は求められています. ((3)使用上の注意 b.注意③参照）.
30000	N＜1，K＜1，NF＜1, NEV（1）＜1, NF＋NEV（1）＞N，N＞K	処理を打ち切る.
30100	SL（i）×SU（i-1）≦0,行列が対称化できなかった.	

(3)　使用上の注意

a.　使用する副プログラム

SSL II・・・ UTMLS,UZRDM,UTDEY,UTDEX,UIBBS,UIBFC,UIBFE,UIBSL,
UITBS,UITFC,UITFE,UITSL,AMACH,MGSSL

b.　注意

①　本機能を利用して解ける問題.

本機能は，$l_i u_{i-1}>0$ のみを要求します．このため，(3.1) の一般化固有値問題を $\boldsymbol{T}\leftarrow\boldsymbol{TD}^{-1}$ と置き換えて解くことができます.

$$\boldsymbol{Tx}=\boldsymbol{\lambda}\boldsymbol{Dv} \tag{3.1}$$

ここで，$D>0$ の対角行列です.

また, T に対する固有値問題を対称な一般化固有値問題に変換することもできます.

$$(\boldsymbol{DT}-\boldsymbol{\lambda}\boldsymbol{D})\,v=0 \tag{3.2}$$

ここで，$D_1=1$，$D_i=u_{i-1}D_{i-1}/l_i$，$i=2,...,n$ です.

D_i がスケーリング問題を引き起こす可能性があるとき，以下の対称な問題を考えることが好まれます.

$$(\boldsymbol{D}^{1/2}\boldsymbol{TD}^{-1/2}-\boldsymbol{\lambda}\boldsymbol{I})w=0 \tag{3.3}$$

このとき固有ベクトルの間には $w=\boldsymbol{D}^{1/2}v$ なる関係があります.

②　本ルーチンは，多重固有値(cluster)に対して特別の考慮を払っています.

ε＝ETOL としたとき，連続する固有値 λ_j，$j=s-1,s,...,s+k,(k\geqq 0)$ に対して,

$$\frac{\left|\lambda_i-\lambda_{i-1}\right|}{1+\max\left(\left|\lambda_{i-1}\right|,\left|\lambda_i\right|\right)}\leq\varepsilon \tag{3.4}$$

$i=s,s+1,...,s+k$ となる i に対して（3.4）を満たし，$i=s-1$ および $i=s+k+1$ について（3.4）を満たさないとき，これらの固有値 λ_j，$j=s-1,s,...,s+k$ は数値的に多重であるとみなされます.

ETOL の標準値は倍精度では 3.0D－16（単精度では 2.0D－7）～丸め誤差の単位であり，このとき固有値は計算機で求め得る精度まで分離されます.

(3.4)がε＝ETOL に対して成立しないとき，λ_{i-1}およびλ_iは異なる固有値と考えます．

ε＝ETOL で異なる固有値とみなされた連続する固有値 $\lambda_m, m=t-1,t,\ldots,t+k$,（$k \geqq 0$）に対して，ε＝CTOL としたとき，$i=t,t+1,\ldots,t+k$ について（3.4）を満たし，$i=t-1$ および $i=t+k+1$ について（3.4）を満たさないとき，これらの異なる固有値 $\lambda_m, m=t-1,t,\ldots,t+k$ を近似的に多重（cluster）と見なします．これは，cluster に対応する不変部分空間，つまり 1 次独立な固有ベクトルを計算するために利用されます．つまり，対応する固有ベクトルは直交する初期ベクトルを用いて計算され，再直交化されます．もちろん，CTOL≧ETOL を満たさなければなりません. この条件を満たさないとき. CTOL はETOL の値と等しく設定されます．

③ r 個数の固有値を求めるとき，最初または最後の固有値が，1 より大きい多重度を持つとき，r 個以上の固有値が求まります．対応する固有ベクトルは増えた固有値も含んで, 対応する固有ベクトルの格納域が足りるときのみ計算されます．MAXNE に，計算できる固有値の最大個数を指定できます．固有値の総数が MAXNE を超えた場合，ICON＝20000 を返却します．このとき，固有ベクトルの計算を行うよう指定されていたら固有ベクトルの計算はできません．固有値は求められていますが，多重度分だけ繰り返して格納はされてはいません．つまり，固有値に関しては，求められた異なる固有値が，E（1:NEV (1)）に，および対応する固有値の多重度が M（1:NEV（1）,1）にそれぞれ格納されています．固有値がすべて異なり，近似的多重な固有値もない場合，MAXNE は入力として NEV（1）に指定する求める固有値の総数で十分です．

c. 使用例

数値的多重固有値を持つことが知られている Wilkinson による変形（“付録　参考文献一覧表”の [45] 参照）に基づくモデル問題の ne 個の固有値と対応する固有ベクトルを計算する問題を解きます．

```
C     **EXAMPLE**
      IMPLICIT  REAL*8 (A-H,O-Z)
      INTEGER K,P1,Q1,N,N0,N1,MAX_CLUS,NE,MAXNE,NVW,NIVW
      REAL*8 ETOL,CTOL
      PARAMETER (K=1000)
      PARAMETER (P1=350,Q1=2,N=P1*Q1,N0=584,N1=686,NE=N1-N0+1)
      PARAMETER (MAX_CLUS=2*Q1,MAXNE=NE+MAX_CLUS)
      PARAMETER (NVW=12*N,NIVW=9*MAXNE+128)
C
      REAL*8 SL(N),D(N),SU(N),E(MAXNE),EV(K,MAXNE),VW(NVW)
      INTEGER M(MAXNE,2),NEV(3),IVW(NIVW),NF,ICON,NEVAL,I
     &        ,J,KK,IVEC
      LOGICAL EVAL_OUTPUT,DBG_OUTPUT
C
      ETOL=3.D-16
      CTOL=5.D-12
      IVEC=1
```

```
C
C     Blocked W ^+_n (Wilkinson):  Pathologically close
C     eigenvalues in each p1 x p1 (p1 odd, small) block,
C     with q1 blocks so that multiplicity of largest
C     eigenvalues is 2*q1.  If maxnev <2*q1 then error
C     condition 20000 is obtained.
C
      J = ( P1 + 1 ) / 2
      D(J) = 0.D0
      DO 10 I=1,J-1                    ! first block
         SL(I+1)  = 1.D0
         SU(I)    = 1.D0
         SL(J+I)  = 1.D0
         SU(J+I-1) = 1.D0
         D(I)     = FLOAT(J-I)
         D(2*J-I) = D(I)
   10 CONTINUE
      SL(1)  = 0.D0
      SU(P1) = 0.D0
      DO 20 KK=2,Q1                    ! subsequent blocks
         II = (KK-1) * P1
         DO 20 I=1,P1
            SL(II+I) = SL(I)
            SU(II+I) = SU(I)
            D(II+I) = D(I)
   20 CONTINUE
      SL(1) = 0.D0
      SU(N) = 0.D0
      NF     = N0
      NEV(1) = NE
      ICON = 0
      CALL DVTDEV(D,SL,SU,N,NF,IVEC,ETOL,CTOL,NEV,E,MAXNE
     &            ,EV,K,M,VW,IVW,ICON)
C
      DBG_OUTPUT = .FALSE.
      IF( ICON .EQ. 20000 ) DBG_OUTPUT = .TRUE.
      EVAL_OUTPUT = .TRUE.
      IF ( ICON .EQ. 30000 .OR. ICON .EQ. 30100 )
     &   EVAL_OUTPUT = .FALSE.
      IF ( EVAL_OUTPUT ) THEN
         NEVAL = NF
         WRITE(*,*)' ICON = ',ICON
```

```
            II=1
            DO 30 J=1,NEV(1)
                  WRITE(*,900) NEVAL,E(II),M(J,1)
               IF ( DBG_OUTPUT ) THEN
                  II = II + 1
               ELSE
                  II = II + M(J,1)
               ENDIF
               NEVAL = NEVAL + M(J,1)
   30       CONTINUE
         ENDIF
C
  900 FORMAT(' EIGENVALUE(',I5,')=',E25.18,2X,
     &       ' WITH MULTIPLICITY=',I5)
C
         STOP
         END
```

