

第 11 章

Fourier 解析

名古屋大学 杉本 香菜子

数値シミュレーションによって得られた密度分布や速度場、磁場がどのような波数のゆらぎから成るかを調べるには、フーリエ変換が有効である。本章では、富士通 SSL II (Scientific Subroutine Library II)¹を使用した 3 次元離散型の高速実フーリエ変換の手法と、スペクトルの計算法について解説する。解析の方法と手順に焦点を絞るため、フーリエ変換や高速フーリエ変換 (FFT) アルゴリズムに関する一般的な解説は各種テキストに譲り (例えば、章末の参考文献を参照)、SSL II の FFT サブルーチンが利用できることを前提に話をすすめる。

実空間のスカラー分布 $F[\mathbf{x}]$ と波数空間での複素振幅分布 $A[\mathbf{k}]$ との間の 3 次元離散 Fourier 変換と逆フーリエ変換は以下のように表せる。

$$\begin{aligned} A[\ell_1, \ell_2, \ell_3] &= \frac{1}{N_1 N_2 N_3} \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} \sum_{j_3=0}^{N_3-1} F[j_1, j_2, j_3] \\ &\quad \times \exp\left[-i \frac{2\pi j_1 \ell_1}{N_1}\right] \exp\left[-i \frac{2\pi j_2 \ell_2}{N_2}\right] \exp\left[-i \frac{2\pi j_3 \ell_3}{N_3}\right], \end{aligned} \quad (11.1)$$

$$\begin{aligned} F[j_1, j_2, j_3] &= \sum_{\ell_1=0}^{N_1-1} \sum_{\ell_2=0}^{N_2-1} \sum_{\ell_3=0}^{N_3-1} A[\ell_1, \ell_2, \ell_3] \\ &\quad \times \exp\left[i \frac{2\pi \ell_1 j_1}{N_1}\right] \exp\left[i \frac{2\pi \ell_2 j_2}{N_2}\right] \exp\left[i \frac{2\pi \ell_3 j_3}{N_3}\right], \end{aligned} \quad (11.2)$$

ここで、 N_1, N_2, N_3 はそれぞれ x_1, x_2, x_3 方向の格子数を表す。実際の数値計算では、 $F[\mathbf{x}]$ は密度、圧力、運動エネルギーなどの物理量の実数スカラー配列であり、 $A[\mathbf{k}]$ はそれらを波数成分に分解した振幅を表す複素数配列である。

さらに、実フーリエ変換 ($F[\mathbf{x}]$ が実数関数) では以下のような複素共役関係が成立する。

$$A[\ell_1, \ell_2, \ell_3] = A^*[N_1 - \ell_1, N_2 - \ell_2, N_3 - \ell_3]. \quad (11.3)$$

¹富士通 SSL II は科学用サブルーチンライブラリであり、国立天文台データ解析計算センターでは、Fortran および C 言語 (C-SSL II) で使用可能である (2004 年度現在)。また、パーソナルコンピュータなどの逐次計算用に加えて、ベクトル計算機用の SSL II/VP (Fortran 用) および C-SSL II/VP (C 言語用)、ベクトル並列計算機用の SSL II/VPP (VPP Fortran 用) および SSL II/HPF (HPF 用) も利用できる。詳細については SSL II 使用手引書 (国立天文台データ解析計算センターのホームページ等に掲載) を参照。

ただし、 $\ell_i = 0$ ($i = 1, 2, 3$) の時は、 $N_i - \ell_i = 0$ とする。

一般に、(11.1) または (11.2) 式の離散フーリエ変換は、 $N_1^2 \times N_2^2 \times N_3^2$ 回の乗算を要するが、離散点の数 N_1, N_2, N_3 の値によっては計算量を大幅に減らすことが出来る。例えば N_i ($i = 1, 2, 3$) が 2 の中乗で表せるときは、演算の回数は $N_1 N_2 N_3 \log_2(N_1 N_2 N_3)$ に減らせる。この他に計算効率がよくできるのは N_i が小さな素数の中乗で素因数分解できるときで、このときの素因数は基底 (または基数) と呼ばれる。

11.1 数値計算ライブラリ SSLII による高速フーリエ変換

本節では、SSL II/VPP のサブルーチン DP_V3DRCF を例に、FFT の手順を解説する。SSL II のその他の実 FFT サブルーチンとその機能については、§11.4 の一覧を参照して欲しい。サブルーチン DP_V3DRCF は VPP Fortran 用に開発された 3 次元離散型実フーリエ変換ルーチン (2, 3 及び 5 の混合基底: $N_i = 2^p 3^q 5^r$ の場合に使用可) であり、VPP Fortran で記述されたプログラム内で call 文によって呼び出すと、フーリエ変換または逆フーリエ変換として次のような結果を返す。3 次元フーリエ変換は

$$N_1 N_2 N_3 A[\ell_1, \ell_2, \ell_3] = \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} \sum_{j_3=0}^{N_3-1} F[j_1, j_2, j_3] \times \exp\left[-i \frac{2\pi j_1 \ell_1}{N_1} r\right] \exp\left[-i \frac{2\pi j_2 \ell_2}{N_2} r\right] \exp\left[-i \frac{2\pi j_3 \ell_3}{N_3} r\right], \quad (11.4)$$

3 次元逆フーリエ変換は

$$F[j_1, j_2, j_3] = \sum_{\ell_1=0}^{N_1-1} \sum_{\ell_2=0}^{N_2-1} \sum_{\ell_3=0}^{N_3-1} A[\ell_1, \ell_2, \ell_3] \times \exp\left[i \frac{2\pi \ell_1 j_1}{N_1} r\right] \exp\left[i \frac{2\pi \ell_2 j_2}{N_2} r\right] \exp\left[i \frac{2\pi \ell_3 j_3}{N_3} r\right], \quad (11.5)$$

ここで、 $r = 1$ または -1 。従って、 $A[\ell_1, \ell_2, \ell_3]$ を得るためには、FFT 実行後に規格化が必要である。

SSL II シリーズの他の 3 次元実フーリエ変換サブルーチンを使用する場合は、呼び出し時に必要な変数や作業領域等が異なるが、戻り値は (11.4) または (11.5) 式と等しい。個々のサブルーチンの用法については、「SSL II 使用手引書」を参照。

