附章2 “NILS モデル”の漸化式より分配関数を求める！

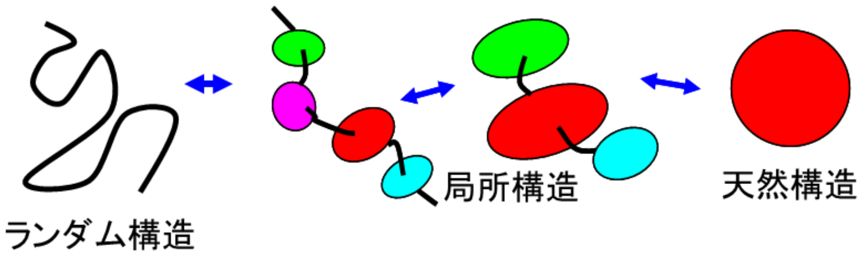
ーFortran プログラムと解説ー

＜要旨＞
タンパク質分子のフォールディング過程の統計力学モデルである“NILS モデル”のキーポイントは次の点にあると強調している：『フォールディング過程において，“天然接触相互作用”が局所構造内のみで働くと仮定する．』
この仮定を採用すると系の分配関数を求めるための漸化式が導かれることを示している．その漸化式を，2 つの変数，λ（天然状態にある残基数を数えるための変数）と，ξ（局所構造に固有の量を加算していくための変数）を用いて変形している．変形した漸化式を実行するために，コンピュータのメモリーに作業用領域を確保し，その領域を利用して，補助的な分配関数を順次記憶していく手順を示している．
漸化式を実行して，最終的な 2 次元の表から分配関数が求められ，その分配関数から，天然状態にあるアミノ酸残基数が η であるときの自由エネルギー$F(\eta)$を求める式を導いている．この自由エネルギー関数 $F(\eta)$ と分配関数 Z より Φ の理論値が求まり，フォールディング過程の“遷移状態”での構造形成に関して議論できると述べている．
章末に，漸化式より分配関数を求める手順を Fortran 言語でプログラミングしたものを収録している．同時に，プログラムの簡単な解説も行っている．

“NILS モデル”の仮定と系の分配関数

タンパク質分子のフォールディング過程の統計力学モデルである“NILS モデル”の描像は次の通りである：

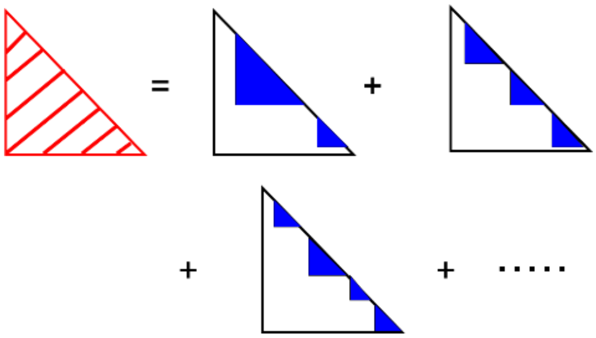
『タンパク質分子のフォールディング過程は，ランダム・コイル構造から，まず近距離相互作用によって小さな局所構造が構築され、順次，より中距離相互作用が働いて，これらの局所構造が成長し，更に，長距離の相互作用が働いて局所構造が融合され，より大きな局所構造となる．最終的に，短距離相互作用や長距離相互作用などの分子内相互作用の間に互いに矛盾が無く整合的である天然構造が形成される．』（下図参照）



