

# アミロースのコンホメーション解析用 コンピュータプログラム

岡本高洋・中田 靖・林 武・恒川行康・北村進一・久下 喬

TAKAHIRO OKAMOTO, YASUSHI NAKATA, TAKESHI HAYASHI,  
YUKIYASU TSUNEKAWA, SHINICHI KITAMURA and TAKASHI KUGE

Computer programs for the conformational analysis of amylose

**要旨:** アミロース鎖のコンホメーション解析のためのコンピュータプログラムをフォートラン言語で作成した。全体は、6個の主プログラムから構成されている。その第Ⅰのプログラムは、 $\alpha$ -D-maltose の原子座標を maltose 関連化合物の中性子線あるいは X 線解析データから求めるためのものである。第Ⅱのプログラムは、半経験的なエネルギー関数を用いて  $\alpha$ -D-maltose のコンホメーションエネルギーを求めるためのものである。このプログラムは、エネルギーを等高線地図として表示する機能及び、各コンホメーションの存在確立を求める機能を有している。第Ⅲのプログラムは、モンテカルロ法によって、各コンホメーションの存在確立に従ってアミロース鎖を生成し、さらに生成アミロースのコンホメーションの特性を表わす各種パラメータを算出するためのものである。出力は、分子鎖の末端間距離の分布を表わすヒストグラム及び、特性比等である。代表的なモンテカルロ鎖のグラフィックディスプレイへの表示は、他のプログラム(第Ⅴのプログラム)を用いてなされる。第Ⅳのプログラムは、高分子統計力学を基礎とした解析的方法によって、分子鎖のコンホメーションの平均を表わすパラメータを算出するものである。出力は、特性比、平均持続ベクトルの座標及び、持続長等である。最後(第Ⅵ)のプログラムは、アミロース鎖の排除体積効果を計算するためのものである。効率的に摂動鎖を生成するために、Wall と Erpenbeck の方法<sup>1)</sup>を用いた。互いのプログラムは、オンラインで連結でき、データの再入力が必要がない様に設計されている。上記のプログラム群は、他の多糖のコンホメーションの理論的解析に対しても応用できるものである。

## はじめに

水溶液中におけるアミロース溶液は、会合、沈殿化(老化)、ヨーソあるいはブタノールなどの有機化合物との包接、ゲル化などの興味深い挙動を示す。このため、多くの研究者が溶液中アミロースのコンホメーションに関心を寄せてきたが、その溶液物性の分子論的な解明、特にその局所コンホメーションは、十分に解明されていない。

コンピュータの大型高速化に伴い、最近、高分子統計力学的な方法<sup>2)</sup>を用い、溶液中アミロースのコンホメーションを理論的に予測する試みがなされている<sup>3-5)</sup>。理論的研究は、実験で得られた結果をより定量的に解釈したり、実験では得られていない性質を予測するためのものであり、両者は互いに相補的な関係にある。その意味において、理論的研究は、今後のアミロース研究に必要不可欠なものである。

本報告は、当研究室で開発されたアミロースのコン

ホメーションの理論的解析のためのコンピュータプログラムの概略的説明、使用法、及び主なプログラムのリストである。本プログラムは、アミロースのみならず他の多糖のコンホメーションの理論的研究にも広く利用される可能性を持つものである。

## I $\alpha$ -D-Maltose (結合角 $\tau=180^\circ$ ) の座標作成プログラム

### I-1 YASUSHI. FORT (AMALT1)

使用言語: FORTRAN (自由型式)

#### I-1-1 概要

本プログラムは、 $\beta$ -D-maltose monohydrate の中性子線回折<sup>7)</sup>あるいは、methyl  $\beta$ -maltoside monohydrate<sup>8)</sup>、cyclohexaamylose-potassium acetate complex<sup>9)</sup>のX線回折のデータから、 $\alpha$ -maltose ( $\tau=180^\circ$ )の座標を算出することを目的として作成された。たとえば文献7)の回折データは、単斜晶系 (monoclinic) である単位格子 (cell unit) の座標分率 (fractional atomic coordinate) として与えられているので、これより直接  $\beta$ -maltose の非還元残基の原子座標を直交座標系で求めることができる。座標分率から原子座標への変換は、単位格子長を  $a, b, c$  として  $a$  軸,  $c$  軸のつくる角度を  $\beta$  とすると次の行列  $F$  で表現される。

$$F = \begin{bmatrix} a & 0 & c \cos \beta \\ 0 & b & 0 \\ 0 & 0 & c \sin \beta \end{bmatrix} \quad (1)$$

そして直交座標系で得られた非還元残基の原子座標

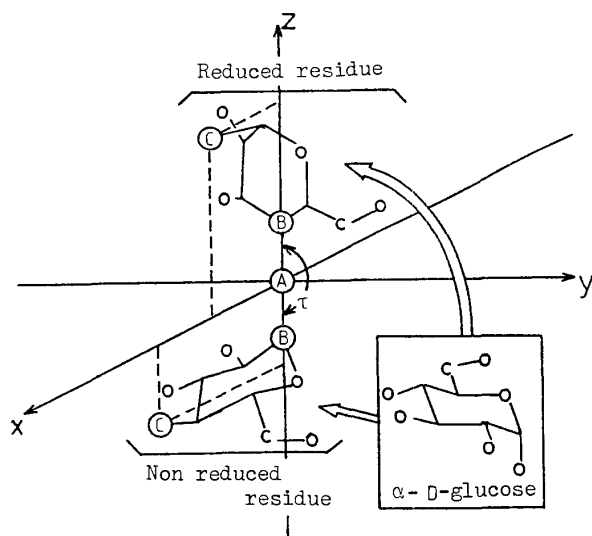


Fig. 1. Construction of  $\alpha$ -D-maltose on a right-handed cartesian coordinate system from geometric data of  $\alpha$ -D-glucose.

を用い、Fig. 1 に示すようなグリコシド結合角  $\tau$  が  $180^\circ$  の  $\alpha$ -maltose の原子座標を求める。さらにその原子座標を直交座標系から極座標系に変換する。

#### I-1-2 フローチャート

計算過程のフローチャートを Fig. 2 に示す。以下フローチャートに付した番号に従ってその内容を説明する。

1.  $\beta$ -Maltose の非還元残基の座標分率データを読み込む。
2. 読み込みデータをプリンタに出力する。
3. 座標分率を直交座標系の原子座標に変換する。
4. 3で得られた原子座標を座標変換し、 $\alpha$ -maltose ( $\tau=180^\circ$ ) の非還元残基の座標データを作成する。
5. 3で得られた原子座標を座標変換し、 $\alpha$ -maltose ( $\tau=180^\circ$ ) の還元残基の座標データを作成する。
6. 4で得られた非還元残基の座標データをプリンタ

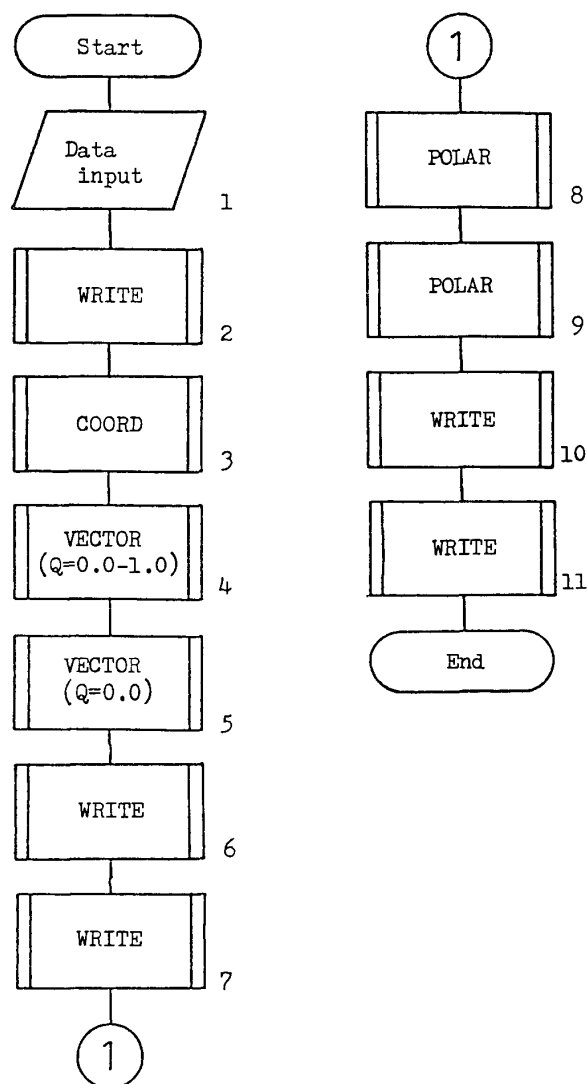


Fig. 2. Flow chart for YASUSHI. FORT (AMALT1) program.