SSL II/VPP のサブルーチンを使用する時は、コンパイル時にオプション `-lssl2vpp` を指定する。

```
frt -Wx -lssl2vpp program_name.f
```

FFT のプログラムは以下の手順に分けられる。

1. 変数の宣言
2. 被変換関数 $F[j_1, j_2, j_3]$ を定義
3. 前処理: 作業用配列の初期化など

4. FFT の実行。DP_V3DRCF を呼び出す
5. 後処理: 結果の規格化、スペクトルの計算、データの出力など

以上の手順を VPP Fortran で記述したサンプル副プログラム (fft_FtoA.f) を本節の末尾に添付する。この副プログラムおよび次節のサンプル副プログラムは、付属の CD-ROM に含まれるメインプログラム (main_fft.f) を実行するときに呼び出されるように設定されている。このメインプログラムは、データファイル (init.data) から被変換配列 $F[j_1, j_2, j_3]$ を読み込んで、FFT を実行・出力する。メインプログラムの主要部分は以下の通りである。

```

      INTEGER NPE
      PARAMETER (NPE=2)
      !XOCL PROCESSOR PE(NPE)
      C
      !XOCL PARALLEL REGION                                ← 並列化の開始
      CCC READ INITIAL DATA FROM A FILE
      !XOCL SPREAD REGION /PE(1)
         FN1='init.data'
         OPEN(1,FILE=FN1,FORM='UNFORMATTED')
         READ(1) F_G                                       ← 初期配列 F をファイルから読み込む
         CLOSE(UNIT=1)
      !XOCL END SPREAD
      CCC FFT F[j1,j2,j3] -> A[l1,l2,l3]
         CALL FFT_FtoA                                     ← FFT を実行 (fft_FtoA.f の呼び出し)
      CCC OUTPUT F[j1,j2,j3] & A[l1,l2,l3]
         CALL OUT_FFT                                      ← FFT 結果をファイルに出力
      CCC CALCULATE & OUTPUT ENERGY SPECTRUM
         CALL SPECTRUM                                     ← スペクトルの計算・出力 (sec. 10.2)
      !XOCL END PARALLEL                                  ← 並列化の終了
      STOP
      END

```

サンプル副プログラム (fft_FtoA.f) はメインプログラム中の並列実行領域 (!XOCL PARALLEL REGION !XOCL END PARALLEL の間) から呼び出す。

このサンプル副プログラム (fft_FtoA.f) 中にも示した通り、DP_V3DRCF は

```
CALL DP_V3DRCF(X_G,KX,NX,N1,N2,N3,ISIN,ISN,WR_G,WI_G,KW,W,IW,ICON)
```

のようにして呼び出す。フーリエ変換 ([11.4] 式) のときは $ISN = 1$ 、逆フーリエ変換 ([11.5] 式) のときは $ISN = -1$ とする。被変換配列および変換結果は配列 $X_G[1:KX, 1:N2, 1:NX]$ に格納される。 KX および NX は配列 X_G の要素数を表す変数で、整数 $N1, N2, N3$ はそれぞれ x_1 -, x_2 -, x_3 - 方向 (プログラム中では、 i -, j -, k - 方向) の格子数である。整数 $ISIN = 1$ または -1 は変換の方向を表し、(11.4), (11.5) 式の r の値と一致させる。以降は作業領域 (WR_G, WI_G, W) と、その配列の要素数 (KW, IW)、および、コンディションコード ($ICON$) である。グローバル配列 X_G, WR_G, WI_G は全て、1 次元目を均等分割する。各引数の詳細に付いては表 11.1 または使用手引書を参照。

フーリエ変換が実行されると、配列 X_G の要素 $X_G[1:N1, 1:N2, 1:N3/2+1]$ には $A[0 : N_1 - 1, 0 : N_2 - 1, 0 : N_3/2]$ の実部が、 $X_G[1:N1, 1:N2, N3/2+2:2(N3/2+1)]$ には $A[0 : N_1 - 1, 0 : N_2 - 1, 0 : N_3/2]$ の虚部が格納される。残りのモードの振幅は (11.3) 式の複素共役関係から得られる。付属 CD-ROM のサンプルプログラムでは、配列 X_G のこれらの要素を配列 A に代入している。

サンプルメインプログラム (main_fft.f) を実行したときに読み込まれる初期配列 F と、そのフーリ

表 11.1: 3次元離散実フーリエ変換サブルーチン DP_V3DRCF の引数 (「富士通 SSL II/VPP 使用手引書」より編集)

引数	型	備考
X_G[KX, N2, NX]	(倍精度) 実数型グローバル配列	入力: 被変換配列、出力: 変換結果。1次元目を均等分割
KX	整数	配列 X_G の 1次元目の大きさ。N1 以上で 2*NPE の倍数 (NPE: 並列数) 最適値: $KX = 2*NPE*[(N1+2*NPE-1)/(2*NPE)]$
NX	整数	配列 X_G の 3次元目の大きさ。 $NX \geq 2*(N3/2 + 1)$
N1	整数	変換する実データの 1次元目の大きさ。 $N1 = 2^p 3^q 5^r$ (p, q, r は 0 または正整数)
N2	整数	変換する実データの 2次元目の大きさ $N1 = 2^p 3^q 5^r$ (p, q, r は 0 または正整数)
N3	整数	変換する実データの 3次元目の大きさ $N1 = 2^p 3^q 5^r$ (p, q, r は 0 または正整数)
ISIN	整変数	変換の方向。ISIN = -1 または 1。(11.4)、(11.5) 式の r と一致させる
ISN	整変数	1(変換)、-1(逆変換)
WR_G[KW, N2, KX]	倍精度実数型グローバル配列	作業領域。1次元目を均等分割
WI_G[KW, N2, KX]	倍精度実数型グローバル配列	作業領域。1次元目を均等分割
KW	整数	配列 WR_G と WI_G の 1次元目の大きさ。N3/2 + 1 以上で NPE の倍数 最適値: $KW = NPE*[(N3/2 + NPE)/NPE]$
W[IW]	倍精度実数型 1次元配列	作業領域
IW	整数	配列 W の大きさ。 $IW = 2*MAX(N1, N2, N3)$
ICON	整変数	コンディションコード