(4)　手法概要

Sturm カウントに基づいたアルゴリズムを用いています．区間を最低 3 つの区間に分けて符号の変化を検出します（“付録　参考文献一覧表”の [37] 参照）．つまり，少なくとも 4×MAXNE 個の点で調べることになります．この値が計算におけるベクトル長を決定し，VPP シリーズの最低のベクトル長が 64 であるため，本ルーチンでは評価点の数を 4×MAXNE より大きな 64 の倍数に設定しています．

混成的なデータ構造が利用されます．固有値の順序付けおよび多重区間分けの両方の処理を行うために LIFO (*last in, first out*)リスト構造とベクトル化のための Array 構造が結合されています．これに関しては“付録　参考文献一覧表”の [33] を参照してください．

この処理は，ETOL で与えられる区間の細分化の限界まで行われます（“付録　参考文献一覧表”の [45] 参照）．標準値が選定されるとき，固有値の精度は行列のスケールに相対的な計算機精度となります．

Sturm カウントに関しては“付録　参考文献一覧表”の [10] を参照してください．

符号のカウントは IEEE 浮動小数点演算で，固有値をパラメタに持つ単調関数になる特性があります．

固有ベクトルは，逆反復法で計算されます．

数値的に多重（または，近似的に多重）な固有値が検出されたときを除いて，初期ベクトルは Sturm 列の符号構造を利用して決定されます．

数値的に多重（または，近似的に多重）な固有値に対してはランダムな初期ベクトルが追加生成され，クラスタの他の固有ベクトルと直行化されます．たいていの場合逆反復の反復回数は 1 回で十分です．クラスタ固有値に対応する固有ベクトルも逆反復の後に再直行化されます．

A72-25-0101 VTFQD,DVTFQD

非対称または不定値のスパース実行列の連立 1 次方程式 （TFQMR 法，対角形式格納法）
CALL VTFQD(A,K,NDIAG,N,NOFST,B,ITMAX,EPS,IGUSS,X,ITER,VW, ICON)

(1)　機能

$n \times n$ の非対称／不定値なスパース行列を係数行列とする連立 1 次方程式を転置なし準最小残差法（TFQMR:Transpose-Free Quasi-Minimal Residual method）で解きます．

$$\boldsymbol{Ax} = \boldsymbol{b}$$

$n \times n$ の係数行列は，対角形式の格納法で格納します．ベクトル $\boldsymbol{b}$ および $\boldsymbol{x}$ は n 次元ベクトルです．

反復計算が係数行列や初期ベクトルの特性のため続けられない場合（break down）があります．これは再帰的計算公式で，非零を期待される計算のある中間結果が零になるためです．この場合 break down を起こさない MGCR 法をお使い下さい．

反復解法の収束および利用指針に関しては，“第 I 部 概説 第 4 章 連立一次方程式の反復解法と収束性”を参照してください．

(2)　パラメタ

A.....入力．係数行列の非零要素を格納します．
A（K,NDIAG）なる 2 次元配列．A（1:N,NDIAG）に係数行列 $\boldsymbol{A}$ を対角形式で格納します．
対角形式の格納方法については，“第 I 部 概説 3.2.1.1 一般スパース行列の格納方法 b.一般スパース行列の対角形式の格納方法”を参照してください．

K.....入力．配列 A の整合寸法．

NDIAG.....入力．係数行列 $\boldsymbol{A}$ の非零要素を含む対角ベクトル列の総数．
配列 A の 2 次元目の大きさ．

N.....入力．行列 $\boldsymbol{A}$ の次数 n．

NOFST.....入力．A に格納される対角ベクトルに対応した主対角ベクトルからの距離を格納します．上対角ベクトル列は正，下対角ベクトル列は負の値で表します．
NOFST（NDIAG）なる 1 次元配列．

B.....入力．大きさ n の 1 次元配列．連立 1 次方程式の右辺の定数ベクトルを格納します．

ITMAX .....入力．TFQMR 法の反復回数の上限値（＞0）．2000 程度で十分です．

EPS.....入力．収束判定に用いられる判定値．
EPS が 0.0 以下のとき，EPS は倍精度ルーチンでは 10^{-6} が，単精度ルーチンでは 10^{-4} が設定されます．
（(3)使用上の注意 b.注意①参照）．

IGUSS.....入力．配列 X に指定された解ベクトルの近似値から反復計算を行うかを指定する制御情報．
0 のとき　解ベクトルの近似値を指定しません．
0 以外のとき 配列 X に指定された解ベクトルの近似値から反復計算を開始します．

X.....入力．大きさ *n* の 1 次元配列．解ベクトルの近似値を指定することができます．
出力．解ベクトルが格納されます．

ITER.....出力．TFQMR 法での実際の反復回数．

VW.....作業域．大きさ，K×10＋N＋NBANDL＋NBANDR なる 1 次元配列．
NBANDL は下バンド幅，NBANDR は上バンド幅の大きさ．

ICON.....出力．コンディションコード．
表 VTFQD-1 参照．

表 VTFQD-1　コンディションコード

コード	意　味	処理内容
0	エラーなし．	ー
20000	break down を起こした．	処理を打切る．
20001	反復回数の上限に達した．	処理を打ち切る．配列 X には，そのときまでに得られている近似値を出力するが，精度は保証できない．
30000	K＜1，N＜1，K＜N， NDIAG＜1，K＜NDIAG， または ITMAX≦0	処理を打ち切る．
32001	｜NOFST（I）｜＞N-1	