に出力する。

7. 5 で得られた還元残基の座標データをプリンタに出力する。
8. 非還元残基の原子座標を直交座標系から極座標系に変換する。
9. 還元残基の原子座標を極座標系に変換する。
10. 8 で得られた非還元残基の原子座標をプリンタに出力する。
11. 9 で得られた還元残基の原子座標をプリンタに出力する。

なお、計算はすべて倍精度で行なった。また、得られた  $\alpha$ -maltose ( $\tau=180^\circ$ ) の極座標データは、後述のポテンシャルエネルギー計算プログラムで使用する必要があるため、AMALT1 の出力型式をファイルに格納するように変更した YASUSHI. FORT (AMALT2) も用意した。

### I-1-3 サブルーチン

#### (1) WRITE

機能

座標データを glucose 単位でプリンタに出力する。データは原子の種類別 (C, O, H) に整理して出力される。

呼び出し型式

CALL WRITE (X, Y, Z)

引数の説明

X, Y, Z : glucose の座標あるいは座標分率。実数型一次元配列。入力。

#### (2) COORD

機能

座標分率を行列  $F$  を使って直交座標に変換する。

呼び出し型式

CALL COORD (X, Y, Z, a, b, c,  $\beta$ )

引数の説明

X, Y, Z : glucose の座標分率 (入力時), 直交座標 (出力時)。実数型一次元配列。入出力。

a, b, c : 単位格子長。実数型。入力。

$\beta$  : a 軸, c 軸のつくる角度。実数型。入力。

#### (3) VECTOR

機能

Glucose の原子座標から,  $\alpha$ -maltose ( $\tau=180^\circ$ ) の非還元残基あるいは還元残基の原子座標を, 直交座標系で算出する。

呼び出し型式

CALL VECTOR (X, Y, Z, Q, L, M, N, X', Y', Z')

引数の説明

X, Y, Z : Glucose の原子座標。実数型一次元配列。入力。

Q : 本サブルーチンは,  $Q=0.0-1.0$  の時非還元残基の座標を,  $Q=1.0$  の時還元残基の座標を計算する。実数型。入力。

L : 原点に来る原子①の配列要素の添字の値。

M : x 軸上に来る原子②の配列要素の添字の値。

N : xz 平面に来る原子③の配列要素の添字の値 (Fig. 1 参照)。L, M, N はともに整数型。入力。

X', Y', Z' :  $\alpha$ -Maltose の glucose 残基の原子座標。実数型一次元配列。出力。

#### (3) POLAR

機能

直交座標を極座標に変換する。

呼び出し型式

CALL POLAR (X, Y, Z, R, T, P)

引数の説明

X, Y, Z : Glucose 残基原子の直交座標。実数型一次元配列。入力。

R, T, P : Glucose 残基原子の極座標 ( $r, \theta, \phi$ )。実数型一次元配列。出力。

### I-2 DLVT. FORT77

使用言語 : FORTRAN (自由型式)

#### I-2-1 概要

本プログラムは, YASUSHI. FORT (AMALT1) の計算結果の検定を目的とする。原子間の結合距離 (bond length), 結合角 (valence angle), 二面角あるいはねじれ角 (dihedral angle or torsion angle) を, AMALT1 の計算結果である原子の座標データより計算する。(表記法は Fig. 3 を参照)

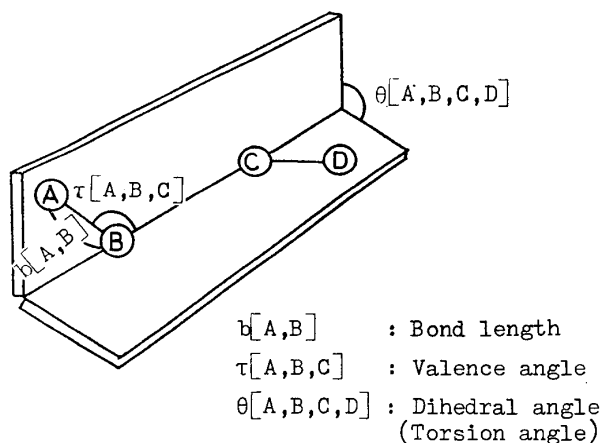


Fig. 3. An illustrative definitions of bond length, valence angle, and dihedral angle.

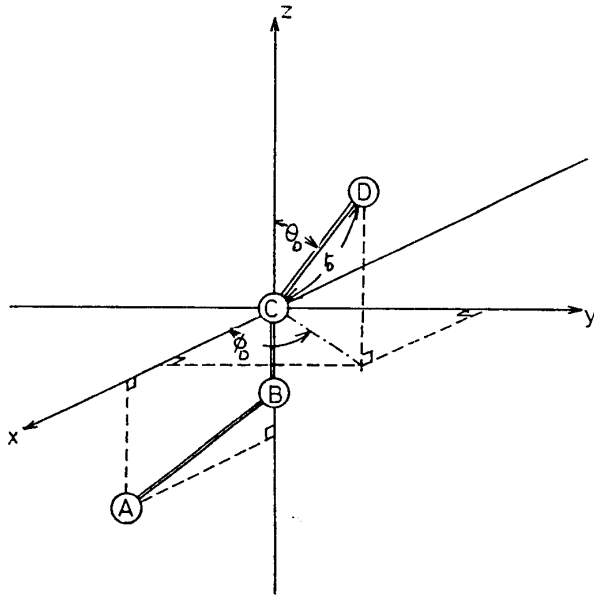


Fig. 4. Relation between  $\phi_D$  and  $\theta[A, B, C, D]$ .

I-2-2 フローチャート

結合角及び結合距離の計算は、常法により行なった。二面角の計算は、AMALT1 における手法を利用し、次に示す計算方法に従い行なった。二面角を  $\theta[A, B, C, D]$  とすると、原子 A, B, C, D を Fig. 4 に示す位置に来るように座標変換を行なう。この時、原子 D の極座標 ( $r_D, \theta_D, \phi_D$ ) を求めると、二面角  $\theta[A, B, C, D]$  は、

$$\phi_D = \theta[A, B, C, D] \quad (2)$$

となる。

次に、Fig. 5 に示したフローチャートに従って計算過程を説明する。

1. i に原子数を読み込む。
2.  $x_i, y_i, z_i$  に原子の座標を読み込む。
- 3~5.  $i=25$  の時、glucose の座標データとみなし、原子別に整理して出力する。 $i \neq 25$  の時は、そのまま出力する。
- 6, 7. 計算型 GO TO 文により、読み込んだ j (= 1~4) の値に従って目的の計算エリアへジャンプする。
- 8, 9. 二面角  $\theta[A, B, C, D]$  の各原子の座標 (x, y, z) の配列要素の添字 a, b, c, d を読み込み、9 で二面角  $\theta[A, B, C, D]$  を計算する。
- 10, 11. 結合角  $\tau[A, B, C]$  を計算する。
- 12, 13. 結合距離  $b[A, B]$  を計算する。

計算は、すべて倍精度で行なった。Fig. 6 に、データの書式と出力例を示す。

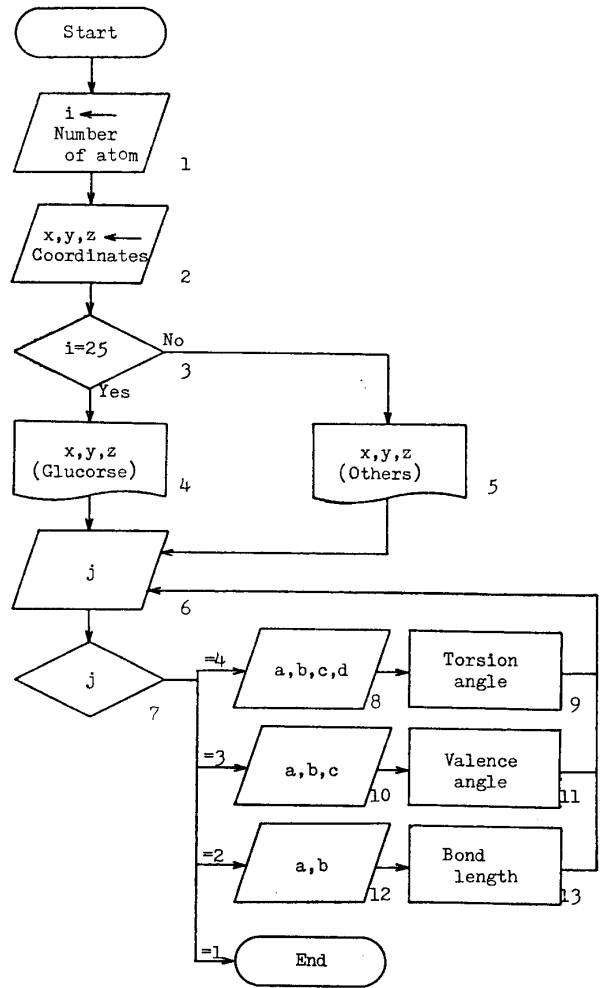


Fig. 5. Flow chart for DLVT. FORT77 program.