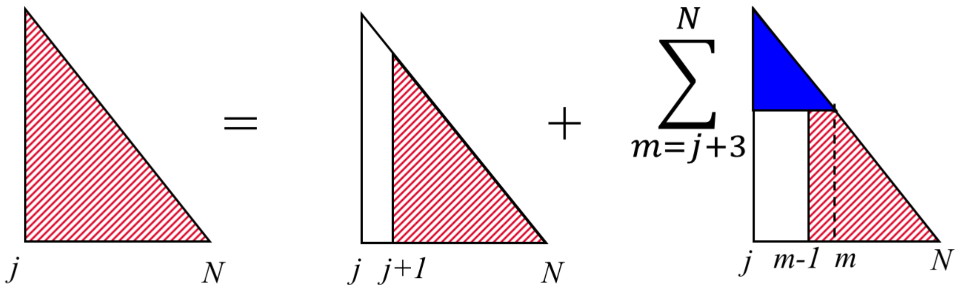
この“NILS モデル”による描像のキーポイントは，フォールディング過程において，“天然接触相互作用”は局所構造内だけで働き，次のような場合の相互作用はすべて無視する点である（“天然非接触相互作用”も無視する）：

(a) 局所構造間の相互作用（図 a），
(b) 局所構造とランダム・コイル構造間の相互作用（図 b）
(c) ランダム・コイル間の相互作用

従って，“NILS モデル”による分配関数は，ポリペプチド鎖に沿って形成される，あらゆる可能な局所構造の組み合わせを考慮すれば求められる（下図参照）.



特に，3 次元格子タンパク質に対して特化すると，“NILS モデル”による分配関数を求めるための漸化式は次のイラストで表示できる：



局所構造内のエネルギー値を$E(j,m)$とすると，漸化式は次式のように書くことができる：

$$Z_{j,N} = Z_{j+1,N} + \sum_{m=j+3}^{N} f(j,m)^{-1} \exp\{-\beta E(j,m)\} Z_{m-1,N} = \sum_{m=j+2}^{N} w_{j,m} Z_{m-1,N}$$

$$(\ \mathrm{j} = N-3, N-4, \ldots, 3,2,1\ )$$

ただし，　$Z_{N-1,N} = Z_{N-2,N} = 1$ とおく．

この漸化式では，3 次元格子タンパク質における最小の局所構造は 4 個のアミノ酸残基（ユニットとみなす）からなる構造であることを考慮している．また，アミノ酸残基がランダム・コイル状態にあるときを基準状態とみなす（つまり，ランダム・コイル状態での統計重率を 1 とみなす）場合の漸化式である．

$f(j,m)$ は，j 番目の残基からm 番目の残基までのポリペプチド鎖のセグメントが局所構造を形成したときのエントロピー損失を表し，$f(j,m) = 1.4048 \times (4.750)^{m-j-2}$ とおく（**『タンパク質分子のフォールディング統計力学』，第 4 章，**p.46 **参照**）．

ここで，次のような 2 つの変数 λ, ξ を導入する：

λ：天然状態にある残基数を数えるための変数（最終的に$\lambda = 1$ とおく）

（ λ のべき乗は，局所構造における天然状態にある残基数，η に対応）

ξ ： 局所構造$[j,m]$に固有の量（$h(j,m)$：整数）を加算していくための変数

これらの 2 つの変数 λ, ξ を用いると，統計重率 $w_{j,m}$ は次のように書ける：

$$w_{j,m} = f(j,m)^{-1} \exp\{-\beta E(j,m)\} = e^{-\{E(j,m)+T\ln f(j,m)\}/T}$$

$$= \lambda^{m-j-2}\, \xi^{h(j,m)}$$

ただし，　$\xi = e^{-1/T}$，$h(j,m) = -\{E(j,m) + T\ln f(j,m)\} \times 100$

無次元量：$\frac{kT}{\epsilon_0} = T^*$ より，T^* を改めて T と置き換えている．

最終的に漸化式は，2 つの変数 λ,ξ を用いると次のように書ける：

$$Z_{j,N} = \sum_{m=j+2}^{N} \lambda^{m-j-2}\, \xi^{h(j,m)}\, Z_{m-1,N}$$

“NILS モデル”の漸化式より分配関数を求める手順

コンピュータのメモリーに作業用領域を確保し，その作業領域を利用して，漸化式に従って，変数 λ ,ξ のそれぞれの“べき乗”で，順次，補助的な分配関数を記憶していく．その手順は次の通りである：