(3)　使用上の注意

a.　使用する副プログラム

SSL II･･･ AMACH,URGWD,URIPA,URITI,URITT,URMVD,USSCP,URSTE,
USVCN,USVCP,USVSU,USVUP,USVN2,URELT,MGSSL,UTFQD,
UTFQR,UQBBM

b.　注意

① TFQMR 法は，残差のユーグリッドノルムが最初の残差のユーグリッドノルムと EPS の積以下になったとき収束したと見なします．正確な解と求められた近似解の誤差はほぼ行列 ***A*** の条件数と EPS の積に等しくなります．

② 対角形式を使う上での注意
係数行列 ***A*** の外側の対角ベクトルの要素は零を設定する必要があります．
対角ベクトル列を配列 A に格納する順序に制限は特にありません．
この方法の利点は，行列ベクトル積が間接指標を使わずに計算できる点です．
対角構造を持たない行列は効率よく格納できないという点が欠点です．

c.　使用例

領域［0,1］×［0,1］×［0,1］でディリクレ境界条件（境界で関数値は零）のもとで，“第 I 部 概説 3.2.2 偏微分演算子の離散化とその格納例”に記述された偏微分演算子を離散化して得られる係数行列を持つ連立 1 次方程式を解きます．

INIT_MAT_DIAG は，“第 I 部 概説 3.2.2 偏微分演算子の離散化とその格納例”を参照してください．GET_BANDWIDTH_DIAG はバンド幅見積りのルーチン，INIT_SOL は，求めるべき解ベクトルを乱数で生成するルーチンです．

```
C     **EXAMPLE**
      PROGRAM TEST_ITER_SOLVERS
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER MACH
      PARAMETER (MACH = 0)
      PARAMETER (K = 10000)
      PARAMETER (NX = 20,NY = 20,NZ = 20,N = NX*NY*NZ)
      PARAMETER (NDIAG = 7,NVW = 10*K+N+400+400)
      REAL*8 A(K,NDIAG),X(N),B(N),SOLEX(N),VW(NVW)
      INTEGER NOFST(NDIAG)
C
      CALL INIT_SOL(SOLEX,N,1D0,MACH)
      PRINT*,'EXPECTED SOLUSIONS'
      PRINT*,'X(1) = ',SOLEX(1),' X(N) = ',SOLEX(N)
C
      PRINT *
      PRINT *,'    TFQMR METHOD'
      PRINT *,'    DIAGONAL FORMAT'
C
      VA1 = 3D0
      VA2 = 1D0/3D0
      VA3 = 5D0
      VC = 1.0
      XL = 1.0
      YL = 1.0
      ZL = 1.0

      CALL INIT_MAT_DIAG(VA1,VA2,VA3,VC,A,NOFST
     &          ,NX,NY,NZ,XL,YL,ZL,NDIAG,N,K)
      CALL GET_BANDWIDTH_DIAG(NOFST,NDIAG,NBANDL,NBANDR)
         DO 110 I = 1,N
            VW(I+NBANDL) = SOLEX(I)
 110     CONTINUE
      CALL DVMVSD(A,K,NDIAG,N,NOFST,NBANDL,VW,B,ICON)
      PRINT*,'DVMVSD ICON=',ICON
```

```
C
      EPS = 1D-10
      IGUSS = 0
      ITMAX = 2000
      CALL DVTFQD(A,K,NDIAG,N,NOFST,B,ITMAX
     &            ,EPS,IGUSS,X,ITER,VW,ICON)
C
      PRINT* ,'ITER = ',ITER
      PRINT* ,'DVTFQD ICON = ',ICON
      PRINT*, 'COMPUTED VALUES'
      PRINT*, 'X(1) = ',X(1),' X(N) = ',X(N)
      STOP
      END
```

(4)　手法概要

TFQMR 法は“付録　参考文献一覧表”の [12] を参照してください.

A72-26-0101 VTFQE,DVTFQE

非対称または不定値のスパース実行列の連立 1 次方程式 （TFQMR 法，ELLPACK 形式格納法）
CALL VTFQE(A,K,IWIDT,N,ICOL,B,ITMAX,EPS,IGUSS,X,ITER,VW, ICON)

(1)　機能

$n \times n$ の非対称／不定値なスパース行列を係数行列とする連立 1 次方程式を転置なし準最小残差法（TFQMR:Transpose-Free Quasi-Minimal Residual method）で解きます.

$$\boldsymbol{Ax} = \boldsymbol{b}$$

$n \times n$ の係数行列は，ELLPACK 形式の格納法で格納します．ベクトル $\boldsymbol{b}$ および $\boldsymbol{x}$ は n 次元ベクトルです.

反復計算が係数行列や初期ベクトルの特性のため続けられない場合(break down)があります．これは再帰的計算公式で，非零を期待される計算のある中間結果が零になるためです．この場合 break down を起こさない MGCR 法をお使いください.

反復解法の収束および利用指針に関しては，“第 I 部 概説 第 4 章 連立一次方程式の反復解法と収束性”を参照してください.

(2)　パラメタ

A.....入力．係数行列の非零要素を格納します.
A（K,IWIDT）なる 2 次元配列.
ELLPACK 形式の格納方法については，“第 I 部 概説 3.2.1.1 一般スパース行列の格納方法”を参照してください.

K.....入力．A および ICOL の整合寸法（≧n）.

IWIDT.....入力．係数行列 $\boldsymbol{A}$ の非零要素の行ベクトル方向の最大個数.
A および ICOL の 2 次元目の大きさ.

N.....入力．行列 A の次数 n.

ICOL .....入力．ELLPACK 形式で使用される列指標で，A の対応する要素がいずれの列ベクトルに属すかを示します.
ICOL（K,IWIDT）なる 2 次元配列.

B.....入力．大きさ n の 1 次元配列．連立 1 次方程式の右辺の定数ベクトルを B に格納します.

ITMAX .....入力．TFQMR 法の反復回数の上限値（＞0）．2000 程度で十分です.

EPS.....入力．収束判定に用いられる判定値.
EPS が 0.0 以下のとき，EPS は倍精度ルーチンでは 10^{-6} が，単精度ルーチンでは 10^{-4} が設定されます.
（(3) 使用上の注意 b.注意①参照）.

IGUSS.....入力．配列 X に指定された解ベクトルの近似値から反復計算を開始するかを示す制御情報.

0 のとき　解ベクトルの近似値を指定しません.
0 以外のとき 配列 X に指定された解ベクトルの近似値から反復計算を開始します.

X.....入力. 大きさ *n* の 1 次元配列. 解ベクトルの近似値を指定することができます.
出力. 解ベクトルが格納されます.

ITER.....出力. TFQMR 法での実際の反復回数.

VW.....作業域. 大きさ, K×13 なる 1 次元配列.

ICON.....出力. コンディションコード.
表 VTFQE-1 参照.