II Maltose のポテンシャルエネルギー及び存在確率計算プログラム

II-1 MALTOSSE. FORT

使用言語：FORTRAN (固定型式)

II-1-1 概要

Maltose は、D-glucose が  $\alpha$ -1,4 結合した二糖類である。ピラノース環の構造は堅い (rigid) と考えられるので、マルトースのコンホメーションは、Fig. 8 に示すようなグリコシド結合角  $\tau$  と二つの二面角  $\phi$  と  $\psi$  にのみ依存する。本プログラムにより、任意の結合角  $\tau$  を持つ  $\alpha$ -maltose について、グリコシド結合の  $\phi, \psi$  を変化させながら各コンホメーションの非結合原子間相互作用エネルギーを計算することができる。同時に、プログラム中でサブルーチンと呼ぶことによりエネルギー値から各コンホメーションの存在確率を計算することができる。

結合角  $\tau$ , 二面角  $\phi, \psi$  を、Brant<sup>5)</sup> に従い次のように定義する (Fig. 7)。

```

# I DLVT DATA
KE0528001 AKH0112.DLVT.DATA
j → 04
A → 1.0(x) 0.0(y) 0.0(z)
B → 1.0 1.0 0.0
C → 0.0 1.0 0.0
D → 0.0 1.0 1.0
j → 4
ABCD → 01020304
ACB → 010302
AC → 0103
1
KE0528001 END OF DATA

```

```

# FREE F(FT05F001)
# ALLOC F(FT05F001) DA(DLVT.DATA) SHR
# RUN DLVT.FORT
GE COMPILER ENTERED
END OF COMPILATION

***** DATA *****
(x) (y) (z)
( 1) 1.000000 0.0 0.0
( 2) 1.000000 1.000000 0.0
( 3) 0.0 1.000000 0.0
( 4) 0.0 1.000000 1.000000

TORTION ANGLE= 90.000000 (DEG) ( 1)-( 2)-( 3)-( 4)
VALECE ANGLE= 45.000000 (DEG) ( 1)-( 3)-( 2)
LENGTH= 1.414213 (ANGSTROM) ( 1)-( 3)

END OF GO,SEVERITY CODE= 0

```

Fig. 6. An example of input and output data using DLVT. FORT77 program.

C(1), O(1), C(4') の3原子を含む平面上に O(4), O(1) があり, Fig. 8 に示すような位置関係に各原子がある時,  $\phi=0^\circ$ ,  $\psi=180^\circ$  とする。両残基の回転は矢印の方向を正とした。 $\phi$ ,  $\psi$  は次のように表現される。

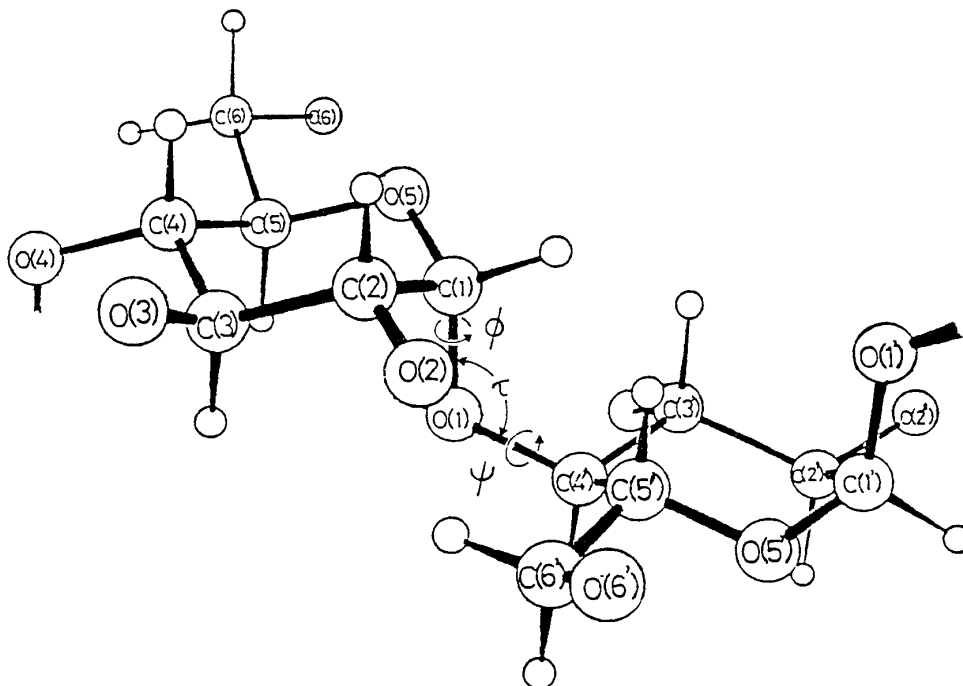


Fig. 7. A dimeric skeletal segment of the amylosic chain, i.e., the maltose unit, in the conformation  $\phi=0^\circ$   $\psi=180^\circ$ .

$$\begin{aligned}\phi &= \theta[C(4'), O(1), C(1), O(4)] + 180^\circ \\ &= \theta[O(4), C(1), O(1), C(4')] + 180^\circ \quad (3)\end{aligned}$$

$$\begin{aligned}\psi &= \theta[C(1), O(1), C(4'), O(1')] + 180^\circ \\ &= \theta[O(1'), C(4'), O(1), C(1)] + 180^\circ \quad (4)\end{aligned}$$

結合角  $\tau$  は, 小さい方の角度 ( $\tau \leq 180^\circ$ ) をとり, 次のように表現される。

$$\tau = \tau[C(1), O(1), C(4')] \quad (5)$$

非結合原子間の相互作用エネルギーを計算するために, 次の4つのサブルーチンを用意した。

#### (1) KITAYG

Kitaygorodski 関数を用いて van der Waals 力による相互作用エネルギーを計算する。関数を次に示す<sup>10)</sup>。

$$\begin{aligned}E_v(r_{m,n}) &= 3.5 \left[ 8600 \exp\left(\frac{-13r_{m,n}}{r_{m,n}^0}\right) \right. \\ &\quad \left. - 0.04 \left(\frac{r_{m,n}}{r_{m,n}^0}\right)^{-6} \right] \quad (6)\end{aligned}$$

ここで,  $r_{m,n}$ : 原子間距離,

$r_{m,n}^0$ : Kitaygorodski の定数

#### (2) LJF

Lennard-Jones “6-12” 型関数を用いて van der Waals 力による相互作用エネルギーを計算する。関数を次に示す<sup>3)</sup>。

$$E_v(r_{m,n}) = \frac{a_{m,n}}{r_{m,n}^{12}} - \frac{c_{m,n}}{r_{m,n}^6} \quad (7)$$

ここで,  $a_{m,n}$ : 反発力の相互作用に関する定数

$c_{m,n}$ : 引力の相互作用に関する定数

### (3) COULOM

Coulomb 力の相互作用エネルギーを計算する。  
関数を次に示す<sup>3)</sup>。

$$E_{el}(r_{m,n}) = \frac{332 q_m q_n}{\epsilon r_{m,n}} \quad (8)$$

ここで,  $q_m, q_n$ : 部分電荷 (partial electronic charge)

$\epsilon$ : 誘電率 (effective dielectric constant)

### (4) TORSIO

Torsional strain による相互作用エネルギーを計算する。“inherent” torsion-strain energy に加え, ゴーシュ (gauche) 効果 (あるいは, “exoanomeric” 効果) も考慮に入れている。関数を次に示す<sup>4)</sup>。

$$E_t(\phi_t) = \left(\frac{K_i^\phi}{2}\right) (1 + \cos 3\phi_t) + \left(\frac{K_g^\phi}{2}\right) (1 - \cos 3\phi_t) \quad (9)$$

$$E_t(\psi_t) = \left(\frac{K_i^\psi}{2}\right) (1 + \cos 3\psi_t) \quad (9')$$

ここで,  $K_i^\phi, K_i^\psi$ : “inherent” torsion-strain energy に関する定数

$K_g^\phi$ : ゴーシュ効果に関する定数

$\phi_t$ :  $\theta$ [H(1), C(1), O(1), C(4')]

$\psi_t$ :  $\theta$ [C(1), O(1), C(4'), H(4')]

確率変換サブルーチン PROBAB について次に述べる。本サブルーチンは, 二面角  $\phi, \psi$  によって決まる各コンホメーションの存在確率を計算する。計算式を次に示す<sup>5)</sup>。

$$P_i = 100 \times \frac{\exp\{-V_i/RT\}}{\sum_{\phi_i} \sum_{\psi_i} \exp\{-V_i/RT\}} \quad (10)$$

ここで,  $\phi_i, \psi_i$ : コンホメーション状態  $i$  の  $\phi, \psi$

$P_i$ : コンホメーション状態  $i$  の存在確率

$V_i$ : コンホメーション状態  $i$  のポテンシャルエネルギー

$R$ : 気体定数

$T$ : 絶対温度

## II-1-2 フローチャート

計算過程のフローチャートを Fig. 8 に示す。以下

フローチャートに付した番号に従ってその内容を説明する。

1. 座標データ (プログラム AMALT2 で得られた  $\alpha$ -maltose ( $\tau=180^\circ$ ) の極座標), 関数のパラメータをファイルとデータ文より読み込む。
2. 読み込んだデータをプリンタに出力する。
3. 初期値の設定及び, 度とラジアンの変換等を行なう。
4. 還元残基の回転操作を行なう。Fig. 9 に示すように, 還元残基を  $z$  軸まわりに  $\psi$  だけ回転し, 次に  $y$  軸まわりに ( $\pi-\tau$ ) だけ回転する。同時に原子座標を極座標から直交座標に変換する。
5. 非還元残基の回転操作を行なう。Fig. 9 に示すように, 非還元残基を  $z$  軸まわりに  $\phi$  だけ回転する。同時に原子座標を極座標から直交座標に変換する。
6. 4 と 5 の回転操作で得られた  $\alpha$ -maltose の直交座標を使って非結合原子間の相互作用エネルギーを計算する。
7.  $\phi$  が指定した計算範囲内ならば,  $\phi$  に角度 (計算間隔) C2 を加えて 5 へもどる。
8.  $\psi$  が計算範囲内ならば,  $\psi$  に角度 C2 を加えて 4 にもどる。
9. 得られたエネルギー値をファイル 1 に格納する。
10. サブルーチン PROBAB を読んでエネルギー値から存在確率を計算する。
11. (以下サブルーチン PROBAB) エネルギー値から各コンホメーションの存在確率を計算する。
12. 得られた存在確率値をファイル 2 に格納する。