エ変換の結果 A をそれぞれある断面について示したのが図 11.1である。 $A(\mathbf{k})$ は 3次元の波数空間分布を持つため、このままではどのようなスケールのゆらぎが主な成分なのかが解りにくい。そこで、次節では $A(\mathbf{k})$ から 1次元的な波数空間でのスペクトル $P(k)dk$ をとる手順を解説する。

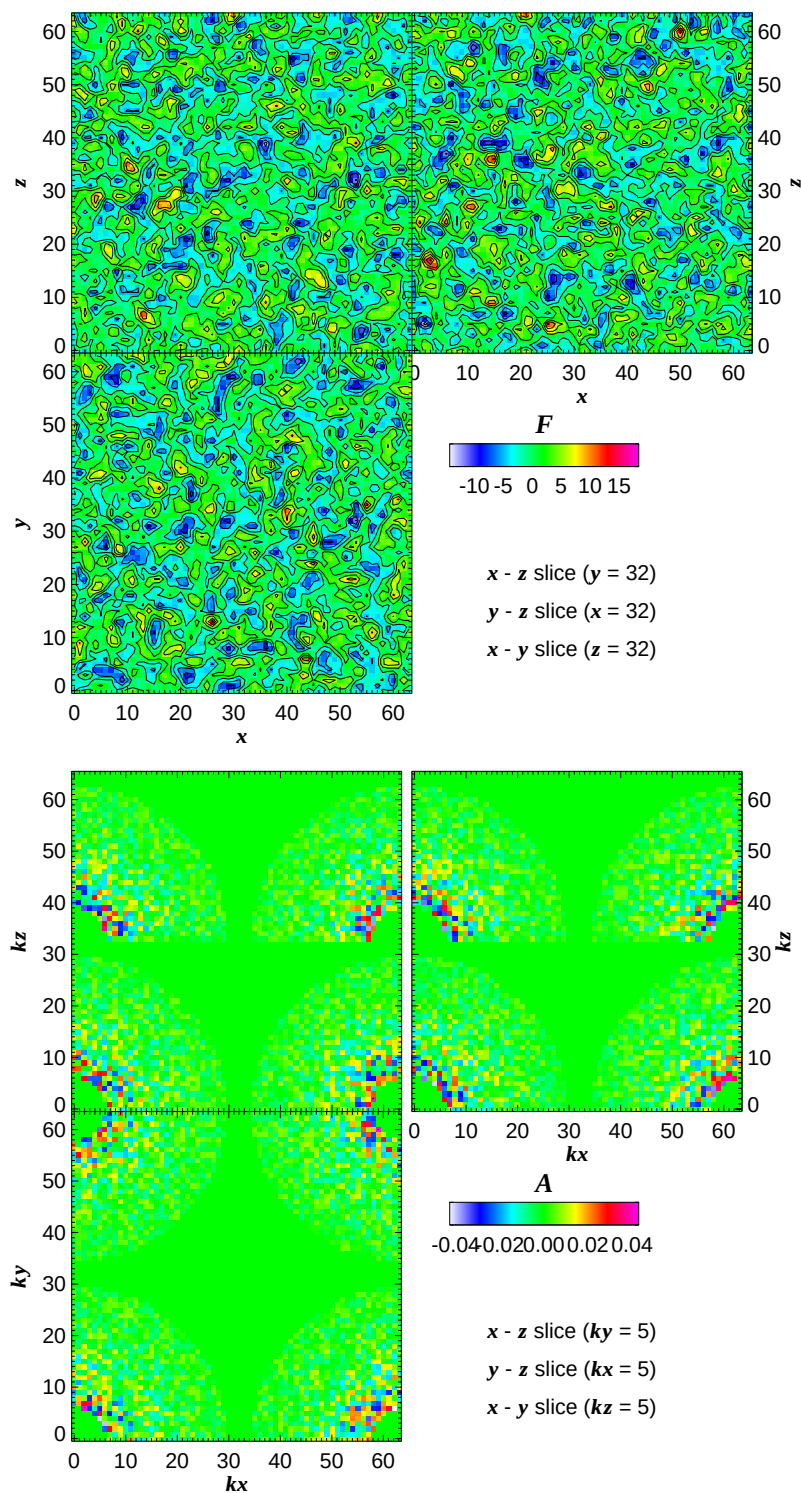


図 11.1: サンプルプログラム (main_fft.f) の被変換配列 $F[x, y, z]$ (上図) と FFT 結果 $A[kx, ky, kz]$ (下図) の断面図。

DP_V3DRCF による FFT のサンプルプログラム (fft_FtoA.f)