① まず、 $j = N-2$ のとき（ $Z_{N-2,N}$ ）の処理を行う（“作業メモリー”に出力する)：

$$Z_{N-2,N} = \lambda^0\, \xi^0$$

② 次に、$j = N-3$ のとき（ $Z_{N-3,N}$ ）の処理を行う（“作業メモリー”に出力する)：

$$Z_{N-3,N} = \lambda^0\, \xi^0 + \lambda\, \xi^{h(N-3,N)}$$

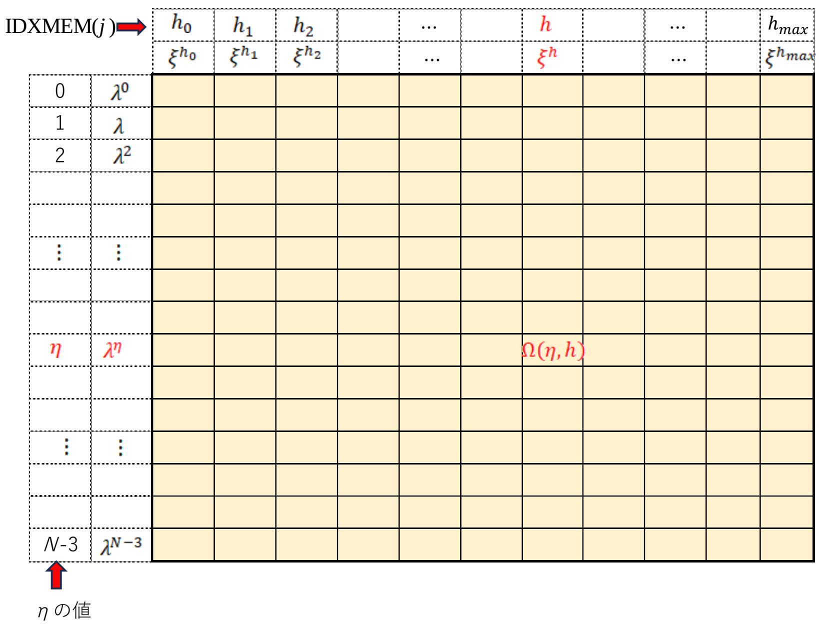
③ ループ ($j = N-4, N-5, \cdots, 3, 2, 1$) を実行するために漸化式を次のように書き換える：

$$Z_{j,N} = \lambda^{N-j-2} \xi^{h(j,N)} + \sum_{jj=j+2}^{N-2} \lambda^{jj-j-1}\, \xi^{h(j,jj+1)}\, Z_{jj,N} + \lambda^0 \xi^0 Z_{j+1,N}$$

④ 最終的に， λ ,ξ の“べき乗”の 2 次元の表が生成される．

分配関数より天然状態にあるアミノ酸残基数が η であるときの自由エネルギー $F(\eta)$ を求める

漸化式を実行すると，右図のような 2 次元の表が生成される：



表の行の λ のべき乗は，天然状態にある残基数，η と次のような対応関係にある;

$$\lambda^0, \lambda^1, \lambda^2, \cdots, \lambda^{N-3} \leftrightarrow \eta = 0,1,2, \cdots, N-3$$

表の列の IDXMEM(j)は，j 列目に格納される h の値（整数値）を表している．

最終的な分配関数 ($Z \equiv Z_{1,N}$) は，2 次元の表から次式のように求まる：

$$Z = \sum_{\eta} \sum_{h} \Omega(\eta, h) \lambda^{\eta} \xi^{h}$$

ただし，$\Omega(\eta, h)$ は，与えられた η, h を満たす全てのコンフォメーション数である．

分配関数は次式のようにも書ける：

$$Z = \sum_{\eta} W(\eta) \lambda^{\eta}$$

ここで，$W(\eta)$ は，天然状態にある残基数が η である全ての状態の統計重率の和であり，次式となる：

$$W(\eta) = \sum_{h} \Omega(\eta, h) \xi^{h}$$

天然状態にある残基数が η であるときの自由エネルギーは次式で与えられる：

$$F(\eta) = -k\,T \ln W(\eta)$$

ここで，$\eta = 0,1,2, \cdots, N-3$

この自由エネルギー関数 $F(\eta)$ と分配関数 Z より，フォールディング・レート k_f ，及び，アンフォールディング・レート k_u が計算でき，最終的に，Φ の理論値を求めることができる． Φ 値が求められると，フォールディング過程の“遷移状態”での各アミノ酸残基の構造形成に関して議論できる（『タンパク質分子のフォールディング統計力学』，第 7 章参照）．