## II-1-3 サブルーチン

### (1) KITAYG

機能

(6)式を使って, van der Waals 力による相互作用エネルギーを計算する。

呼び出し型式

CALL KITAYG (I1, I2, J1, J2, R0, A, B, TOTALV, N, L)

引き数の説明

I1, I2: 相互作用を考える非還元残基の原子を指定する変数。整数型。入力。

J1, J2: 相互作用を考える還元残基の原子を指定する変数。整数型。入力。

R0: Kitaygorodski の定数。実数型。入力。

A, B: 非還元残基 (A), 還元残基 (B) に格納されている変数名。実数型二次元配列, 入力。

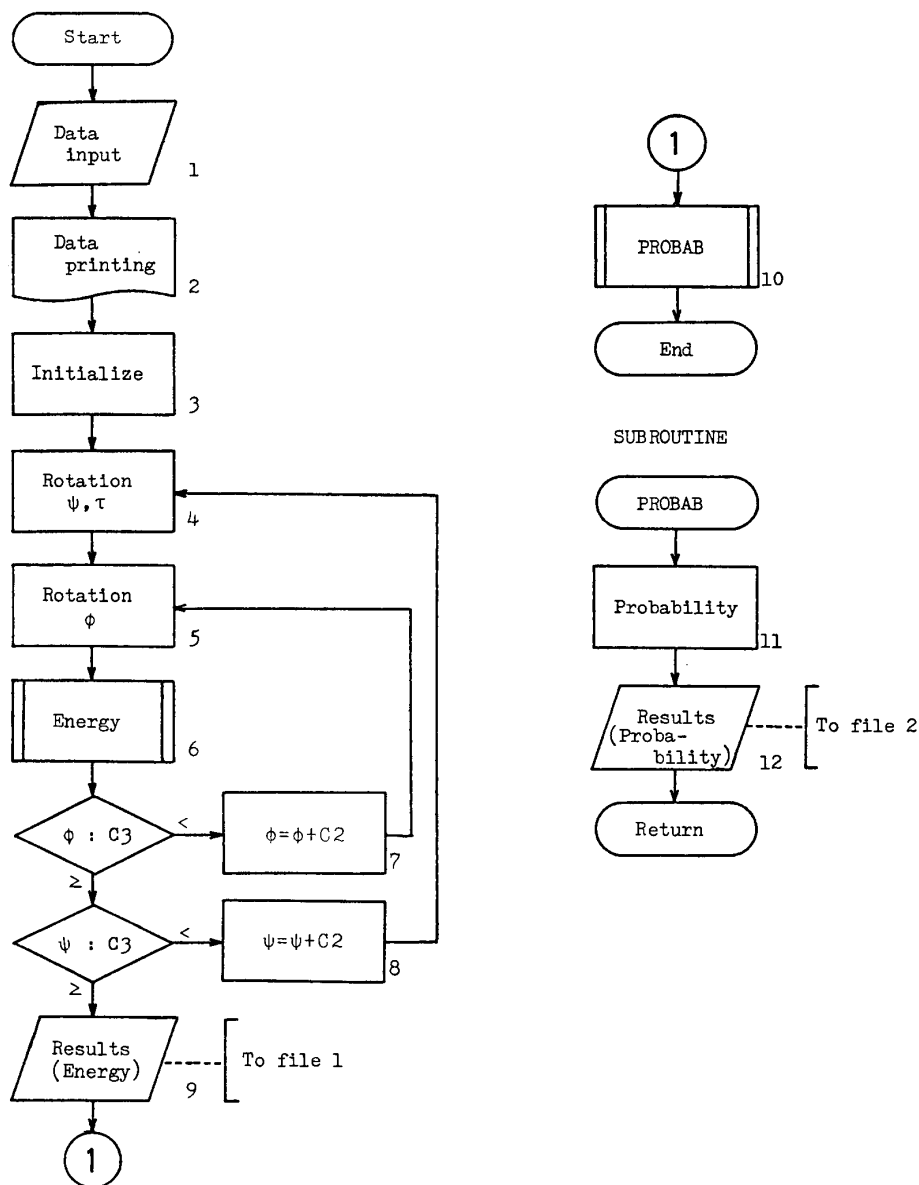


Fig. 8. Flow chart for MALTOSE. FORT program.

TOTALV : 計算結果(エネルギーを格納する変数名)。実数型二次元配列。出力。

N, L : TOTALV の配列要素の添字。整数型。入力。

(2) LJF

機能

(7)式を使って, van der Waals 力による相互作用エネルギーを計算する。

呼び出し型式

CALL LJF (I1, I2, J1, J2, CA, CC, A, B, TOTALV, N, L)

引数の説明

I1, I2, J1, J2, A, B, TOTALV, N, L : KITAYG に同じ

CA : constant A。 (7)式中の  $a_{m,n}$  のことで, 反発力の相互作用に関する定数。実数型。入力。

CC : constant C。 (7)式中の  $c_{m,n}$  のことで, 引力の相互作用に関する定数。実数型。入力。

(3) COULOM

機能

(8)式を使って Coulomb 力による相互作用エネルギーを計算する。

呼び出し型式

CALL COULOM (I1, I2, J1, J2, Q1, Q2, A, B, TOTALV, N, L)

引き数の説明

I1, I2, J1, J2, A, B, TOTALV, N, L : KITAYG に同じ。

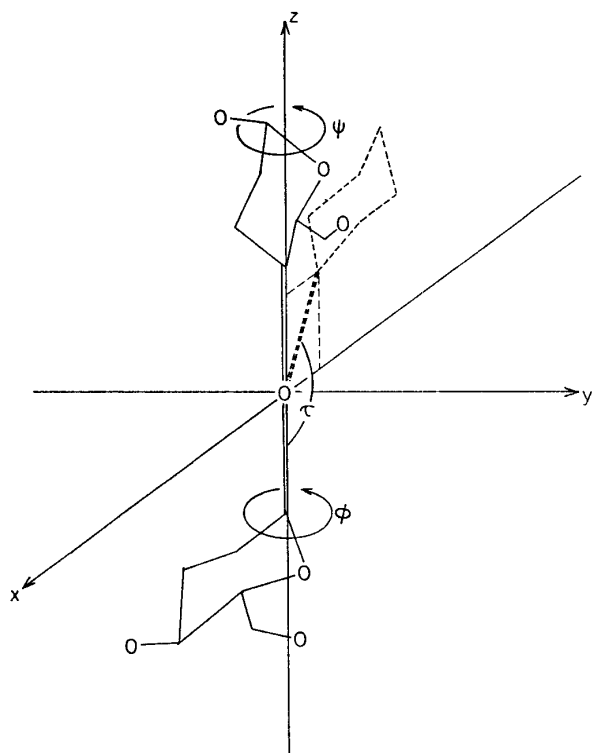


Fig. 9. Rotations of a pair of rotational angles  $\phi$  and  $\psi$ , and bridge angle of glucosidic linkage  $\tau$ .

Q1, Q2: (8)式中の  $q_m, q_n$  のことであり, 部分電荷を意味する。実数型。入力。

#### (4) TORSIO

機能

(9)(9')式を使って, torsional strain による相互作用エネルギーを計算する。

呼び出し型式

CALL TORSIO (M, NN, DPHI, DPSI, TOTALV, N, L)

引数の説明

M, NN: 二面角  $\psi, \phi$ 。整数型。入力。

DPHI:  $\phi_t - \phi$ 。メインプログラム中で定義した二面角  $\phi$  を, (9)式で定義した二面角  $\phi_t$  に変換するための定数。実数型。入力。

DPSI:  $\psi_t - \psi$ 。DPHI と同様にして  $\psi_t$  を  $\psi$  に変換するための定数。実数型。入力。

TOTALV, N, L: KITAYG に同じ。

#### (5) PROBAB

機能

(10)式を使って, エネルギー値を確率分布に変換する。

呼び出し型式

CALL PROBAB (L1, N1, TOTALV, C1, C2, C3, R, T)

引数の説明

L1: TOTALV の  $\psi$  の配列の個数。整数型。入力。

N1: TOTALV の  $\phi$  の配列の個数。整数型。入力。

TOTALV: 各コンホメーション ( $\phi, \psi$ ) のポテンシャルエネルギー。実数型二次元配列。入力。

C1: 結合角  $\tau$  の値。実数型。入力。

C2: 計算間隔。実数型。入力。

C3: 計算範囲。実数型。入力。

R: 気体定数。実数型。入力。

T: 絶対温度。実数型。入力。

### II-2 XYPLOT. FORT

使用言語: FORTRAN (固定型式)