表 VTFQE-1　コンディションコード

コード	意　味	処理内容
0	エラーなし.	ー
20000	break down を起こした.	処理を打切る.
20001	反復回数の上限に達した.	処理を打ち切る. 配列 X には, そのときまでに得られている近似値を出力するが, 精度は保証できない.
30000	K ＜ 1, N ＜ 1, K ＜ N, IWIDT ＜ 1, K ＜ IWIDT, または ITMAX ≦ 0	処理を打ち切る.

(3)　使用上の注意

a.　使用する副プログラム

SSL II . . . AMACH,URIPA,URITI,URITT,URMVE,USSCP,URSTE,USVCN,
USVCP,USVSU,USVUP,USVN2,URELT,MGSSL,UTFQE,UTFQR,
UQBBM

b.　注意

① TFQMR 法は, 残差のユーグリッドノルムが最初の残差のユーグリッドノルムと EPS の積以下になったとき収束したと見なします.
正確な解と求められた近似解の誤差はほぼ行列 ***A*** の条件数と EPS の積に等しくなります.

c.　使用例

領域［0,1］×［0,1］×［0,1］でディリクレ境界条件（境界で関数値は零）のもとで, “第 I 部 概説 3.2.2 偏微分演算子の離散化とその格納例”に記述された偏微分演算子を離散化して得られる係数行列を持つ連立 1 次方程式を解きます. INIT_MAT_ELL は, “第 I 部 概説 3.2.2 偏微分演算子の離散化とその格納例”を参照してください. INIT_SOL は, 求めるべき解ベクトルを乱数で生成するルーチンです.

```
C      **EXAMPLE**
       PROGRAM TEST_ITER_SOLVERS
       IMPLICIT REAL*8 (A-H,O-Z)
       PARAMETER (MACH = 0)
       PARAMETER (K = 10000)
```

```
      PARAMETER (NX = 20,NY = 20,NZ = 20,N = NX*NY*NZ)
      PARAMETER (IWIDT = 7,NVW = K*13)
      REAL*8 A(K,IWIDT),X(N),B(N),SOLEX(N),VW(NVW)
      INTEGER ICOL(K,IWIDT)
C
      CALL INIT_SOL(SOLEX,N,1D0,MACH)
      PRINT*,'EXPECTED SOLUTION'
      PRINT*,'X(1) = ',SOLEX(1),' X(N) = ',SOLEX(N)
      PRINT*
      PRINT*,'     TFQMR METHOD'
      PRINT*,'     ELLPACK FORMAT'
C
      VA1 = 3D0
      VA2 = 1D0/3D0
      VA3 = 5D0
      VC = 5D0
      XL = 1.0
      YL = 1.0
      ZL = 1.0
C
      CALL INIT_MAT_ELL(VA1,VA2,VA3,VC,A,ICOL
     &          ,NX,NY,NZ,XL,YL,ZL,IWIDT,N,K)
      CALL DVMVSE(A,K,IWIDT,N,ICOL,SOLEX,B,ICON)
      PRINT*,'DVMVSE ICON = ',ICON
C
      EPS = 1D-10
      IGUSS = 0
      ITMAX = 800
      CALL DVTFQE(A,K,IWIDT,N,ICOL,B,ITMAX
     &           ,EPS,IGUSS,X,ITER,VW,ICON)
C
      PRINT*,'DVTFQE ICON = ',ICON
      PRINT*,'COMPUTED VALUE'
      PRINT*,'X(1) = ',X(1),' X(N) = ',X(N)
      STOP
      END
```

(4)　手法概要

TFQMR 法は“付録　参考文献一覧表”の [12] を参照してください.

F18-11-0101 VWFLT,DVWFLT

ウェーブレットフィルターの生成
CALL VWFLT(F,N,ICON)

(1) 機能

コンパクトな台を持つ次数 *n* の Daubechies のウェーブレットに対応するフィルターを生成します．次数 *n* は 2,4,6,12 および 20 のフィルターが生成できます．

(2) パラメタ

F .....出力．大きさ 2×N の 1 次元配列． 変換で使われるウェーブレットフィルターが格納されます．
（(3) 使用上の注意 b.注意①参照）．

N.....入力．ウェーブレットフィルターの係数の数．
2,4,6,12 または 20 のいずれかの数．

ICON.....出力．コンディションコード．
表 VWFLT-1 参照．

表 VWFLT-1　コンディションコード

コード	意　味	処理内容
0	エラーなし．	ー
30000	N が 2,4,6,12,20 のいずれの数でもない．	処理を打切る．

(3) 使用上の注意．

a. 使用する副プログラム

なし．

b. 注意

① フィルターの条件

一般に本機能で使われる直交フィルター(orthogonal filter)は，長さ 2×N のベクトルで，F（1）,...,F（N）は，Low-pass フィルターを，F（N＋1）,...,F（2×N）は High-pass フィルターを含んでいます．これらの係数には，以下の関係があります．

$$\sum_{i=1}^{n} F(i)^2 = 1, F(2N+1-i) = (-1)^i F(i), i = 1,\ldots, N$$

詳細は，“付録　参考文献一覧表”の [7] , [9] を参照．

c. 使用例

大きさ *n*＝1024 のデータの 1 次元ウェーブレット変換および逆変換を行います．

```
C     **EXAMPLE**
      IMPLICIT REAL*8(A-H,O-Z)

C     -------- Constants ------------
      INTEGER MaxK, MaxSize
      PARAMETER (MaxK = 20,MaxSize = 1024)

C     -------- Variables and formats ------------
      INTEGER N,K,i,ISN,ICON,LS
      REAL*8 X(1:MaxSize),T(1:MaxSize),Y(1:MaxSize),
     &       F(1:2*MaxK),
     &       ireal,Emax,diff,temp,Xmax,Erel

C     -------- Generate input ------------
      N  = 1024
      K  = 6
      LS = 3

      DO 100 i= 1,N
        ireal = i
        temp = 0.5 - abs(ireal/N - 0.5)
        X(i) = temp                    ! Input vector
        T(i) = temp                    ! Reference vector
 100  CONTINUE

C     ---------------  Initialize filter ------
      CALL DVWFLT(F,K,ICON)

C     ---------------- Transform Data ----------------

      ISN=1
      CALL DV1DWT(X,N,Y,ISN,F,K,LS,ICON)
      IF (ICON .NE. 0) THEN
        PRINT*,'ERROR IN 1D Wavelet Transform,ICON = ',ICON
        STOP
      ENDIF

C     ---------------- Transform back ---------------

      ISN=-1
      CALL DV1DWT(X,N,Y,ISN,F,K,LS,ICON)
      IF (ICON .NE. 0) THEN
```

```
        PRINT*,'ERROR IN Inverse of 1D Wavelet Transform,'
     &        ,'ICON = ',ICON
        STOP
       ENDIF

C      ------ Verify result ------

       Emax = 0.0
       Xmax = 0.0
       DO 200 i=1,N
         diff = abs(X(i)-T(i))
         IF (diff .GT. Emax) Emax = diff
         IF (abs(X(i)) .GT. Xmax) Xmax = abs(X(i))
 200   CONTINUE

       Erel = Emax/Xmax
       IF (Erel .GT. 1.0e-4) THEN
         PRINT*,'Relative Max error (FWT):',Erel
         STOP
       END IF
       PRINT*,'1D Wavelet Transform OK'

       STOP
       END
```