次に，漸化式より分配関数を求める手順を Fortran 言語でプログラミングしたものを収録している．偶数ページのプログラムの内容を，奇数ページにイラスト付きで解説している．

NILS モデルの漸化式より分配関数を求めるプログラム（N0.1）

```
C   ----  パラメターの設定と COMMON 文 ----
      SUBROUTINE NILSCA
         PARAMETER (NDPW=150,NEMINW=-200,NEMAXW=3000)
         REAL*8   FW,FJ
C ----- COMMON 文 ---------------------------------------------
         COMMON /ESU/ NESUMT(NDPW,NDPW)
         COMMON /PF2/ FJ(NDPW,NEMAXW)
         COMMON /NBL/ NENIN,NEMINDM,NEMAX
         COMMON /PF1/ FW(NDPW,NEMAXW)
         COMMON /NDL/ NDP,NLR
         COMMON /IED/ IDXEND,IDXENG(NEMINW:NEMAXW),IDXMEM(NEMAXW)
         DATA   NRESD, NDW /1, 27/
C ----- IDXENG(I) の初期化  ---------------------------
              DO   9    I= -200, 3000
           9    IDXENG(I)= -1
             NEMINDM=2000
C -----  作業領域の先頭にセット------------------------
             REWIND NDW
C -----  変数：IDXEND   と N2 の初期化     -----------------------
             IDXEND =0
             N2=NDP-2
C =======<分配関数：Z_{NDP-2,NDP}   を求める> ===============
             IDXE=IDXEG(0)
              NJA= 1
              FW( NJA,IDXE )= 1.0
C   ----  分配関数：Z_{NDP-2,NDP}  の結果を作業領域に出力する   -------------
               WRITE( NDW )      N2, NJA, IDXEND, FW(1,IDXE)
C =====<分配関数：Z_{NDP-3,NDP} を求める> ===============
              N3= NDP-3
              NEN3=NESUMT(N3,NDP)
              IDXE=IDXEG(NEN3)
              NJA= 2
C   ----     Z_{NDP-3,NDP}  の初期化：全ての要素に 0.0 を代入--------------
              DO   10   IJ=1, IDXEND
              DO   10    I=1, NJA
  10           FW( I, IJ )= 0.0
C -------分配関数：Z_{NDP-3,NDP}  の要素の決定--------------
               FW(1,1 )= 1.0
               FW(NJA, IDXE )= 1.0
C -- 分配関数：Z_{NDP-3,NDP}  の結果を作業領域に出力する---------
          WRITE(NDW) N3,NJA,IDXEND, ((FW(I,IJ),I=1,NJA),IJ=1,IDXEND )
C================= No.2 へ続く=====================================
```

プログラムの解説（その 1）

○ ----- メインプログラムにおいて次の準備をしておく-------------------

- 機番 18 → 最終的に求まる分配関数 $FW(i,k)$ の出力（FORMAT 形式で）
- 機番 29 → 最終的に求まる分配関数 $FW(i,k)$ の出力 （UNFORMAT 形式で）
- 機番 27 (=NDW) → 作業領域の確保

  例）OPEN(NDW, FILE='ファイル名', FORM= 'UNFORMATTED')
- $NESUMT(i,k)$ ($= h(i,k)$) ：局所構造(i,k) に固有の量

  $NESUMT(i,k)$の最小値 →NENIN, 最大値 →NEMAX を求めておく

○ ------ 分配関数の定義 -- :