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C PROGRAM NAME: fft_FtoA.f
C WHAT CAN THIS DO?: Fast Fourier Transformation using DP_V3DRCF
C                      F[i,j,k] -> A[i,j,k].          ※このプログラムでは、非変換配列 F は既に
C                                                         与えられているとした。
C FILES TO OUTPUT: fft.data (binary)
C
C The original version by Kanako Sugimoto (2004/09/01)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE FFT_FtoA
CC --- DEFINE VARIABLES          ← (1) ここから変数、配列の宣言
      INTEGER NPE
      PARAMETER (NPE=2)
!XOCL PROCESSOR PE(NPE)
!XOCL SUBPROCESSOR PES(NPE)=PE(1:NPE)
      INTEGER nmax,N1,N2,N3
      PARAMETER (nmax=64,N1=nmax,N2=nmax,N3=nmax)
      INTEGER i,j,k
      REAL*8 normi
      REAL*8 F(N1,N2,N3),F_G(N1,N2,N3)
CC --- PARAMETERS FOR FFT
      INTEGER KX, NX, ISIN, ISN, KW, IW, ICON
      PARAMETER (IW=2*N1)
      PARAMETER (NX=2*(N3/2+1))
      PARAMETER (KX=(N1+2*NPE-1)/(2*NPE)*2*NPE)
      PARAMETER (KW=(N3/2+NPE)/NPE*NPE)
      REAL*8 A(N1,N2,NX),A_G(N1,N2,NX)
      REAL*8 X(KX,N2,NX),X_G(KX,N2,NX)
      REAL*8 W(IW), WR_G(KW,N2,KX), WI_G(KW,N2,KX)
      equivalence (F_G,F),(A_G,A),(X_G,X)
      common /func/ F_G, A_G
      common /ampx/ X_G
!XOCL INDEX PARTITION INF=(PES,INDEX=1:N3,PART=BAND)
!XOCL INDEX PARTITION INA=(PES,INDEX=1:N3+2,PART=BAND)
!XOCL INDEX PARTITION INFFT=(PES,INDEX=1:KX,PART=BAND)
!XOCL INDEX PARTITION INW=(PES,INDEX=1:KW,PART=BAND)
!XOCL LOCAL F(:, :, /INF(OVERLAP=(1,1))), A(:, :, /INA), X(/INFFT, :, :)
!XOCL GLOBAL F_G, A_G, X_G, WR_G(/INW, :, :), WI_G(/INW, :, :)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC PRE TRANSFORMATION PROCESSING
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCC INITIALIZE TEMPORAL ARRAYS          ← (3) 作業用配列の初期化 [推奨]
!XOCL SPREAD DO /INFFT
      DO i=1,KX
        DO k=1,NX
          DO j=1,N2
            X(i,j,k)=0.d0
          END DO
        END DO
      END DO
!XOCL END SPREAD DO
!XOCL SPREAD DO /INW
      DO i=1,KW
        DO k=1,KX
          DO j=1,N2
            WI_G(i,j,k)=0.d0
            WR_G(i,j,k)=0.d0
          END DO
        END DO
      END DO

```

```

        END DO
      END DO
    END DO
!XOCL END SPREAD DO
!XOCL SPREAD MOVE /INFFT
    DO i=1,N1
      DO k=1,N3
        DO j=1,N2
          X(i,j,k)=F_G(i,j,k)          ← f[i,j,k] を入出力配列X[i,j,k] に代入
        END DO
      END DO
    END DO
  END DO
!XOCL END SPREAD(ID)
!XOCL MOVE WAIT(ID)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC CALL FFT FUNCTION
CCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCC FOURIER TRANSFORMATION: F(i,j,k) -> A(i,j,k)
    ISN=1
CC DIRECTION FOR THE TRANSFORMATION:
CC    ISIN = 1(r_i=+1); 0(no-transformation); -1(r_i=-1)
    ISIN=1
C
    CALL                                ← (4) FFT サブルーチンを CALL
    & DP_V3DRCF(X_G,KX,NX,N1,N2,N3,ISIN,ISN,WR_G,WI_G,KW,W,IW,ICON)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC POST TRANSFORMATION PROCESSING
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCC NORMALIZE AMPLITUDE                ← (5) ここから後処理
    NORMI=1.0d0/DFLOAT(N1*N2*N3)
!XOCL SPREAD DO /INFFT
    DO i=1,N1
      DO k=1,NX
        DO j=1,N2
          X(i,j,k)=X(i,j,k)*NORMI      ← 振幅の規格化
        END DO
      END DO
    END DO
!XOCL END SPREAD
C
!XOCL SPREAD MOVE /INA
    DO k=1,NX
      DO j=1,N2
        DO i=1,N1
          A(i,j,k)=X_G(i,j,k)          ← X[i,j,k] を A[i,j,k] に代入
        END DO
      END DO
    END DO
!XOCL END SPREAD MOVE(ID)
!XOCL MOVE WAIT(ID)
RETURN
END

```

11.2 スペクトルの計算法

モード毎の振幅配列 $A(\mathbf{k})$ からスペクトル、

$$P(k) dk = \sum_{|\mathbf{k}|=k}^{k+\Delta k} A^2(\mathbf{k}), \quad (11.6)$$

を計算する手順は、

1. 変数の宣言と初期化: スペクトルを取る波数範囲と刻み幅などを設定する
2. スペクトルの波数目盛り $k(i)$ を定義する
3. 各モードの3次元波数の絶対値をとる: $\mathbf{k}[\ell_1, \ell_2, \ell_3] \rightarrow k[\ell_1, \ell_2, \ell_3] = |\mathbf{k}[\ell_1, \ell_2, \ell_3]|$
4. 各々の波数目盛り (ビン) ごとにモードの振幅 $A(k)$ を足し合わせる ([11.6] 式)
5. 結果の出力

である。

サンプル副プログラムを節末に添付してある (spectrum.f)。このサンプル副プログラムを基に、前節の図 11.1 の分布からスペクトルを計算した結果が図 11.2 である。サンプルプログラムで与えられた初期配列 F は、波数範囲 $8 \leq k \leq 30$ に、 $P(k)d \log k = \text{const}$ なるスペクトルをもつゆらぎ ($1/f$ ゆらぎ) であったことが解る。

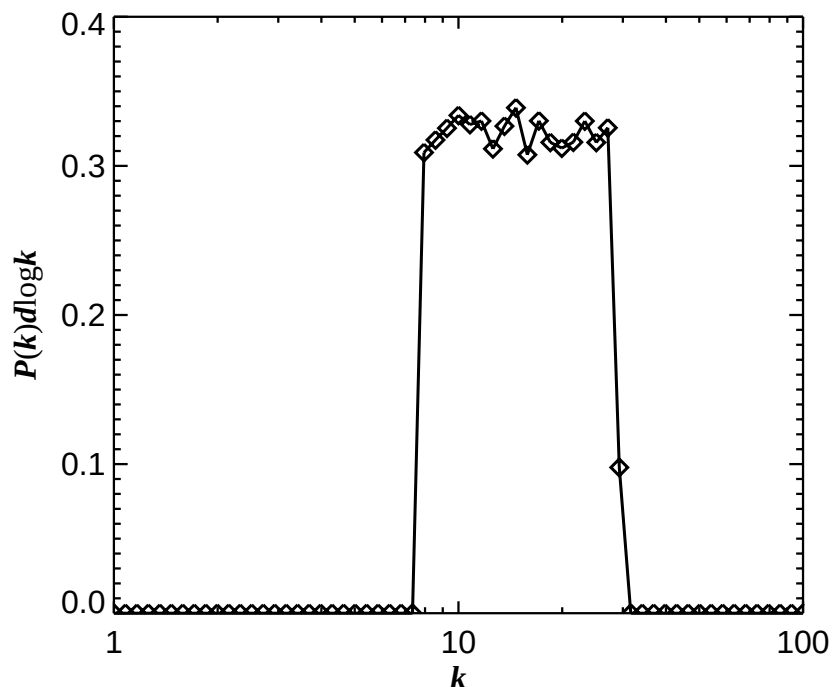


図 11.2: サンプルプログラムのペクトル。図 11.1の分布を元に計算した。

スペクトル計算のサンプルプログラム (spectrum.f)

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C PROGRAM NAME: spectrum.f
C WHAT CAN THIS DO?: CALCULATE SPECTRUM, A[i,j,k] -> A[1]^2.
C FILE TO OUTPUT: spcec.data (binary)
C
C The original version by Kanako Sugimoto (2004/09/02)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE SPECTRUM
      INTEGER NPE
      PARAMETER (NPE=2)
!XOCL PROCESSOR PE(NPE)
!XOCL SUBPROCESSOR PES(NPE)=PE(1:NPE)
      INTEGER nmax,N1,N2,N3,KX,NX
      PARAMETER (nmax=64,N1=nmax,N2=nmax,N3=nmax)
      PARAMETER (NX=2*(N3/2+1))
      PARAMETER (KX=(N1+2*NPE-1)/(2*NPE)*2*NPE)
      INTEGER i,j,k,normi
      INTEGER wnx(N1),wny(N2),wnz(NX)
      REAL*8 X(N1,N2,NX),X_G(N1,N2,NX)
      equivalence (X_G,X)
      common /nwave/ wnx,wny,wnz
      common /ampx/ X_G
!XOCL INDEX PARTITION INFFT=(PES,INDEX=1:KX,PART=BAND)
!XOCL LOCAL X(/INFFT,,:,:)
!XOCL GLOBAL X_G
CCC VARIABLES FOR SPECTRUM
      REAL*8 A2(KX,N2,NX)