F18-12-0101 V1DWT,DV1DWT

1 次元ウェーブレット変換
CALL V1DWT(X,N,Y,ISN,F,K,LS,ICON)

(1)　機能

1 次元ウェーブレット変換および逆変換を行います．基底関数は，コンパクトな台を持つ直交ウェーブレットを利用します．

(2)　パラメタ

X.....入力／出力．X（N）なる 1 次元配列．変換するベクトルデータを格納します．変換のとき（ISN＝1）入力となり，逆変換のとき（ISN＝－1）出力となります．

N.....入力．変換されるデータの長さ．2 の巾で表せる正整数．
（(3) 使用上の注意 b.注意①参照）．

Y.....出力／入力．Y（N）なる 1 次元配列．変換されたベクトルデータが格納されます．変換のとき（ISN＝1）出力となり，逆変換のとき（ISN=－1）入力となります．
（(3) 使用上の注意 b.注意②参照）．

ISN.....入力．変換か逆変換かを指定します．
変換:ISN＝1
逆変換:ISN＝－1

F.....入力. 大きさ 2×K の 1 次元配列. 変換で使われるウェーブレットフィルターが格納されます．VWFLT を本ルーチンに先立って呼び出し，1 次元ウェーブレット変換で使われるフィルターの係数を設定する必要があります．
（(3) 使用上の注意 b.注意③参照）．

K.....入力．ウェーブレットフィルターの係数を表す正の偶数．

LS.....入力．各列に対する変換の深さを表す正整数．$N \geqq 2^{LS}$．$N = 2^{LS}$ のとき，完全なウェーブレット変換が行われます．

ICON.....出力．コンディションコード．
表 V1DWT-1 参照．

表 V1DWT-1　コンディションコード

コード	意　味	処理内容
0	エラーなし.	－
30000	ISN ≠ 1 かつ ISN ≠-1	処理を打ち切る.
30002	N ＜ 2	
30004	N が 2 の巾で表せない数である.	
30008	K が偶数でない. または，LS ＜ 0，LS ＞ $\log_2 N$	

(3) 使用上の注意

a. 使用する副プログラム

SSL II . . . UWFT1,UWFI1,UWPT1,UWVI1,UWPI1,UWVT1,MGSSL

b. 注意

① 変換するデータの大きさが 2 の巾で表せないときは，残りをゼロ詰めしてそれより大きな 2 の巾で表せる大きさ N のデータとして利用できます.

② 変換結果の格納について

入力 1 次元データのベクトル v_j に対して, ウェーブレット変換の各段の High-pass フィルターの結果は，v_j(N×2^{-i}＋1:N×2^{-i+1})，i = 1,...,LS に格納されます．第 1 段の部分的なウェーブレット変換の High-pass フィルターの出力結果は，Y（N/2＋1:N,M/2＋1:M）に格納されています.

③ フィルターの条件

一般に本機能で使われる直交フィルター（orthogonal filter）は，長さ 2×K のベクトルで，F（1）,...,F（K）は，Low-pass フィルターを，F（K＋1）,...,F（2×K）は High-pass フィルターを含んでいます．これらの係数には，以下の関係があります.

$$\sum_{i=1}^{K} \mathrm{F}(i)^2 = 1,\ \mathrm{F}(2\mathrm{K}+1-i) = (-1)^i \mathrm{F}(i),\ i = 1,\ldots,\mathrm{K}$$

詳細は，“付録　参考文献一覧表”の [7] , [9] を参照.

c. 使用例

大きさ n＝1024 のデータの 1 次元ウェーブレット変換および逆変換を行います.

```
C     **EXAMPLE**
      IMPLICIT REAL*8(A-H,O-Z)

C     -------- Constants ------------
      INTEGER MaxK, MaxSize
      PARAMETER (MaxK = 20,
     &           MaxSize = 1024)

C     -------- Variables and formats ------------
      INTEGER N,K,i,ISN,ICON,LS
      REAL*8 X(1:MaxSize),T(1:MaxSize),Y(1:MaxSize),
     &       F(1:2*MaxK),
     &       ireal,Emax,diff,temp,Xmax,Erel
C     -------- Generate input ------------
      N  = 1024
      K  = 6
      LS = 3

      DO 100 i= 1,N
        ireal = i
```

```
        temp = 0.5 - abs(ireal/N - 0.5)
        X(i) = temp                     ! Input vector
        T(i) = temp                     ! Reference vector
 100  CONTINUE

C     ---------------  Initialize filter ------
      CALL DVWFLT(F,K,ICON)

C     ---------------- Transform Data ----------------

      ISN=1
      CALL DV1DWT(X,N,Y,ISN,F,K,LS,ICON)
      IF (ICON .NE. 0) THEN
        PRINT*,'ERROR IN 1D Wavelet Transform,ICON = ',ICON
        STOP
      ENDIF

C     ---------------- Transform back ---------------

      ISN=-1
      CALL DV1DWT(X,N,Y,ISN,F,K,LS,ICON)
      IF (ICON .NE. 0) THEN
        PRINT*,'ERROR IN Inverse of 1D Wavelet Transform,'
     &        ,'ICON = ',ICON
        STOP
      ENDIF

C     ------ Verify result ------

      Emax = 0.0
      Xmax = 0.0
      DO 200 i=1,N
        diff = abs(X(i)-T(i))
        IF (diff .GT. Emax) Emax = diff
        IF (abs(X(i)) .GT. Xmax) Xmax = abs(X(i))
 200  CONTINUE
        Erel = Emax/Xmax
        IF (Erel .GT. 1.0e-4) THEN
          PRINT*,'Relative Max error (FWT):',Erel
          STOP
        END IF
```

```
PRINT*,'1D Wavelet Transform OK'

STOP
END
```

(4)　手法概要

長さ N のベクトル *s*（普通は信号）の部分的なウェーブレット変換は, Low-pass フィルターと High-pass フィルターを適用することで達成できます. 部分ベクトル $w_1,...,w_{n/2}$ は, Low-pass フィルターを *s* に適用した結果であり，$w_{n/2+1},...,w_n$ は High-pass フィルターを *s* に適用した結果です.

ウェーブレット変換は，Low-pass フィルターの結果を含む部分ベクトルに対して，部分的なウェーブレット変換を $\log_2(n)$ 回再帰的に適用することです．適用するたびに，その前の結果の半分のデータを使います.

最初のステップの計算量が支配的で，O（K×N）のオーダです．K は使うウェーブレット変換の次数であり，N は変換するベクトルの長さです.