$FW(i,k)$：最終的な分配関数（途中の分配関数を作業領域へ出力する）

$FJ(i,k)$ ：仮想的な分配関数（作業領域からデータを入力する）

<コメント> プログラムの解説のため，次のような具体的な数値を仮定する：$NDP = 36$, $h(33,36) = 4,\ h(32,35) = 6,\ h(32,36) = 7,\ h(31,34) = 8,\ h(31,35) = 6,\ h(31,36) = 13$

◎ $j = NDP - 2$ (=34) のとき

$$Z_{N-2,N} = \lambda^0\,\xi^0 \quad \rightarrow Z_{34,,36} = \lambda^0\,\xi^0$$

○ IDXE=IDXEG(0)→文関数：FUNCTION IDXEG(IE)において,IDXE=IDXEG(0)=-1 となり，文番号 20 へ→DXEND=IDXEND+1→IDXEND=1→ IDXENG(0)=1, IDXE=1

		-2	-1	0	1	2	3	4	5	6	7	8	9	10
IDXENG(i)	→	・・・	-1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	・・・

→IDXMEM(1) (=IE) = 0

		1	2	3	4	5	6	7	8	9	10
IDXMEM(i)	→	0									・・・

○ 作業領域に$Z_{34,36}$の結果を出力：WRITE(NDW) N2, NJA, IDXEND, FW(1,IDXE)

作業領域→		34	1	1	1.0						・・・

◎ $j = NDP - 3$ (= 33) のとき

$$Z_{N-3,N} = \lambda^0\,\xi^0 + \lambda\,\xi^{h(N-3,N)} \rightarrow Z_{33,36} = \lambda^0\,\xi^0 + \lambda\,\xi^{h(33,36)}$$

$$\rightarrow Z_{33,36} = \lambda^0\,\xi^0 + \lambda\,\xi^4$$

		-2	-1	0	1	2	3	4	5	6	7	8	9	10
IDXENG(i)	→	・・・	-1	1	-1	-1	-1	2	-1	-1	-1	-1	-1	・・・

		1	2	3	4	5	6	7	8	9	10
IDXMEM(i)	→	0	4								・・・

○ 作業領域 に $Z_{N-3,N}$ (=$Z_{33,36}$)の結果を出力する：

WRITE(NDW) N3,NJA,IDXEND, ((FW(I,IJ),I=1,NJA),IJ=1,IDXEND)

作業領域→		34	1	1	1.0	33	2	2	1.0	0.0	0.0	1.0	・・・

$Z_{33,36} = \lambda^0\,\xi^0 + \lambda\,\xi^4 \quad \rightarrow$

			1	2
IDXMEM(j) →			(0)	(4)
FW(i,j)=	1	λ^0	1.0	0.0
	2	λ	0.0	1.0

NILS モデルの漸化式より分配関数を求めるプログラム（N0.2，続き）

```
              J= NDP-4
C ======    J = NDP-4  ～  J=1 のループ  ================================
          15       NJA= NDP-J-1
C   ---- FW(I,IJ)の初期化：全ての要素に 0.0 を代入-----------
             DO    20    IJ=1, IDXEND
             DO    20     I=1, NJA
                20         FW( I,IJ )= 0.0
C
C <漸化式の第 1 項：λ^{N-j-2}ξ^{h(j,N)}  の処理>
               NEJN = NESUMT(J,NDP)
               IDXE=IDXEG(NEJN)
               FW( NJA, IDXE )=1.0
C
C <漸化式の第 2 項：Σ_{jj=j+2}^{N-2} λ^{jj-j-1} ξ^{h(j,jj+1)} Z_{jj,N} > JJ=NDP-1～  JJ=J+3  のループ  ----

               JJ=NDP-1
               REWIND NDW
       25      CONTINUE
C   ----    作業領域の分配関数: Z_{JJ-1,NDP} を入力-------

               READ(NDW) JW,NJANOW,IDXNOW,((FJ(I,IJ),I=1,NJANOW),IJ=1,IDXNOW )
                            IF( JW.NE.JJ-1) STOP    '166 JW.NE.JJ-1'
               NEJJJ= NESUMT(J,JJ)
               NJA2= JJ-J-2
C
C     ====== DO 305  のループ  =========
         DO     305     IJ=1, IDXNOW
                     NEID = NEJJJ+IDXMEM(IJ)
                     IDXE= IDXEG(NEID)
C        ==== DO 30  のループ  ====
               DO     30     I=1, NJANOW
                  FW(NJA2 +I,IDXE )= FW(NJA2 +I,IDXE ) +FJ(I,IJ)
             30        CONTINUE
       305    CONTINUE
C
C ------- JJ=NDP-1～  JJ=J+3  のループの終了-------

             IF( JJ.LE.J+3) GO    TO    35
             JJ=JJ-1
       GO    TO    25
C================== No.3  へ続く=======================================
```