京都大学大型計算機センター・プログラム・ライブラリ J6/CONTOR (XY プロットによる等高線の作図) を使用。

#### II-2-1 概要

本プログラムは, MALTOSE. FORT により計算されたコンホメーションエネルギーあるいは存在確率を XY プロットに等高線地図として出力するためのものである。XY プロットは, 京都大学大型計算機センター (以下センターと略す) の VERSATEC-1200 を使用した。プログラムは, センターが用意した図型出力用ライブラリ (プロット・ライブラリ)<sup>12)</sup> 及びライブラリ (J6/CONTOR)<sup>13)</sup> を使用して作成した。ジョブは, バッチジョブで依頼する。

プログラムは 4 種類ある。次にプログラムのメンバー名及び機能を出力例と共に示す。

##### (1) CONTORE (エネルギー地図作成プログラム)

ファイル 1 に格納されている MALTOSE. FORT の計算結果 ( $\alpha$ -maltose のポテンシャルエネルギー) を読み込み, エネルギー地図として印刷する。地図は, 横軸を  $\phi$ , 縦軸を  $\psi$  にとり, 等エネルギー線を等高線として表示する (Fig. 10)。

##### (2) CONTORP (確率地図作成プログラム)

ファイル 2 に格納されている MALTOSE. FORT の計算結果 ( $\alpha$ -maltose の各コンホメーションの存在確率) を読み込み, 確率地図を出力する。CONTORE と同様に, 横軸を  $\phi$ , 縦軸を  $\psi$  にとり, 等確率線を等高線として表示する。

##### (3) VALUEE (エネルギー値出力プログラム)

エネルギー値をファイル 1 から読み込みエネルギー地図と同じスケールで, 数値を単位 kcal/mol でひとつおきに出力する。ただし, 100 kcal/mol 以上



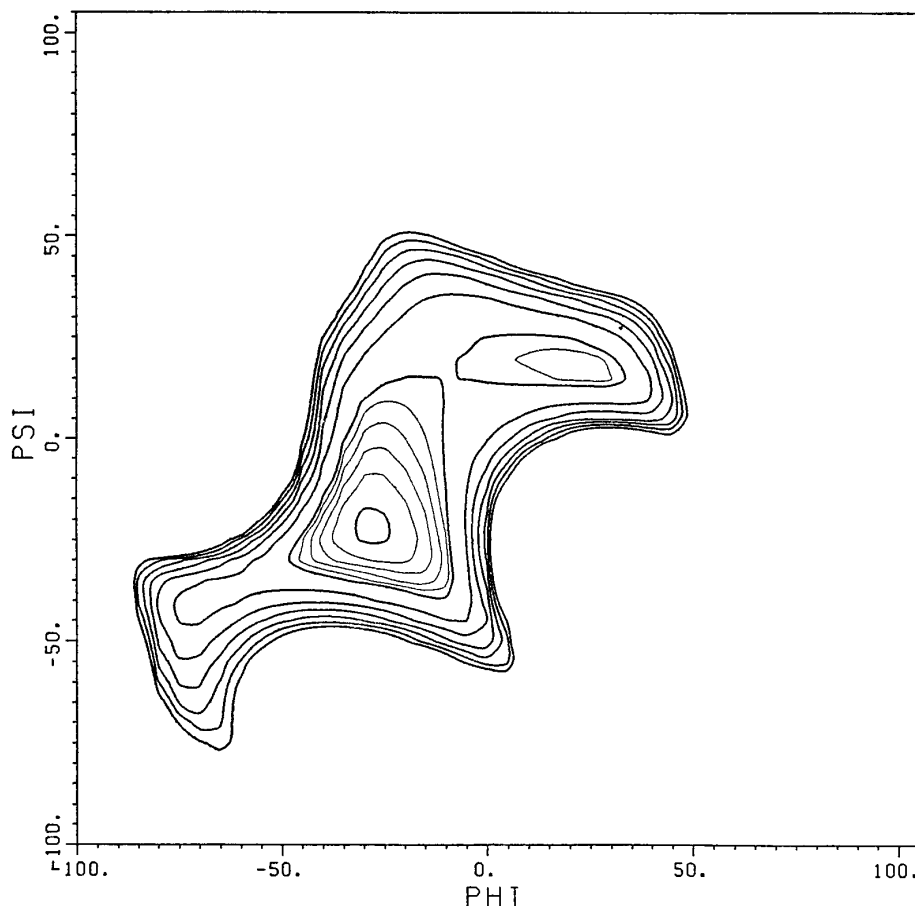


Fig. 10. An example of conformational energy map for maltose unit. The geometry used for the maltose unit was determined from neutron scattering study of  $\beta$ -maltose monohydrate.  $\theta=118$ . Contours lie in the energy portion under  $5.0 \text{ kcal mol}^{-1}$ , where solid contours are given at intervals of  $1 \text{ kcal mol}^{-1}$  for  $\geq -1 \text{ kcal mol}^{-1}$  and  $0.2 \text{ kcal mol}^{-1}$  for  $< -1 \text{ kcal mol}^{-1}$ , respectively.

の数値はプロットされない。

(4) VALUEP (存在確率値出力プログラム)

存在確率をファイル2から読み込み、確率地図と同じスケールで、数値をパーセントでひとつおきに表示する。

Ⅲ モンテカルロ法によるアミロース分子鎖作成プログラム TAC. FORT77 (MAIN)

使用言語: FORTRAN (自由型式)

Ⅲ-1 概要

本プログラムは、高分子統計力学的な手法を基礎として、モンテカルロ法によりアミロース分子鎖を生成するものである。

アミロースは、D-glucose が  $\alpha$ -1,4 結合で結合した直鎖状多糖である。ピラノース環が、常に C1 コンホメーションをとると仮定すると、アミロースのコン

ホメーションは、グリコシド結合の回転の自由度にのみ支配される。また、グリコシド結合の二つの回転角(二面角)  $\phi$ ,  $\psi$  は、隣接残基間の非結合原子間の相互作用によって支配されると考えられる。それゆえ、 $\alpha$ -maltose の非結合原子間のポテンシャルエネルギーの総和より計算されるコンホメーションエネルギー地図から、アミロースのグリコシド結合の二面角( $\phi$ ,  $\psi$ )の存在確率を予想することができる。アミロース分子鎖内のひとつのグリコシド結合の状態を  $j$  で表わすと、グリコシド結合が状態  $j$  をとり得る確率  $P_j$  は次式で与えられる<sup>6)</sup>。

$$P_j = \frac{\exp\left[-\frac{V(\phi_j, \psi_j)}{RT}\right]}{\sum_j \exp\left[-\frac{V(\phi_j, \psi_j)}{RT}\right]} \quad (11)$$

ここで、 $V(\phi_j, \psi_j)$  はグリコシド結合の二つの二面角が、 $\phi_j, \psi_j$  である状態  $j$  のポテンシャルエネルギー

一であり、 $\sum_j$ は一個のグリコシド結合がとり得るすべての状態についての和を表わす。また、 $\sum_j P_j = 1$  である。この場合に、 $(0, 1)$  の一様乱数  $\varepsilon_i (i=1, 2, \dots, x-1)$  を発生し、 $\varepsilon_i$  を以下の手順で  $\phi_i$  と  $\psi_i$  に変換すれば、確率分布  $P(\phi, \psi)$  に従って分布する  $\phi_i$  と  $\psi_i$  の  $x-1$  個の組  $\{\phi_i, \psi_i\} (i=1, 2, \dots, x-1)$  が得られる。

$$\begin{array}{l} 0 \leq \varepsilon_i \leq P_1 \text{ のとき } \phi_i = \phi_1, \psi_i = \psi_1 \\ \vdots \\ P_{j-1} \leq \varepsilon_i \leq P_j \text{ のとき } \phi_i = \phi_j, \psi_i = \psi_j \\ \vdots \end{array}$$

$\{\phi_i, \psi_i\}, (i=1, 2, \dots, x-1)$  は、座標変換マトリックス  $\{T_i\}, (i=1, 2, \dots, x-1)$  と対応づけられる。座標変換マトリックス  $T_i$  は、グルコース残基  $i$  に固定した座標系から、グルコース残基  $i-1$  への座標変換をするものである。 $T_i$  は次式で与えられる<sup>6)</sup>。

$$\begin{aligned} T_i = & R_x(\omega) R_z(\eta) R_x(\pi - \phi_i) R_z(\pi - \theta) \\ & R_x(\pi - \psi_i) R_z(\xi) \end{aligned} \quad (12)$$

ただし、 $x$  軸まわりの回転マトリックス  $R_x(\rho)$ 、 $z$  軸まわりの回転マトリックス  $R_z(\tau)$  はそれぞれ

$$R_x(\rho) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \rho & \sin \rho \\ 0 & -\sin \rho & \cos \rho \end{bmatrix} \quad (13)$$

$$R_z(\tau) = \begin{bmatrix} \cos \tau & \sin \tau & 0 \\ -\sin \tau & \cos \tau & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (14)$$