本機能は，周期的なデータを扱うことができます．非周期的なデータに適用しますと，不自然な不連続性が境界に現れ，変換に影響を与えます．この影響を最小化するために，フーリエ変換で使われる技術（補間，鏡映データ詰め）を使うことができます.

ウェーブレット変換の入門資料として“付録 参考文献一覧表”の [15] , [40] を，更なる応用については“付録 参考文献一覧表”の [36] を，より深い扱いについては“付録 参考文献一覧表”の [7] , [9] を参照してください.

F18-13-0101 V2DWT,DV2DWT

2 次元ウェーブレット変換
CALL V2DWT(X,M,N,Y,ISN,F,K,LSX,LSY,ICON)

(1)　機能

2 次元ウェーブレット変換および逆変換を行います．基底関数は，コンパクトな台を持つ直交ウェーブレットを利用します．

(2)　パラメタ

X.....入力／出力．X（M,N）なる 2 次元配列．変換する 2 次元行列データを格納します．変換のとき（ISN=1）入力となり，逆変換のとき（ISN=－1）出力となります．

M.....入力．変換されるデータの行の数．2 の巾で表せる正整数．
（(3) 使用上の注意 b.注意①参照）．

N.....入力．変換されるデータの列の数．2 の巾で表せる正整数．
（(3) 使用上の注意 b.注意①参照）．

Y.....出力／入力．Y（N,M）なる 2 次元配列．変換された結果が転置された形で格納されます．変換のとき (ISN=1) 出力となり，逆変換のとき (ISN=－1) 入力となります．
（(3) 使用上の注意 b.注意②参照）．

ISN.....入力．変換か逆変換かを指定します．
変換:ISN=1
逆変換:ISN=－1

F.....入力. 大きさ 2×K の 1 次元配列. 変換で使われるウェーブレットフィルターが格納されます．VWFLT を本ルーチンに先立って呼び出し，2 次元ウェーブレット変換で使われるフィルターの係数を設定する必要があります．
（(3)使用上の注意 b.注意③参照）．

K.....入力．ウェーブレットフィルターの係数の数．

LSX.....入力．各列に対する変換の深さを表す正整数．$M \geqq 2^{LSX}$．$M = 2^{LSX}$ のとき，完全なウェーブレット変換が行われます．

LSY.....入力．各列に対する変換の深さを表す正整数．$N \geqq 2^{LSY}$．$N = 2^{LSY}$ のとき，完全なウェーブレット変換が行われます．

ICON.....出力．コンディションコード．
表 V2DWT-1 参照．

表 V2DWT-1　コンディションコード

コード	意　味	処理内容
0	エラーなし.	ー
30000	ISN ≠ 1 かつ ISN ≠ -1	処理を打ち切る.
30002	M ＜ 2 または N ＜ 2	
30004	M または N が 2 の巾で表せない数である.	
30008	K が偶数でない. または，LSX＜0，LSX＞$\log_2$M, LSY＜0，LSY＞$\log_2$N	

(3)　使用上の注意

a.　使用する副プログラム

SSL II . . . UWFT2,UWFI2,UWPT2,UWVI2,UWPI2,UWTRP,UWVT2,MGSSL

b.　注意

①　変換するデータの各次元の大きさが 2 の巾で表せないときは，残りの列と行にゼロを詰めてそれより大きな 2 の巾で表せる次元の大きさ（M,N）を持つデータとして利用できます.

②　変換結果の格納について

入力 2 次元データの列ベクトル c_j および行ベクトル r_j に対して,ウェーブレット変換の各段の High-pass フィルターの結果は,それぞれ c_j (M×2^{-i}＋1:M×2^{-i+1})，i＝1,...,LSX および r_j(N×2^{-i}＋1:N×2^{-i+1}),i＝1,...,LSY に格納されます. 2 次元ウェーブレット変換の結果は，転置されて配列 Y に格納されています.

第 1 段の部分的なウェーブレット変換の High-pass フィルターの出力結果は, Y(N/2＋1:N,M/2＋1:M) に格納されています.

③　フィルターの条件

一般に本機能で使われる直交フィルター（orthogonal filter) は，長さ 2×K のベクトルで，F（1），...，F（K）は，Low-pass フィルターを，F（K＋1），...,F（2×K）は High-pass フィルターを含んでいます．これらの係数には，以下の関係があります.

$$\sum_{i=1}^{K} \mathrm{F}(i)^2 = 1, \quad \mathrm{F}(2\mathrm{K}+1-i) = (-1)^i \mathrm{F}(i), \quad i = 1,...,\mathrm{K}$$

詳細は，“付録　参考文献一覧表”の [7] , [9] を参照.

c.　使用例

2 次元データ（1024×512）の 2 次元ウェーブレット変換および逆変換を行う.

```
C     **EXAMPLE**
      IMPLICIT REAL*8(A-H,O-Z)

C     -------- Constants ------------
```

```
      INTEGER MaxK,MaxSize
      PARAMETER (MaxK = 20,MaxSize = 512*1024)

C     -------- Variables and formats ------------
      INTEGER M,N,K,i,row,index2D,ISN,ICON,LSX,LSY
      REAL*8 X(1:MaxSize),T(1:MaxSize),Y(1:MaxSize),
     &       F(1:2*MaxK),
     &       ireal,Emax,diff,temp,Xmax,Erel

C     -------- Generate input ------------
      M   = 1024
      N   = 512
      K   = 6
      LSX = 3
      LSY = 4

      DO 99 row = 1,M
        DO 100 i= 1,N
          ireal = i
          temp = 0.5 - abs(ireal/N - 0.5)
          ireal = row
          temp = temp + 0.5 - abs(ireal/M - 0.5)
          index2D = row + (i-1)*M
          X(index2D) = temp              ! Input vector
          T(index2D) = temp              ! Reference vector
 100    CONTINUE
 99   CONTINUE

C     ---------------  Initialize filter ------
      CALL DVWFLT(F,K,ICON)

C     ---------------- Transform Data ----------------

      ISN=1
      CALL DV2DWT(X,M,N,Y,ISN,F,K,LSX,LSY,ICON)
      IF (ICON .NE. 0) THEN
        PRINT*,'ERROR IN 2D Wavelet Transform,ICON = ',ICON
        STOP
      ENDIF

C     ---------------- Transform back ---------------
```

```
      ISN=-1
      CALL DV2DWT(X,M,N,Y,ISN,F,K,LSX,LSY,ICON)
      IF (ICON .NE. 0) THEN
        PRINT*,'ERROR IN Inverse of 2D Wavelet Transform,'
     &        ,'ICON = ',ICON
        STOP
      ENDIF

C     ------ Verify result ------

      Emax = 0.0
      Xmax = 0.0
      DO 199 row =1,M
        DO 200 i=1,N
          index2D = row + (i-1)*M
          diff = abs(X(index2D)-T(index2D))
          IF (diff .GT. Emax) Emax = diff
          IF (abs(X(index2D)) .GT. Xmax)
     &       Xmax = abs(X(index2D))
 200    CONTINUE
 199  CONTINUE

      Erel = Emax/Xmax
      IF (Erel .GT. 1.0e-4) THEN
        PRINT*,'Relative Max error (FWT):',Erel
        STOP
      END IF
      PRINT*,'2D Wavelet Transform OK'

      STOP
      END
```

(4)　手法概要

長さ N のベクトル *s*（普通は信号）の部分的なウェーブレット変換は,Low-pass フィルターと High-pass フィルターを適用することで達成できます. 部分ベクトル $w_1,...,w_{n/2}$ は, Low-pass フィルターを *s* に適用した結果であり，$w_{n/2+1},...,w_n$ は High-pass フィルターを *s* に適用した結果です.