プログラムの解説（その 2，続き）

◎ J＝NDP-4～J=1 に対する漸化式より，分配関数 $Z_{NDP-4,NDP}$ ～$Z_{1,NDP}$ を順次求めていく

$$Z_{j,N} = \lambda^{N-j-2}\xi^{h(j,N)} + \sum_{jj=j+2}^{N-2} \lambda^{jj-j-1}\,\xi^{h(j,jj+1)}\,Z_{jj,N} + \lambda^0\xi^0 Z_{j+1,N}$$

◎ $j = NDP - 4\ (= 32\ )$ のとき，分配関数 $Z_{32,36}$ を求める

15　NJA＝NDP-J-1=36-32-1=3

○ 漸化式の第 1 項：$\lambda^2\xi^{h(32,36)}$ の処理→NESUM(32,36)=7，IDXENG(7)=3, IDX=3 より，

		-2	-1	0	1	2	3	4	5	6	7	8	9	10
IDXENG(i)	→	・・・	-1	1	-1	-1	-1	2	-1	-1	3	-1	-1	・・・

IDXMEM(3) (=IE) =7 →

		1	2	3	4	5	6	7	8	9	10
IDXMEM(i)	→	0	4	7							・・・

FW(NJA, IDX)= 1.0 →

$\lambda^2\xi^{h(32,36)} = \lambda^2\xi^7$ →

IDXMEM(j) →		1 (0)	2 (4)	3 (7)
FW(i,j)= 1	λ^0	0.0	0.0	0.0
2	λ	0.0	0.0	0.0
3	λ^2	0.0	0.0	1.0

○ 漸化式の第 2 項：（jj-j-1=1, h(32,35)=6, IDXE=4 より）

$$\sum_{jj=j+2}^{N-2} \lambda^{jj-j-1}\,\xi^{h(j,jj+1)}\,Z_{jj,N} \rightarrow \lambda\xi^{h(32,35)}Z_{34,36} = \lambda\xi^6 Z_{34,36} = \lambda\xi^6$$

IDXENG(6)=4 →

		-2	-1	0	1	2	3	4	5	6	7	8	9	10
IDXENG(i)	→	・・・	-1	1	-1	-1	-1	2	-1	4	3	-1	-1	・・・

IDXMEM(4) =6 →

		1	2	3	4	5	6	7	8	9	10
IDXMEM(i)	→	0	4	7	6						・・・

FW(2,4)= FW(2,4) +FJ(1,1) →

IDXMEM(j) →		1 (0)	2 (4)	3 (7)	4 (6)
FW(i,j)= 1	λ^0	0.0	0.0	0.0	0.0
2	λ	0.0	0.0	0.0	1.0
3	λ^2	0.0	0.0	1.0	0.0

○ 漸化式の第 3 項：

$$\lambda^0\xi^0 Z_{j+1,N} \rightarrow \lambda^0\xi^0 Z_{33,36} = \lambda^0\xi^0(\lambda^0\,\xi^0 + \lambda\,\xi^4) = \lambda^0\,\xi^0 + \lambda\,\xi^4$$