ここで、 $\omega$  は二面角  $\theta[C(4), O(4), O(1), C(1)]$  であり、 $\eta, \theta, \xi$  はそれぞれ  $\tau[C(1), O(1), C(4')]$ 、 $\tau[O(4), O(1), C(1)]$ 、 $\tau[C(4'), O(1), O(1')]$  である。 $\phi, \psi$  は、上記のグリコシド結合の二面角であり、次式で与えられる。

$$\phi = \theta[C(4'), O(1), C(1), O(4)] + 180^\circ \quad (15)$$

$$\psi = \theta[C(1), O(1), C(4'), O(1')] + 180^\circ \quad (16)$$

また、グルコース残基  $i+1$  の  $O(1)$  の位置ベクトル  $O(1)_{i+1}$  は次式で与えられる。

$$O(1)_{i+1} = l + T_1 l + T_1 T_2 l + \dots + T_1 T_2 \dots T_i l \quad (17)$$

この場合に、非還元末端の酸素原子の座標は  $(0, 0, 0)$  である。ここで、マトリックス  $A_k$  を以下のように定義する。

$$A_k = \begin{bmatrix} T_k & l \\ 0 & 1 \end{bmatrix} \quad (18)$$

$0$  は  $1 \times 3$  の  $0$  ベクトルを表わす。マトリックス  $A_k$  を用いて(17)式を書き換えると次式が得られる。

$$O(1)_{i+1} = [T_1, l] \left[ \prod_{k=2}^i A_k \right] \begin{bmatrix} l \\ 1 \end{bmatrix} \quad (19)$$

(19) 式の操作を  $i=1$  から  $x-1$  まで繰り返すと、 $O(1)_2$  から  $O(1)_x$  までのベクトルが得られ、重合度  $x$  のアミロース分子鎖のコンホメーションを仮想ボンダが連結した形で描くことができる。また、(19)式の操作を  $N$  回繰り返すと、 $N$  個のサンプルについて末端間距離が算出される。さらに、等しい末端間距離を持つサンプルを計算することによって末端間距離の分布関数  $W(R)$  をヒストグラムの形で得ることができる。また、分子鎖の広がりを表わすパラメータとして特性比 (Characteristic ratio) を次式により計算する。

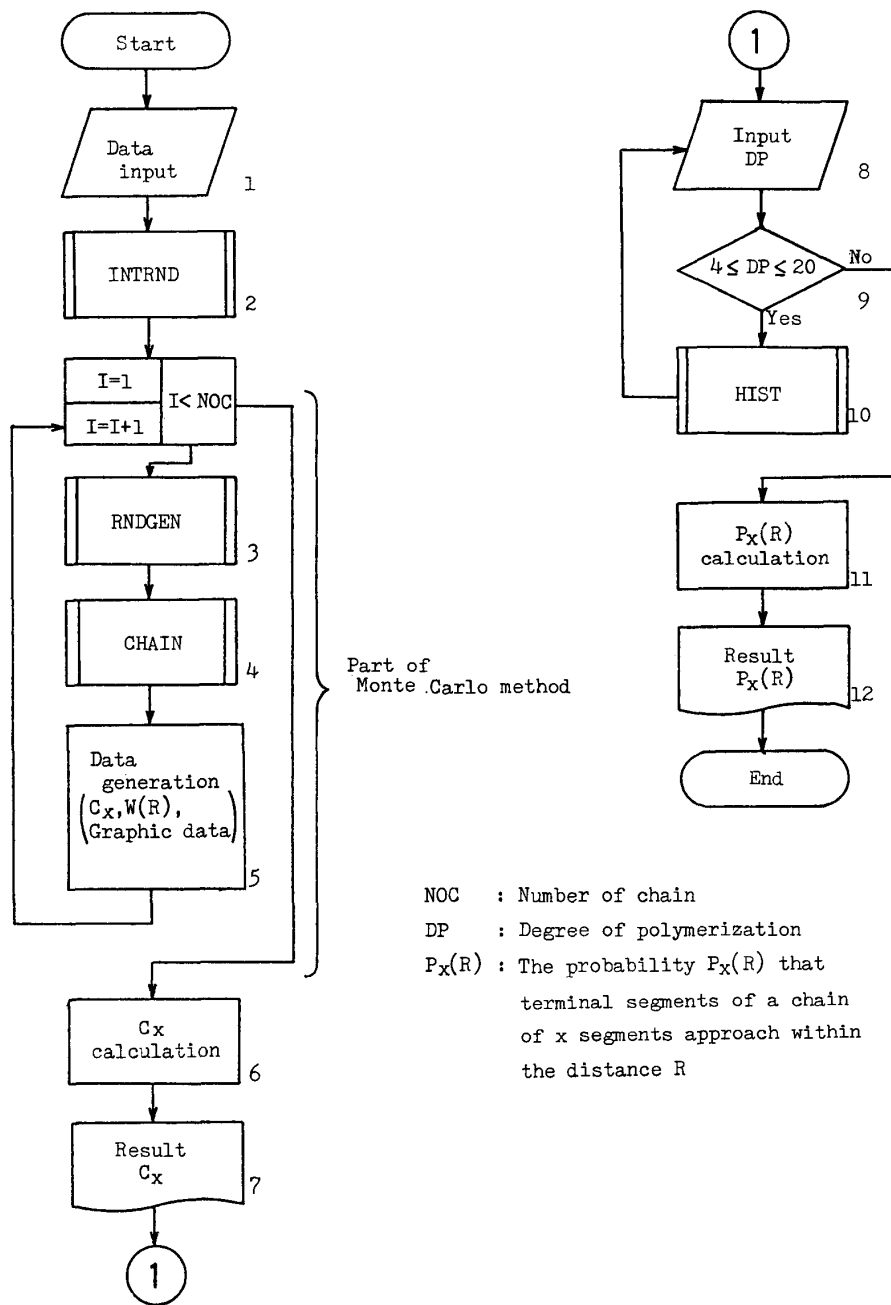
$$C_x = \frac{\langle R^2 \rangle}{x l^2} \quad (20)$$

ここで、 $\langle R^2 \rangle$  は高分子鎖の末端間距離の二乗平均、 $l$  は仮想ボンダ長、 $x$  は重合度である。

### III-2 フローチャート

Fig. 11 に計算過程のフローチャートを示す。以下フローチャートに付した番号に従ってその内容を説明する。

1.  $(\phi, \psi)$  の存在確率、重合度、鎖の本数、その他計算に必要なパラメータを読み込む。
2. サブルーチン INTRND を呼んで存在確率のデータを初期化する。
- 3~5. モンテカルロ法によるアミロース鎖の作成。サブルーチン RNDGEN, CHAIN を呼んでアミロース鎖を作成し、 $C_x$  および  $W(R)$  の計算、グラフィック表示 (V 章参照) のためのデータを作成する。この過程を、ループによって鎖の本数 (NOC) 回繰り返す。
- 6, 7.  $C_x$  を計算し、プリンタに出力する。
- 8~10. 重合度 ( $4 \leq DP \leq 20$ ) を入力すると、サブルーチン HIST が呼ばれて、 $W(R)$  がヒストグラムとして出力される。
- 11, 12. 重合度  $x$  ( $4 \leq x \leq 20$ ) の両末端の距離  $R$  が  $3\text{\AA} \sim 7\text{\AA}$  以下になる確率  $P_x(R)$  を計算する。Fig. 12 に計算結果の出力例を示す。



NOC : Number of chain  
 DP : Degree of polymerization  
 $P_x(R)$  : The probability  $P_x(R)$  that terminal segments of a chain of  $x$  segments approach within the distance  $R$

Fig. 11. Flow chart for TAC.FORT77 (MAIN) program.

Ⅲ-3 サブルーチン  
 Ⅳ-3 を参照。

Ⅳ 変換マトリックス  $\langle T \rangle$  によるアミロースの分子ディメンションの算出 TAC.FORT77 (CXPER)

使用言語: FORTRAN (自由型式)

Ⅳ-1 概要

分子鎖の特性比, あるいは, 持続ベクトルは, Flory

らによって確立された高分子統計力学的方法を用いて計算することができる。

アミロース鎖の各グルコース残基には, 個々の座標系が定義されている。 $i$  番目の残基の座標系を  $i-1$  番目の残基の座標系に変換する変換マトリックス  $T$  は, 変数  $\phi, \psi$  の関数となる。ゆえに, 座標変換マトリックス  $T$  の統計力学的平均  $\langle T \rangle$  は,

$$\langle T \rangle = \sum_{\phi_i} \sum_{\psi_i} P(\phi_i, \psi_i) T_i \quad (21)$$