ウェーブレット変換は，Low-pass フィルターの結果を含む部分ベクトルに対して，部分的なウェーブレット変換を $\log_2$（*n*）回再帰的に適用することです．適用するたびに，その前の結果の半分のデータを使います.

最初のステップの計算量が支配的で，*O*（K×N）のオーダです．K は使うウェーブレット変換の次数であり，N は，変換するベクトルの長さです.

2 次元変換の場合，ウェーブレット変換は，各列に深さ LSX 適用され，その後結果の行列の各行に深さ LSY 適用されます．
本機能は，周期的なデータを扱うことができます．非周期的なデータに適用しますと，不自然な不連続性が境界に現れ，変換に影響を与えます．この影響を最小化するために，フーリエ変換で使われる技術（補間，鏡映データ詰め）を使うことができます．
ウェーブレット変換の入門資料として“付録　参考文献一覧表”の [15] , [40] を，2 次元の変換については“付録　参考文献一覧表”の [36] を，より深い扱いについては“付録　参考文献一覧表”の [7] , [9]を参照してください．

付録　参考文献一覧表

[1] S.L.Anderson,
Random number generators on vector supercomputers and other advanced architectures, SIAM Rev. 32 (1990), 221-251.

[2] R.P.Brent,
Uniform random number generators for supercomputers, Proc. Fifth Australian Supercomputer Conference, Melbourne, Dec. 1992, 95-104.

[3] R.P.Brent,
Uniform random number generators for vector and parallel computers, Report TRCS-92-02, Computer Sciences Laboratory, Australian National University, Canberra, March 1992.

[4] R.P.Brent,
Fast normal random number generators on vector processors, Technical Report TRCS-93-04, Computer Sciences Laboratory, Australian National University, Canberra, March 1993.

[5] R.P.Brent,
A Fast Vectorised Implementation of Wallace's Normal Random Number Generator, Technical Report, Computer Sciences Laboratory, Australian National University, To appear.

[6] C.Sidney Burrus,Fellow,IEEE,and Peter W. Eschenbacher,
“An In Place,In-Order Prime Fact or FFT Algorithm”,IEEE Trans. on Acoust.,Speech, Signal Processing, vol.ASSP-29, No.4,pp.806-817,August 1981.

[7] C.K.Chui,
Adv. in Number. Analysis, vol.II, ch. Wavelets and Splines, Oxford Sci. Publ., 1992.

[8] J.K.Cullum and R.A.Willoughby,
“Lanczos algorithm for large symmetric eigenvalue computations”, Birkhauser, 1985.

[9] I.Daubechies,
Ten lectures on wavelets, SIAM, 1992.

[10] J.Demmel and W.Kahan,
Accurate singular values of bidiagonal matrices, SISSC 11, 873-912, 1990.

[11] A.M.Ferrenberg, D.P.Landau and Y.J.Wong,
Monte Carlo simulations: Hidden errors from “good” random number generators, Phys. Rev. Lett. 69 (1992), 3382-3384.

[12] R.Freund,
“A transpose-free quasi-minimal residual algorithm for nonhermitian linear systems”, SIAM J.Sci.Comput. 14 , 1993, pp.470-482.

[13] R.Freund and N.Nachtigal,

"QMR: a quasi minimal resudual method for non-Hermitian linear systems", Numer. Math. 60, 1991, pp.315-339.

[14] G.Golub and C.Van Loan,
Matrix computations, The Johns Hopkins University Press, Baltimore, 1989.

[15] A.Graps,
An introduction to wavelets, IEEE Computational Science and Engineering 9, 1995, no.2.

[16] M.H.Gutknecht
Variants of BiCGStab for matrices with complex spectrum,IPS Research report No. 91-14, 1991.

[17] Markus Hegland,
A self-sorting in-place fast Fourier transform algorithm suitable for vector and parallel processing, accepted for publication in Numerische Mathematik,1994

[18] M.Hegland,
On the parallel solution of tridiagonal systems by wrap-around partitioning and incomplete LU factorization , Numerische Mathematik 59, 1991, no.5, 453-472.

[19] Markus Hegland,
"An Implementation of Multiple and Multivariate Fourier Transforms on Vector Processors" appears in SIAM J. Sci. Comput., Vol. 16, No. 2, pp. 271-288, March 1995

[20] J.R.Heringa, H.W.J.Blöte and A.Compagner,
New primitive trinomials of Mersenne-exponent degrees for random-number generation, International J. of Modern Physics C 3 (1992), 561-564.

[21] M.R. Hestenes and E.Stiefel
Methods of conjugate gradients for solving linear systems. J. Res. Nat. Bur. Standards, 49: 409-435, 1952.

[22] F.James,
A review of pseudorandom number generators, Computer Physics Communications 60 (1990), 329-344.

[23] D. Kincaid, T.Oppe,
ITPACK on supercomputers , Numerical methods, Lecture Notes in Mathematics 1005 (1982).

[24] D. E. Knuth,
The Art of Computer Programming, Volume 2: Seminumerical Algorithms (second edition). Addison-Wesley, Menlo Park, 1981, Sec. 3.4.1, Algorithm P.

[25] Z. Leyk,
"Modified generalized conjugate residuals for nonsymmetric systems of linear equations" in "Proceedings of the 6th Biennial Conference on Computational Techniques and Applications: CTAC93", D. Stewart, H. Gardner and D. Singleton, eds., World Scientific, 1994, pp. 338-344. Also published as CMA Research Report CMA-MR33-93, Australian National University, 1993.

[26] Charles Van Loan
Computational Frameworks for the Fast Fourier Transform, SIAM, 1992.

[27] N. K. Madsen, G. H. Rodrigue, and J. I. Karush,
"Matrix multiplication by diagonals on a vector/parallel processor", Infomation Processing Letters, vol. 5, 1976, pp. 41-45

[28] G.Marsaglia,

A current view of random number generators, Computer Science and Statistics: The Interface (edited by L.Billard), Elsevier Science Publishers B.V. (North-Holland), 1985, 3-10.