```

← (1) ここから変数、配列の宣言

```

!XOCL LOCAL A2(/INFFT, :, :)
      INTEGER iskip,nbin,noffset,ntick,minp,maxp
      REAL*8 tickmin,tickmax,wnum
      PARAMETER (minp=0,maxp=2)
      PARAMETER (ntick=30,tickmin=1.0D1**minp,tickmax=1.0D1**maxp)
      PARAMETER (nbin=(maxp-minp)*ntick+1)
      PARAMETER (noffset=-minp*ntick+1)
      INTEGER L(kx,n2,n3),L_G(kx,n2,n3),nmode(nbin)
      DOUBLE PRECISION Pow(nbin),kwave(nbin)
      EQUIVALENCE (L_G,L)
      common /spec/ Pow,nmode,kwave
!XOCL GLOBAL L_G
!XOCL LOCAL L(/infft, :, :)
      CHARACTER*40 FN3
!XOCL SPREAD DO /INFFT
      DO I=1,KX
        DO K=1,NX
          DO J=1,N2
            A2(I,J,K)=0.0D0
          END DO
        END DO
      END DO
!XOCL END SPREAD DO
      ISKIP=N3/2+1
!XOCL SPREAD DO /INFFT
      DO I=1,N1
        DO K=1,N3/2+1
          DO J=1,N2
            A2(I,J,K)=X(I,J,K)**2+X(I,J,K+ISKIP)**2
          END DO
        END DO
      END DO
!XOCL END SPREAD DO
CCC INITIALIZE SPECTRUM ARRAY & DEFINE THE WAVENUMBER OF EACH BIN
      DO i=1,nbin
        nmode(i) = 0
        Pow(i) = 0.0D0
        kwave(i) = tickmin*1.0D1**(DFLOAT(i-1)/ntick)
      END DO
CC DEFINE TICK LEVEL OF THE DENSITY FOR EACH MESH
!XOCL SPREAD DO /INFFT
      DO i=1,N1
        DO k=1,N3/2+1
          DO j=1,N2
            WNUM=DSQRT(DFLOAT(wnx(i)**2+wny(j)**2+wnz(k)**2))
            L(i,j,k) =
&              IDINT(DLOG10(max(dfloating(wnum),1.D-6))*dfloating(ntick)
&              +dfloating(noffset))
            L(i,j,k) = max(L(i,j,k),1)
            L(i,j,k) = min(L(i,j,k),nbin)
          END DO
        END DO
      END DO
!XOCL END SPREAD DO
CC CALCULATE SPECTRUM
!XOCL SPREAD DO /INFFT
      DO i=1,N1
        DO k=1,N3/2+1

```

← (1) 'ここから配列の初期化

← (2) スペクトルの波数目盛り(ビン)を定義

← (3) 各モードの波数がスペクトルの何番目のビンに相当するか

```

      DO j=1,N2
        Pow(L(i,j,k)) = Pow(L(i,j,k))+A2(i,j,k)    ← (4) ビン毎に振幅の2乗を合計
      END DO
    END DO
  END DO
!XOCL END SPREAD SUM(Pow)
!XOCL SPREAD DO /INFFT
  DO i=1,N1
    DO k=1,N3/2+1
      DO j=1,N2
        nmode(L(i,j,k)) = nmode(L(i,j,k))+1    ← (4)' ビン毎のモード数の合計
      END DO
    END DO
  END DO
!XOCL END SPREAD SUM(nmode)
CCC WRITE BINARY IN SINGLE PRECISION    ← (5) データ出力
  FN3='spec.data'
!XOCL SPREAD REGION /PES(1)
  OPEN(3,FILE=FN3,FORM='UNFORMATTED')
  write(3) nbin
  write(3) (real(kwave(i)),i=1,nbin)
  write(3) (real(Pow(i)),i=1,nbin)
  write(3) (nmode(i),i=1,nbin)
  CLOSE(UNIT=3)
!XOCL END SPREAD
  RETURN
END