で求められる。本プログラムは, この  $\langle T \rangle$  を使って,

```

E
FREE F(F107F001)
E
ALLOC F(F107F001) DA(CHAIN3,DATA*G,USEF)
E
RUN LIB(TACLIB)
FORTRAN 77 COMPILER ENTERED
END OF COMPILATION
-----
01600 ?
200
    << NOC=? >>
01620 ?
5000
    << DO YOU WANT GRAPHIC DATA ?? >>
        YES= 1.0  NO=0.0
01650 ?
1.0
***** INPUT DATA *****
THETA= 118.000000  ETA= 70.4627502  XI= 39.4783305  OMEGA= 6.0670744
DP= 200  NUMBER FO CHAINS= 5000  BOND LENGTH= 4.40988037
***** << CHARACTERISTIC RATIO (C) >> *****
DP= 200  C(DP)= 4.83656502
DP= 1  C(DP)= 0.999743342
DP= 2  C(DP)= 1.705181112
DP= 3  C(DP)= 2.01268578
DP= 4  C(DP)= 2.00546455
DP= 5  C(DP)= 1.87300301
DP= 6  C(DP)= 1.80023479
DP= 7  C(DP)= 1.87741852
DP= 8  C(DP)= 2.07365227
DP= 9  C(DP)= 2.29855156
DP= 10  C(DP)= 2.47229862
DP= 11  C(DP)= 2.58734415
DP= 12  C(DP)= 2.61660576
DP= 13  C(DP)= 2.66675949
DP= 14  C(DP)= 2.74964237
DP= 15  C(DP)= 2.85719204
DP= 16  C(DP)= 2.96753788
DP= 17  C(DP)= 3.05788517
DP= 18  C(DP)= 3.12513542
DP= 19  C(DP)= 3.18190098
DP= 20  C(DP)= 3.24258709
DP= 30  C(DP)= 3.75009823
DP= 40  C(DP)= 4.03584290
DP= 50  C(DP)= 4.24811268
DP= 60  C(DP)= 4.38851261
DP= 70  C(DP)= 4.50116634
DP= 80  C(DP)= 4.54957294
DP= 90  C(DP)= 4.58875370
DP= 100  C(DP)= 4.6099001
DP= 150  C(DP)= 4.76427555
DP= 200  C(DP)= 4.83656502
*** DO YOU WANT THE HISTGRAM W(R) (4<R(20) ??) ***
DP= ?
01660 ?
20
***** HISTGRAM OF DISTRIBUTION FUNCTION, W(R) *****
DP=20  NUMBER OF CHAINS = 5000
INTERVAL  W(R)  0  10  20  30  40  50  60  70  80  90  100
2 - 4  3  I  I  I  I  I  I  I  I  I  I
4 - 6  9  I * I  I  I  I  I  I  I  I  I
6 - 8  16  I * I  I  I  I  I  I  I  I  I
8 - 10  33  I * I  I  I  I  I  I  I  I  I
10 - 12  36  I * I  I  I  I  I  I  I  I  I
12 - 14  49  I * I  I  I  I  I  I  I  I  I
14 - 16  74  I * I  I  I  I  I  I  I  I  I
16 - 18  104  I * I  I  I  I  I  I  I  I  I
18 - 20  126  I * I  I  I  I  I  I  I  I  I
20 - 22  178  I * I  I  I  I  I  I  I  I  I
22 - 24  222  I * I  I  I  I  I  I  I  I  I
24 - 26  244  I * I  I  I  I  I  I  I  I  I
26 - 28  312  I * I  I  I  I  I  I  I  I  I
28 - 30  331  I * I  I  I  I  I  I  I  I  I
30 - 32  339  I * I  I  I  I  I  I  I  I  I
32 - 34  349  I * I  I  I  I  I  I  I  I  I
34 - 36  369  I * I  I  I  I  I  I  I  I  I
36 - 38  348  I * I  I  I  I  I  I  I  I  I
38 - 40  373  I * I  I  I  I  I  I  I  I  I
40 - 42  330  I * I  I  I  I  I  I  I  I  I
42 - 44  306  I * I  I  I  I  I  I  I  I  I
44 - 46  237  I * I  I  I  I  I  I  I  I  I
46 - 48  198  I * I  I  I  I  I  I  I  I  I
48 - 50  142  I * I  I  I  I  I  I  I  I  I
50 - 52  114  I * I  I  I  I  I  I  I  I  I
52 - 54  85  I * I  I  I  I  I  I  I  I  I
54 - 56  44  I * I  I  I  I  I  I  I  I  I
56 - 58  23  I * I  I  I  I  I  I  I  I  I
58 - 60  4  I * I  I  I  I  I  I  I  I  I
60 - 62  1  I * I  I  I  I  I  I  I  I  I
62 - 64  1  I * I  I  I  I  I  I  I  I  I
*** DO YOU WANT THE HISTGRAM W(R) (4<R(20) ??) ***
DP= ?
01660 ?
0
***** << PROBABILITY P(R) (NOC= 5000) >> *****
P( 3 ) 0.0 0.0 0.0 0.16 1.58 0.62 0.16 0.06 0.06 0.04 0.0 0.06 0.02 0.08 0.0 0.04 0.04
P( 4 ) 0.0 0.0 0.0 0.70 2.94 1.52 0.60 0.20 0.10 0.06 0.04 0.14 0.10 0.12 0.06 0.06 0.06
P( 5 ) 0.0 0.0 0.0 0.04 4.16 2.82 1.14 0.36 0.22 0.10 0.06 0.16 0.20 0.16 0.16 0.10 0.10
P( 6 ) 0.0 0.0 0.0 0.20 4.10 6.16 4.34 1.78 0.72 0.42 0.28 0.16 0.28 0.48 0.30 0.32 0.20 0.24
P( 7 ) 0.0 0.0 0.0 1.00 6.38 8.04 5.80 2.50 1.02 0.60 0.36 0.38 0.50 0.80 0.56 0.52 0.48 0.46
END OF GO, SEVERITY CODE=00

```

Fig. 12. An example of output using TAC. FORT77 (MAIN) program.

アミロース分子鎖のキャラクタリゼーションをおこなうためのものである。

分子鎖の特性比  $C_x$ ,  $C_\infty$ , および, 持続ベクトル  $\mathbf{a}$ , 持続長  $a$  は, 次式で与えられる。

$$C_x = [(E + \langle T \rangle)(E - \langle T \rangle)^{-1} - \frac{2\langle T \rangle}{x}(E - \langle T \rangle)^x (E - \langle T \rangle)^{-2}]_{11} \quad (22)$$

$$C_\infty = [(E + \langle T \rangle)(E - \langle T \rangle)^{-1}]_{11} \quad (23)$$

$$\mathbf{a} = \sum_{i \geq j} \langle T \rangle^{i-j} \mathbf{l} \quad (24)$$

$$a = |\mathbf{a}| \quad (25)$$

ここで, 添字の 11 は, マトリックスの 1, 1 要素を表わし,  $E$  は  $3 \times 3$  の単位マトリックス,  $\mathbf{l}$  は仮想ボンドに固定された座標系における仮想ボンドベクトルで  $(l, 0, 0)$  で与えられる。 $l$  は仮想ボンド長である。

#### N-2 フローチャート

Fig. 13 に計算過程のフローチャートを示す。以下フローチャートに付した番号に従ってその内容を説明する。

1.  $(\phi, \psi)$  の存在確率, 計算に必要なパラメータを読み込む。さらに目的とする重合度 (MDP) を入力する。
2. サブルーチン ACTMT を呼んで  $\langle T \rangle$  を計算する。
3. サブルーチン MSETED を呼んで重合度が無限大のときの特性比  $C_\infty$  を計算し, 結果を出力する。
4. 重合度が正しく与えられていない時, 終了する。
- 5, 6. サブルーチン MSETED を呼んで重合度 1 以上 19 以下の特性比  $C_x$  をすべて計算する。
- 7, 8. サブルーチン MSETED を呼んで重合度 20 以上 MDP 以下の特性比  $C_x$  を 10 おきに計算する。
9. サブルーチン PERSIS を呼んで, 持続長  $a$ , 持続ベクトル  $\mathbf{a}$  を計算する。

計算結果の例を Fig. 14 に示す。

#### N-3 サブルーチン

##### (1) ACTMT

###### 機能

グリコシド結合の二面角  $(\phi, \psi)$  の存在確率データ, 及び, 座標変換のパラメータ  $\theta, \eta, \xi, \omega$  から  $\langle T \rangle$  を計算する。このサブルーチンは, サブルーチン INTRND, DCTMT, DIRECT, STAR, 及び, 科学用サブルーチンライブラリ SSL II<sup>14)</sup> より MGGM を使用している。

呼び出し型式

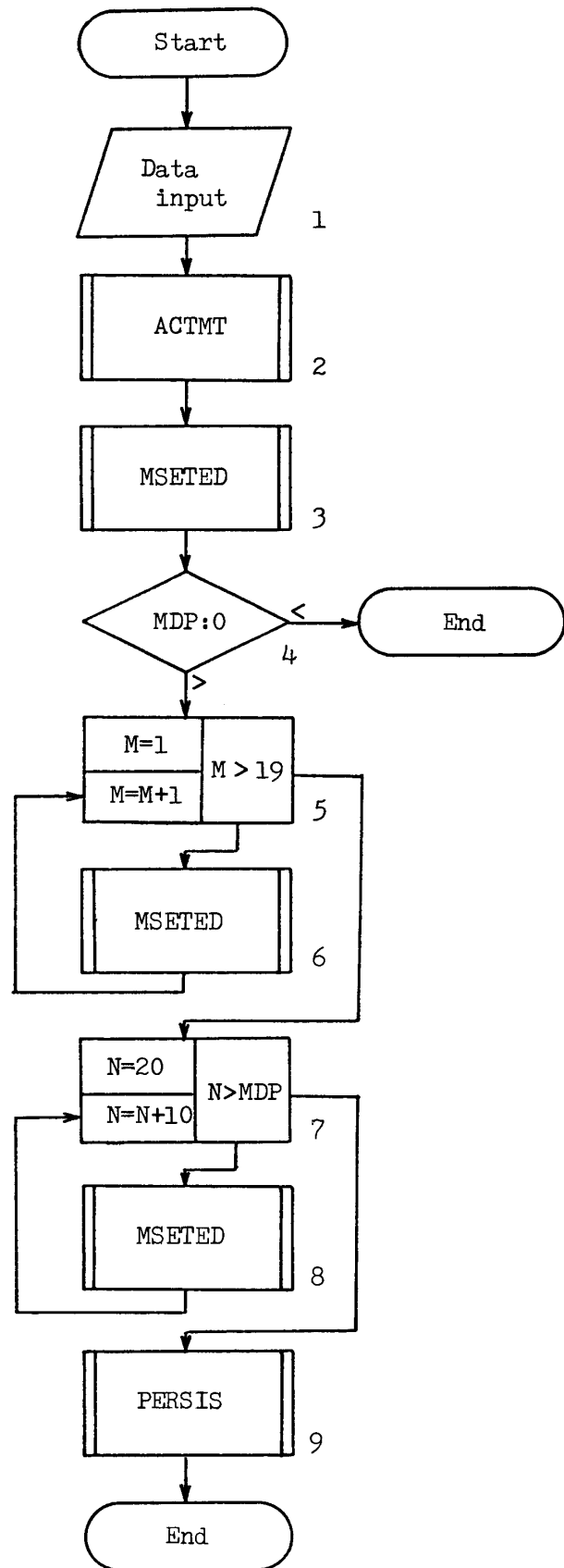


Fig. 13. Flow chart for TAC. FORT77 (CXPER) program.