[29] R.S.Martin, G.Peters and J.H.Wilkinson,
Symmetric Decomposition of A Positive Definite Matrix, Linear Algebra, Handbook for Automatic Computation, Vol.2, pp.9-30, Springer-Verlag, Berlin-Heidelberg-New York, 1971

[30] T. Oppe, W.Joubert and D Kincaid,
An overview of NSPCG: a nonsymmetric preconditioned conjugate gradient package, Computer Physics communications 53 p283 (1989).

[31] T. C. Oppe and D. R. Kincaid,
"Are there iterative BLAS?", Int. J. Sci. Comput. Modeling (to appear).

[32] J.Ortega,
Introduction to parallel and vector solution of linear systems, Plenum Press, 1988

[33] M.R.Osborne,
Computing the eigenvalues of tridiagonal matrices on parallel vector processors, Mathematics Research Report No. MRR 044-94, Australian National University, 1994.

[34] J. R. Rice and R. F. Boisvert,
Solving Elliptic Problems Using ELLPACK, Springer-Verlang, New York, 1985.

[35] Y. Saad and M. H. Schultz,
"GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems", SIAM J. Sci. Stat. Comput. 7, 1986, p.856-869.

[36] D.H.Salesin, E.J.Stollnitz, T.D.DeRose,
Wavelets for computer graphics: A primer, part 1 and 2, IEEE Computer Graphics and Applications 15 (1995).

[37] H.D.Simon,
Bisection is not optimal on vector processors, SISSC 10, 205-209, 1989.

[38] G Sleijpen, D Fokkema,
BICG_STAB(L) for linear equations involving unsymmetric matrices with complex spectrum, Electronic Transactions on Numerical Analysis, Vol 1, p11-32, 1993 .

[39] G.L.G. Sleijpen, H.A. van der Vorst, and D.R. Fokkema.
BiCGSTAB(*l*) and other hybrid Bi-CG methods.Numerical Algorithms, 7:75-109, 1994.

[40] G.Strang, T.Nguyen,
Wavelets and filter banks, Wellesley-Cambridge Press, 1996

[41] Paul N. Swarztrauber,
Bluestein's FFTs for arbitrary N on the hypercube , Parallel Comput. 17 (1991), 607-617.

[42] H.A.Van Der Vorst
"BCG: A fast and smoothly converging variant of BI-CG for the solution of non-symmetric linear systems", SIAM J. Sci. Statist. Comput., 13 p631 1992

[43] C.S.Wallace
"Fast Pseudo-Random Generators for Normal and Exponential Variates", ACM Trans. on Mathematical Software 22 (1996), 119-127.

[44] R.Weiss
Parameter-Free Iterative Linear Solvers. Mathematical Research, vol. 97. Akademie Verlag, Berlin,

1996.

[45] J.H.Wilkinson,

The Algebraic Eigenvalue Problem, O.U.P., 1965.

[46] S. Winograd,

“On computing the discrete Fourier transform”, Math. Computation,. Vol.32, pp.175-199, Jan. 1978.

著作者氏名と著作物の索引

著作者氏名	サブルーチン名	項　　目
Richard Peirce Brent	DVRAN3	正規乱数の生成（倍精度）
Richard Peirce Brent	DVRAN4	正規乱数の生成（倍精度，Wallace 法）
Richard Peirce Brent Peter Frederick Price	DVRAU4	一様乱数［0,1］の生成（倍精度）
Andrew James Cleary	VBLDL	正値対称バンド行列の LDL^T 分解 （変形コレスキー分解）
	VBLDX	LDL^T 分解された正値対称バンド行列の連立 1 次方程式
	VBLU	実バンド行列の LU 分解 （ガウスの消去法）
	VLSBX	正値対称バンド行列の連立 1 次方程式 （変形コレスキー分解）
Murray Leslie Dow	VBCSD	非対称または不定値のスパース実行列の連立 1 次方程式 （BICGSTAB(*l*)法，対角形式格納法）
	VBCSE	非対称または不定値のスパース実行列の連立 1 次方程式 （BICGSTAB(*l*)法，ELLPACK 形式格納法）
	VCGD	正値対称スパース行列の連立 1 次方程式 （前処理付き CG 法，対角形式格納法）
	VCGE	正値対称スパース行列の連立 1 次方程式 （前処理付き CG 法, ELLPACK 形式格納法）
Markus Hegland Judith Helen Jenkinson Murray Leslie Dow	VMCFT	1 次元，多重，多次元離散型複素フーリエ変換（混合基底）
Markus Hegland Christopher Robert Dun	VLTQR	実 3 重対角行列の連立 1 次方程式 （QR 分解）

著作者氏名	サブルーチン名	項　　目
Margaret Helen Kahn	VHEVP	エルミート行列の固有値・固有ベクトル(三重対角化，マルチセクション法，逆反復法)
	VSEVP	実対称行列の固有値・固有ベクトル（三重対角化，マルチセクション法，逆反復法）
Jeoffrey Keating	VMRFT	多重，多次元離散型実フーリエ変換（2, 3 および 5 の混合基底）
	VSRFT	1 次元，多重離散型実フーリエ変換（2, 3 および 5 の混合基底）
Zbigniew Leyk	VCRD	非対称または不定値のスパース実行列の連立 1 次方程式（MGCR 法，対角形式格納法）
	VCRE	非対称または不定値のスパース実行列の連立 1 次方程式（MGCR 法，ELL-PACK 形式格納法）
	VMVSD	スパース実行列と実ベクトルの積（対角形式格納法）
	VMVSE	スパース実行列と実ベクトルの積（ELLPACK 形式格納法）
Zbigniew Leyk Murray Leslie Dow	VQMRD	非対称または不定値のスパース実行列の連立 1 次方程式（QMR 法，対角形式格納法）
	VQMRE	非対称または不定値のスパース実行列の連立 1 次方程式（QMR 法，ELLPACK 形式格納法）
	VTFQD	非対称または不定値のスパース実行列の連立 1 次方程式（TFQMR 法，対角形式格納法）
	VTFQE	非対称または不定値のスパース実行列の連立 1 次方程式（TFQMR 法，ELL-PACK 形式格納法）
Zbigniew Leyk David Lawrence Harrar II	VLAND	実対称スパース行列の固有値・固有ベクトル（Lanczos 法，対角形式格納法）
Ole Møller Nielsen	VWFLT	ウェーブレットフィルターの生成
Ole Møller Nielsen Markus Hegland Gavin John Mercer	V1DWT	1 次元ウェーブレット変換

著作者氏名	サブルーチン名	項　　目
Ole Møller Nielsen Markus Hegland	V2DWT	2 次元ウェーブレット変換
Michael Robert Osborne David Lawrence Harrar II	VTDEV	実 3 重対角行列の固有値・固有ベクトル

索　　引

FUJITSU

本マニュアルは、100%リサイクル可能な用紙を使用しています。