```

11.3 おまけ：乱流の発生法、スペクトルの制御、非圧縮性ゆらぎの生成

§11.1に述べた通り、実 FFT サブルーチン DP_V3DRCF の呼び出し変数 $ISN = -1$ とすると、波数空間の複素振幅配列 A から実スカラー配列 F への逆フーリエ変換が出来る。(実は、サンプルプログラムの初期配列 F もこの手法で作成した。) 本節では、フーリエ逆変換によって、ランダムなスカラー場やベクトル場 (乱流場) を発生させるにあたって、筆者が必要だと感じた注意点とテクニックを述べる。

11.3.1 乱流の発生法 (逆実 FFT の注意点)

数値計算で密度揺らぎなどのランダムなスカラー場を発生させるためには、配列 $A[N_1, N_2, 2(N_3/2+1)]$ にランダムな振幅 (乱数) を与えて、逆フーリエ変換によって実空間の分布 $F[N_1, N_2, N_3]$ を計算すればよい。乱流速度場などのランダムなベクトル場は FFT サブルーチンを複数回呼んでベクトルの各方向成分を得る。このとき、 A と F の配列要素の数が異なることに注意が必要である。 $N_1 \times N_2 \times N_3$ 点の実スカラー分布 F を得るためには、振幅要素 $A[I, J, K]$ を $N_1 \times N_2 \times 2(N_3/2+1)$ 個用意する。FFT の変換結果と同様に、配列 A の要素 $A[1:N_1, 1:N_2, 1:N_3/2+1]$ には複素振幅の実部を、要素 $A[1:N_1, 1:N_2, N_3/2+2:2(N_3/2+1)]$ には複素振幅の虚部を与える。

また、変換後の配列が実数配列になるためには、複素振幅分布 A が (11.3) 式の複素共役関係を満たすように配列 A を与えなくてはならない²。SSL II による FFT サブルーチンでは、この複素共役関係を利用して配列 A の半分を実部、残りを虚部としているので、ほとんどの要素では A に任意の値を与えても複素共役関係を破らない。問題となるのは $k_3 = 1, N_3/2+1, N_3/2+2, 2(N_3/2+1)$ の面で、

$$\begin{aligned} A[I, J, 1] &= A[N_1+1-I, N_2+1-J, 1], \\ A[I, J, N_3/2+1] &= A[N_1+1-I, N_2+1-J, N_3/2+1], \\ A[I, J, N_3/2+2] &= -A[N_1+1-I, N_2+1-J, N_3/2+2], \\ A[I, J, 2(N_3/2+1)] &= -A[N_1+1-I, N_2+1-J, 2(N_3/2+1)], \end{aligned}$$

ここで、 $I, J = 1, 2, 3, \dots, N_1/2+1$ の要素である。

11.3.2 任意のパワースペクトルを持つゆらぎの与え方

一般に、振幅配列 A に定数や一様乱数を与えても、スペクトルは平坦にならない (例外は、1次元のスペクトル $P[k_1]dk_1$)。これは、図 11.3に示したように波数によってスペクトルのビン Δk あたりに含まれるモードの数が異なるためである。

図 11.3は、 $0 \leq k_1, k_2 \leq 4$ について波数幅 $\Delta k = 1.7$ でスペクトルを取ったときに、各ビンに含まれるモードを異なるマークで示している。波数域が $0 \leq k < 1.7$ のビンには4つのモードが、波数域が $1.7 \leq k < 3.4$ のビンには9つのモードが含まれることが解る。このように、波数 k が大きいほど $k \leq |k| < k + \Delta k$ の領域が波数空間に占める面積 (3D では体積) が増えるため、その内部に含まれるモードの数が大きくなる。また、数値計算では格子数が有限なので、扱える波数に上限がある。このため、波数の上限付近では境界の影響によるモード数の減少が起こる (図 11.3の 3, 4 番目の

²ただし、この複素共役関係を破るような A を与えても、数値計算上はエラーメッセージが出ない可能性がある。しかし、複素共役関係を満たさない配列 A を与えると、実行結果の実スカラー配列 F に不連続面が現われるなどの症状が出ることを筆者は確認している。また、これらの複素共役成分に与えた振幅はそれぞれフーリエ変換されて加算されるので注意が必要である。

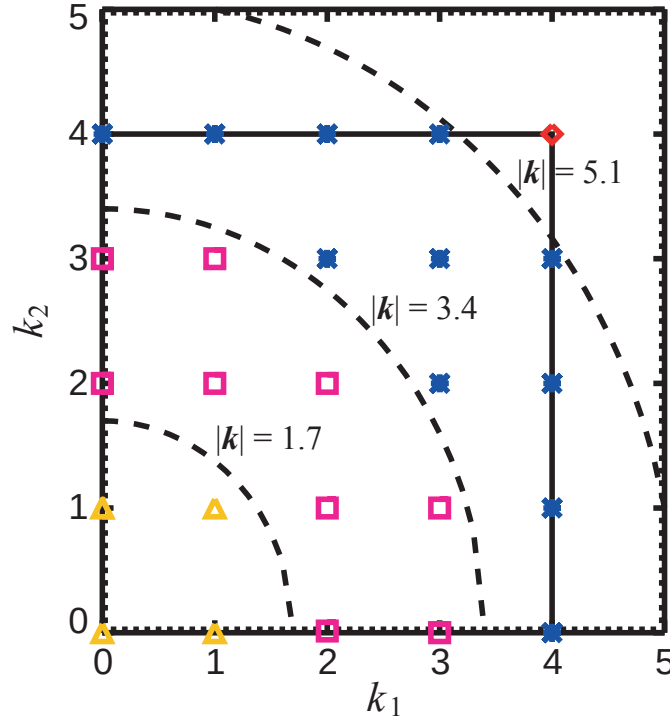


図 11.3: 波数空間分布とモードの数。波数空間 $0 \leq k_1, k_2 \leq 4$ において、 $\Delta k = 1.7$ でスペクトルを取るときの、各ビンに含まれるモードを異なる記号で記した。