○ ------- 分配関数 $Z_{32,36}$ の結果 -------------------------

$$Z_{32,36} = \lambda^2\xi^{h(32,36)} + \lambda\xi^{h(32,35)}Z_{34,36} + \lambda^0\xi^0 Z_{33,36}$$

$$= \lambda^2\xi^7 + \lambda\xi^6(\lambda^0\,\xi^0) + \lambda^0\xi^0(\lambda^0\,\xi^0 + \lambda\,\xi^4)$$

$$\therefore Z_{32,36} = \lambda^2\xi^7 + \lambda(\xi^6 + \xi^4) + \lambda^0\xi^0 \rightarrow$$

IDXMEM(j) →		1 (0)	2 (4)	3 (7)	4 (6)
FW(i,j)= 1	λ^0	1.0	0.0	0.0	0.0
2	λ	0.0	1.0	0.0	1.0
3	λ^2	0.0	0.0	1.0	0.0

		-2	-1	0	1	2	3	4	5	6	7	8	9	10
IDXENG(i)	→	・・・	-1	1	-1	-1	-1	2	-1	4	3	-1	-1	・・・

		1	2	3	4	5	6	7	8	9	10
IDXMEM(i)	→	0	4	7	6						・・・

NILS モデルの漸化式より分配関数を求めるプログラム（N0.3，続き）

```
C  ----漸化式の第 3 項：λ⁰ξ⁰Z_{j+1,N}  の処理-------
C  ----  作業領域の分配関数: Z_{J+1,NDP}を入力-------
     35 READ(NDW) JW,NJANOW,IDXNOW,((FJ(I,IJ),I=1,NJANOW),IJ=1,IDXNOW  )
C  ---  JW=J+1 をチェック → 誤データのとき STOP する→IF( JW.NE.J+1) STOP ----
        IF( JW.NE.J+1)    STOP 998
C  ----  入力した FJ(I,IJ) のデータを，そのまま FW(I,IJ)の要素に入力------------------
        DO   40   IJ= 1, IDXNOW
        DO   40    I=1, NJANOW
        40      FW(I,IJ )= FW(I,IJ) + FJ(I,IJ)
C ---- J=1 まで実行したかをチェック：OK→終了の手続き(45)へ,
        IF(   J.LE. NRESD ) GO   TO   45
C -- Z_{J,NDP}の結果を作業領域へ出力→ J=J-1 として 15 へ戻る-------------------------
       WRITE(NDW)   J,NJA, IDXEND,((FW(I,IJ),I=1,NJA),IJ=1,IDXEND )
          J= J-1
          GO   TO   15
C ----- J=1 まで実行したとき終了の手続き(45)へ（終了していないとき→STOP）-------
        45   IF( J.NE.NRESD) STOP 997
C -------  最終の分配関数の結果を出力（ UNFORMATTED 形式で）----------------------
      WRITE( 29 )   NJA,IDXEND,NEMAX
      WRITE( 29 ) (IDXMEM( I ), I= 1,IDXEND )
      WRITE( 29 ) (IDXENG(I), I=NEMINDM,NEMAX )
      WRITE( 29 ) ((FW(I,IJ), IJ=1,IDXEND ),I=1,NJA)
C -------最終の分配関数の結果を出力（ FORMATTED 形式で）-------------------------
          WRITE( 18,655) NJA,IDXEND
       655   FORMAT(' NJA,IDXEND=',3I5,I10)
          WRITE(18,662)   ( IDXENG(I), I=NEMINDM,NEMAX )
       662   FORMAT('   IDXENG='/ (30I5))
          WRITE(18,661)(IDXMEM( I ), I= 1,IDXEND )
       661   FORMAT('   IDXMEM='/ (30I5) )
         DO 1994 I=1,NJA
           WRITE(18,1862) ( FW(I,IJ), IJ=1,IDXEND )
      1994     CONTINUE
      1862 FORMAT(20E12.4)
C -----  終了の手続きをする  -------
          ENDFILE   NDW
          RETURN
          END
C===== No.4  →文関数：  FUNCTION IDXEG(IE)   へ続く==========
```