ビン)。このようにビン毎のモード数が異なるとスペクトル、

$$P(k) dk = \sum_{|\mathbf{k}|=k}^{k+\Delta k} A^2(\mathbf{k}) = \langle A^2 \rangle n_{\text{mode}}[k \leq |\mathbf{k}| < k + \Delta k], \quad (11.7)$$

はモード数に比例した形になる。ここで、 n_{mode} は $k \leq |\mathbf{k}| < k + \Delta k$ に含まれるモードの数、 $\langle A^2 \rangle$ は一様乱数 (または定数) 配列 A の自乗平均である。

ビン毎のモード数の変化によるスペクトルの波数依存を回避するもっとも一般的な解決法は、波数が連続的に分布している極限での $k \leq |\mathbf{k}| < k + \Delta k$ に含まれるモードの数、即ち、波数空間の面積 (3D では体積) で振幅 A を予め割ることである。2次元波数空間では、

$$n_{\text{mode}} \propto 2\pi k dk \quad (11.8)$$

なので、(11.7) 式から、 $P(k) \propto k^p$ とするためには $A(\mathbf{k})$ を $k^{(p-1)/2}$ 倍すればよい。同様に、3次元の場合、

$$n_{\text{mode}} \propto 4\pi k^2 dk \quad (11.9)$$

なので、 $P(k) \propto k^p$ とするためには $A(\mathbf{k})$ を $k^{(p-2)/2}$ 倍すればよい。

また、サンプルプログラムのスペクトル (図 11.2) のように、 $\log k$ スケールで平坦な $1/f$ ゆらぎ ($kP[k] = \text{const}$) は $\Delta \log k$ 幅のモードの数を考えて、 $A(\mathbf{k})$ を $k^{-3/2}$ 倍すればよい (3次元の場合)。

11.3.3 非圧縮性ゆらぎの与え方

ランダムなベクトル場 $\mathbf{v}$ を初期条件として数値計算するときに、非圧縮性のゆらぎを与えたいことがある。非圧縮性のゆらぎとは、

$$\nabla \cdot \mathbf{v} = 0, \quad (11.10)$$

をみたすベクトル場のことである。圧縮性のゆらぎと非圧縮性のゆらぎを分けて考えるのは、これらが物理的に異なる性質を持つからである。また、磁場ゆらぎ $\delta \mathbf{B}$ を与えるときは、 $\nabla \cdot \delta \mathbf{B} = 0$ をみたすゆらぎを与える必要がある。

非圧縮性ゆらぎ $\mathbf{v}_s$ を与えるためには、ゆらぎのベクトルポテンシャル $\mathbf{V}$ を与えて、

$$\mathbf{v}_s = \nabla \times \mathbf{V}, \quad (11.11)$$

から速度ゆらぎを計算すればよい。これにより $\mathbf{v}_s$ は

$$\nabla \cdot \mathbf{v}_s = \nabla \cdot (\nabla \times \mathbf{V}) = 0, \quad (11.12)$$

をみたす。

プログラム手順は、

1. 振幅配列 $\mathbf{A}$ を逆フーリエ変換して、ランダムなベクトル場を生成する $\rightarrow \mathbf{F}$ 。3次元ベクトル場の場合はこの手順を3回繰り返す: $[\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3] \rightarrow [\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3]$ ($\mathbf{A}^*$ と $\mathbf{F}^*$ はそれぞれ $N_1 \times N_2 \times 2(N_3/2 + 1)$ と $N_1 \times N_2 \times N_3$ 要素の配列) 生成したベクトル場 $[\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3]$ をベクトルポテンシャル $\mathbf{V}$ とする, $\mathbf{V} = [\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3]$

2. $\mathbf{v}_s = \nabla \times \mathbf{V}$ によってランダムな非圧縮性ベクトル場を得る

ここで注意しなくてはならないのは、初期の $[\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3]$ が一様な乱数配列であっても、 $\mathbf{v}_s$ のフーリエ成分は k に比例する波数依存を持つてしまうことである。即ち、ベクトルポテンシャル

$$\mathbf{V} = \sum_{k_1} \sum_{k_2} \sum_{k_3} \mathbf{A} \exp[-i\mathbf{k} \cdot \mathbf{x}] \quad (11.13)$$

として、 $\mathbf{A}$ を一様乱数ベクトルとすると、 $\nabla \times \mathbf{V}$ によって与えられる非圧縮性乱流場は、

$$\mathbf{v}_s = -i \sum_{k_1} \sum_{k_2} \sum_{k_3} \mathbf{k} \times \mathbf{A} \exp[-i\mathbf{k} \cdot \mathbf{x}] \quad (11.14)$$

となり、振幅は典型的に $k = |\mathbf{k}|$ に比例する。これを防ぐためには、初期の振幅 $\mathbf{A}$ を各々の波数 k で割ってから逆FFTをして $\mathbf{V}$ を得ればよい。

同様の理屈で、逆フーリエ変換によってスカラーポテンシャル ϕ を与えて $\mathbf{v} = \nabla \phi$ とすれば、 $\nabla \times \mathbf{v} = 0$ (渦なし) のベクトル場が得られる。

11.4 富士通 SSL II の離散型実フーリエ変換サブルーチン

「富士通 SSL II 使用手引書」より抜粋、編集

表 11.2: SSL II (FORTRAN、逐次計算、ベクトル計算機用)

名前	機能	備考
RFT	1 次元、実 FFT (2 基底)	配列の要素数 $N_1 = 2^p$ (p は正整数)

表 11.3: SSL II/VP (FORTRAN、ベクトル機用)

名前	機能	備考
VRFT1	1 次元、実 FFT (性能優先型、2 基底)	配列の要素数 $N_1 = 2^p$ (p は 0 または正整数)
VRFT2	1 次元、実 FFT (メモリ節約型、2 基底)	配列の要素数 $N_1 = 2^p$ (p は 0 または正整数)
VMRF2	1 次元、多重、多次元実 FFT (混合基底)	1 次元目の要素数 N_1 は偶数。その他の要素数は任意
VMRFT	多次元、多重実 FFT (2, 3 及び 5 の混合基底)	要素数 $N(1 : M)$ を持つ M 次元のデータを変換。 $N(1 : M-1)$ のうち少なくとも 1 つは偶数かつ $N_i = 2^p 3^q 5^r$ ($i = 1, 2, \dots, M; p, q, r$ は 0 又は正整数)
VRPF3	3 次元、素因子実 FFT	1 次元目の要素数 N_1 は偶数、かつ、 N_i ($i = 1, 2, 3$) は $\{2, 3, 4, 5, 7, 8, 9, 16\}$ のうち互いに素な数の積で表せる
VSRFT	1 次元、多重実 FFT (2, 3 及び 5 の混合基底)	要素数 N_1 を持つ、 M 本のデータを変換。 N_1 または M の一方は偶数、かつ、 $N_1 = 2^p 3^q 5^r$ (p, q, r は 0 または正整数)

表 11.4: SSL II/VPP (VPP FORTRAN 用)

名前	機能	備考
DP_V1DRFT	1 次元、実 FFT (2, 3 及び 5 の混合基底)	要素数 N_1 の 1 次元配列を $N_1 = m \times n$ の 2 次元配列にして変換。 n は偶数、かつ、 $m, n/2 = 2^p 3^q 5^r$ (p, q, r は 0 または正整数)
DP_V1DRCF	1 次元、実 FFT (2, 3 及び 5 の混合基底)	要素数 N_1 の 1 次元配列を $N_1 = m \times n$ の 2 次元配列にして変換。 $N_1 = 2^p 3^q 5^r$ (p, q, r は 0 または正整数)
DP_V2DRCF	2 次元、実 FFT (2, 3 及び 5 の混合基底)	各次元方向の要素数 $N_i = 2^p 3^q 5^r$ ($i = 1, 2; p, q, r$ は 0 または正整数)
DP_V3DRCF	3 次元、実 FFT (2, 3 及び 5 の混合基底)	本文参照。各次元方向の要素数 $N_i = 2^p 3^q 5^r$ ($i = 1, 2, 3; p, q, r$ は 0 または正整数)

表 11.5: SSL II/HPF (HPF, VPP 用)

名前	機能	備考
DH_V1DRFT	1 次元、実 FFT (2, 3 及び 5 の混合基底)	要素数 N_1 の 1 次元配列を $N_1 = m \times n$ の 2 次元配列にして変換。 n は偶数、かつ、 $m, n/2 = 2^p 3^q 5^r$ (p, q, r は 0 または正整数)
DH_V1DRCF	1 次元、実 FFT (2, 3 及び 5 の混合基底)	要素数 N_1 の 1 次元配列を $N_1 = m \times n$ の 2 次元配列にして変換。 $N_1 = 2^p 3^q 5^r$ (p, q, r は 0 または正整数)
DH_V2DRCF	2 次元、実 FFT (2, 3 及び 5 の混合基底)	各次元方向の要素数 $N_i = 2^p 3^q 5^r$ ($i = 1, 2; p, q, r$ は 0 または正整数)
DH_V3DRCF	3 次元、実 FFT (2, 3 及び 5 の混合基底)	各次元方向の要素数 $N_i = 2^p 3^q 5^r$ ($i = 1, 2, 3; p, q, r$ は 0 または正整数)

表 11.6: C-SSL II (C 言語、逐次計算、ベクトル計算機用)

名前	機能	備考
c_dvmrft	多次元、多重実 FFT (2, 3 及び 5 の混合基底)	要素数 $N(1 : M)$ を持つ M 次元のデータを変換。 $N(1 : M-1)$ のうち少なくとも 1 つは偶数かつ $N_i = 2^p 3^q 5^r$ ($i = 1, 2, \dots, M; p, q, r$ は 0 又は正整数)
c_dvsrft	1 次元、多重実 FFT (2, 3 及び 5 の混合基底)	要素数 N_1 を持つ、 M 本のデータを変換。 N_1 または M の一方は偶数、かつ、 $N_1 = 2^p 3^q 5^r$ (p, q, r は 0 または正整数)
c_dvrft1	1 次元、実 FFT (2 基底)	配列の要素数 $N_1 = 2^p$ (p は 0 または正整数)
c_dvrft2	1 次元、実 FFT (メモリ節約型、2 基底)	配列の要素数 $N_1 = 2^p$ (p は 0 または正整数)
c_dvrpf3	3 次元、素因子実フーリエ変換	1 次元目の要素数 N_1 は偶数、かつ、 N_i ($i = 1, 2, 3$) は $\{2, 3, 4, 5, 7, 8, 9, 16\}$ のうち互いに素な数の積で表せる

参考文献

- [1] Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P., 1992, Numerical Recipes in C [日本語版] (技術評論社), 379 (12 章: フーリエ変換)
- [2] 今村 勤, 「物理とフーリエ変換」 (岩波書店)
- [3] 小形 正男, 裳華房テキストシリーズ - 物理学「振動・波動」 (裳華房), 133 (7 章: 波束とフーリエ変換)
- [4] 富士通株式会社, 2001, 「FUJITSU C-SSL II 使用手引書」, 第 4 版 (富士通株式会社)
- [5] 富士通株式会社, 1987, 「FUJITSU SSL II 使用手引書」, 初版 (富士通株式会社)
- [6] 富士通株式会社, 1991, 「FUJITSU SSL II 拡張機能使用手引書」, 第 2 版 (富士通株式会社)
- [7] 富士通株式会社, 2001, 「FUJITSU SSL II 拡張機能使用手引書 II」, 第 5 版 (富士通株式会社)
- [8] 富士通株式会社, 2001, 「FUJITSU SSL II/VPP 使用手引書」, 第 2 版 (富士通株式会社)
- [9] 富士通株式会社, 2000, 「FUJITSU SSL II/VPP 使用手引書」, 初版 (富士通株式会社)