プログラムの解説（その 3，続き）

◎ $j = NDP - 5\ (= 31)$ のとき，分配関数 $Z_{31,36}$ を求める

15　NJA = NDP-J-1=36-31-1=4

漸化式：

$$Z_{j,N} = \lambda^{N-j-2}\xi^{h(j,N)} + \{ \sum_{jj=j+2}^{N-2} \lambda^{jj-j-1}\, \xi^{h(j,jj+1)}\, Z_{jj,N} \} + \lambda^0\xi^0 Z_{j+1,N}$$

$$Z_{31,36} = \lambda^3\xi^{h(31,36)} + \{\textstyle\sum_{jj=33}^{34} \lambda^{jj-32}\, \xi^{h(31,jj+1)}\, Z_{jj,N} \} + \lambda^0\xi^0 Z_{32,36}$$

$$= \lambda^3\xi^{h(31,36)} + \{ \lambda^2\xi^{h(31,35)} Z_{34,36} + \lambda\xi^{h(31,34)} Z_{33,36} \} + \lambda^0\xi^0 Z_{32,36}$$

$$= \lambda^3\xi^{13} + \{ \lambda^2\xi^6\lambda^0\xi^0 + \lambda\xi^8(\ \lambda^0\ \xi^0 + \lambda\ \xi^4) \} + \lambda^0\xi^0\{ \lambda^2\xi^7 + \lambda(\xi^6 + \xi^4) + \lambda^0\xi^0 \}$$

$$= \lambda^3\xi^{13} + \{ \lambda^2\xi^6 + \lambda\xi^8 + \lambda^2\ \xi^{8+4} \} + \{ \lambda^2\xi^7 + \lambda(\xi^6 + \xi^4) + \lambda^0\xi^0 \}$$

$$= \lambda^3\xi^{13} + \lambda^2\{ \xi^6 + \xi^{12} + \xi^7 \} + \lambda\{ \xi^8 + \xi^6 + \xi^4 \} + \lambda^0\xi^0$$

,

			1	2	3	4	5	6	7
IDXMEM(j) →			(0)	(4)	(7)	(6)	(13)	(8)	(12)
FW(i,j)=	1	λ^0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
	2	λ	0.0	1.0	0.0	1.0	0.0	1.0	0.0
	3	λ^2	0.0	0.0	1.0	1.0	0.0	0.0	1.0
	4	λ^3	0.0	0.0	0.0	0.0	1.0	0.0	0.0

		-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
IDXENG(i)	→	· · ·	-1	1	-1	-1	-1	2	-1	4	3	6	-1	-1	-1	7	5	· · ·

		1	2	3	4	5	6	7	8	9	10
IDXMEM(i)	→	0	4	7	6	13	8	12			· · ·

“Red House Beside the Lake”

NILS モデルの漸化式より分配関数を求めるプログラム（N0.4, 続き）

```
      FUNCTION IDXEG(IE)
C - パラメターの設定 -----------------
      PARAMETER (NDPW=150,NEMINW=-200,NEMAXW=3000)
      REAL*8    FW
      COMMON /PF1/ FW(NDPW,NEMAXW)
      COMMON /NDL/ NDP,NLR
      COMMON /IED/ IDXEND,IDXENG(NEMINW:NEMAXW),IDXMEM(NEMAXW)
C
          IF(IE.LT.NEMINW) STOP 'IE_FUNCTION IDXEG'
        IDXE= IDXENG( IE )
        IF(IDXE) 20 , 10 , 10
  10    IDXEG= IDXE
        RETURN
C
  20    IDXEND= IDXEND + 1
        IF(IDXEND.GT.NEMAXW) STOP 'IDXEND_FUNCTION IDXEG'
        IDXENG( IE )= IDXEND
        IDXMEM(IDXEND)= IE
        IDXEG= IDXEND
C -----FW(I,IDXEND)の初期化（0.0 を代入する）
        DO   50   I= 1, NDP
  50        FW(I, IDXEND )= 0.0
         RETURN
         END
```

“Red House in Winter”