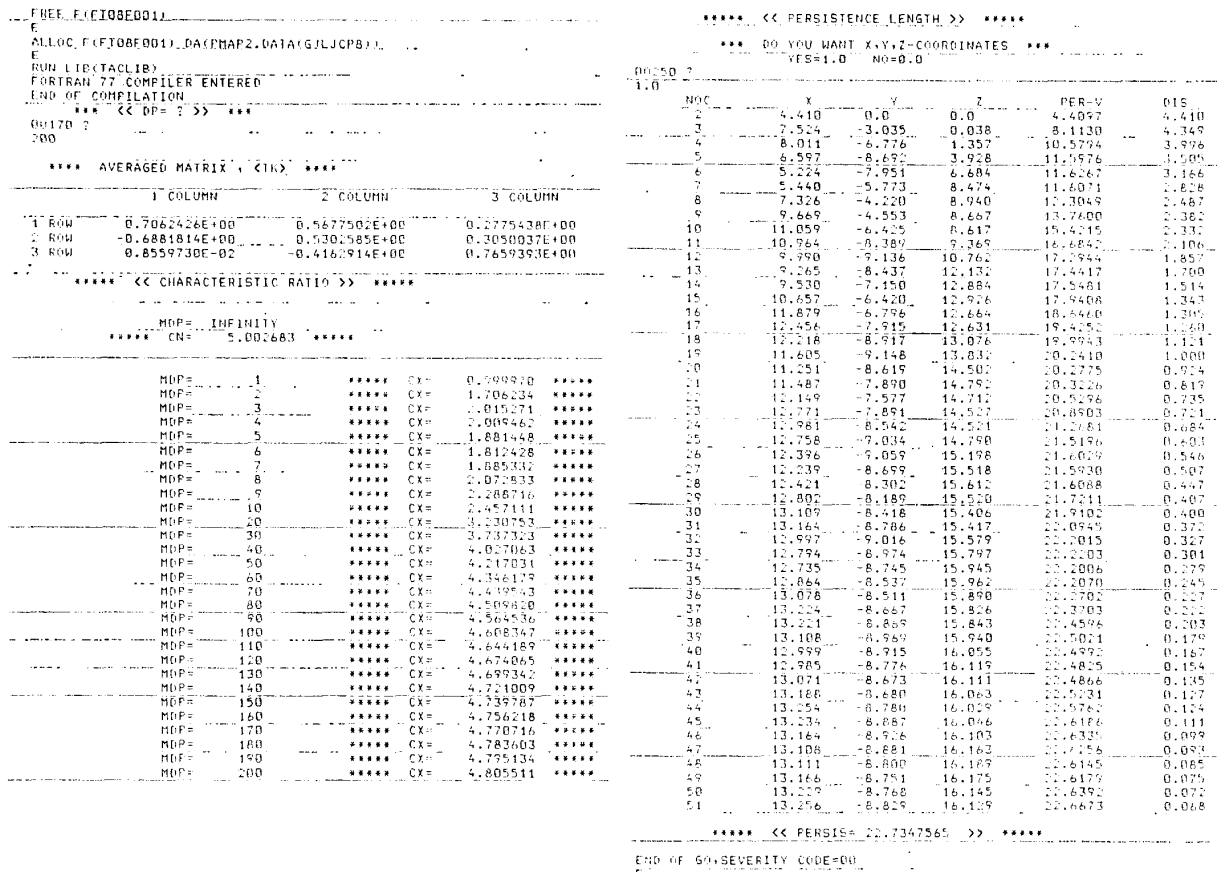


Fig. 14. An example of output data using TAC. FORT77 (CXPER) program.

CALL ACTMT (IX, IY, ID, N, P, ATK, WRT)

引数の説明

IX, IY: それぞれ  $\phi$ ,  $\psi$  の最小値  $\phi_0$ ,  $\psi_0$ . 整数型。入力。

ID:  $\phi$ ,  $\psi$  の計算間隔。整数型。入力。

N: 存在確率のデータの個数の平方根。つまり、データを計算した  $\phi$  及び  $\psi$  の個数。整数型。入力。

P: ( $\phi$ ,  $\psi$ ) の存在確率。実数型一次元配列。入力。

ATK:  $\langle T \rangle$ 。ATK (3, 3) なる実数型二次元配列。出力。

WRT: 書き出し制御変数。1.0 の時  $\langle T \rangle$  を出力。0.0 の時は  $\langle T \rangle$  を出力しない。実数型。入力。

(2) ADD

機能

行列の和を計算するサブルーチン。

呼び出し型式

CALL ADD (A, B, ADDAB, IC, IL)

引数の説明

A, B: 和を求めようとしている二つの行列 **A**, **B**

の変数名。実数型二次元配列。入力。

ADDAB: 行列 **A**, **B** の和。実数型二次元配列。出力。

IC, IL: 行列 **A**, **B** の行数と列数。整数型。入力。

(3) CHAIN

機能

サブルーチン RNDGEN で生成した重合度  $m$  のアミロース鎖のグリコシド結合の二面角 ( $\phi_i$ ,  $\psi_i$ ) ( $i=1, 2, \dots, m$ ) を用いて、グリコシド結合の酸素原子の座標を生成する。サブルーチンを 1 回呼ぶ毎に 1 本の鎖が生成される。このサブルーチンは、サブルーチン DCTMT 及び MGGM (ACTMT 参照) を使用している。

呼び出し型式

CALL CHAIN (MPHI, MPSI, MDP, VLEN, COOX, COOY, COOZ, THE, ETA, XI, OME, WRT, L)

引数の説明

MPHI, MPSI:  $\phi_i$ ,  $\psi_i$ . 整数型一次元配列。入力。

MDP: 重合度 ( $m$ )。整数型。入力。

VLEN: 仮想ボンドの長さ。実数型。入力。

COOX, COOY, COOZ: 酸素原子の座標。実数型一次元配列。出力。

THE, ETA, XI, OME: 座標変換パラメータ  $\theta$ ,  $\eta$ ,  $\xi$ ,  $\omega$ 。実数型。入力。

WRT: 書き出し制御用変数。1.0 のとき座標値を出力し, 0.0 のとき出力しない。

L: MPHI, MPSI, COOX, COOY, COOZ の整合寸法。100,000  $\geq$  L  $\geq$  MDP。整数型。入力。

#### (4) CTMT (DCTMT, QCTMT)

機能

$\phi$ ,  $\psi$ ,  $\theta$ ,  $\eta$ ,  $\xi$ ,  $\omega$  から座標変換マトリックス  $T$  を生成する。CTMT は単精度計算型, DCTMT は倍精度計算型, QCTMT は倍々精度計算型のサブルーチンである。

呼び出し型式

CALL CTMT (PHI, PSI, THE, ETA, XI, OME, TK)

引数の説明

PHI, PSI, THE, ETA, XI, OME: それぞれ  $\phi$ ,  $\psi$ ,  $\theta$ ,  $\eta$ ,  $\xi$ ,  $\omega$ 。実数型。入力。

TK: 座標変換マトリックス  $T$ 。TK (3, 3) なる実数型二次元配列。出力。

#### (5) DIRECT

機能

行列  $A$  と  $B$  の直積  $C$  を計算する。整合寸法 KA, LA, KB, LB の指定により任意の大きさの行列に対応できる。

呼び出し型式

CALL DIRECT (A, MA, NA, B, MB, NB, C, MC, NC, KA, LA, KB, LB)

引数の説明

A: 行列  $A$ 。KA  $\times$  LA の実数型二次元配列。入力。

B: 行列  $B$ 。KB  $\times$  LB の実数型二次元配列。入力。

C: 行列  $C$ 。(KA  $\times$  KB)  $\times$  (LA  $\times$  LB) の実数型 2次元配列。出力。

MA, NA: 行列  $A$  の行数と列数。整数型。入力。

MB, NB: 行列  $B$  の行数と列数。整数型。入力。

MC, NC: 行列  $C$  の行数と列数。整数型。出力。

KA, LA: 配列 A の整合寸法。整数型。入力。

KB, LB: 配列 B の整合寸法。整数型。入力。

#### (6) HIST

機能

$N$  本の鎖の末端間距離  $R_n$  から,  $W(R) \sim R$  のヒ

ストグラムをプロットする。ヒストグラムの区間は  $R_n$  から自動的に決定する。また各区分幅は, サブルーチンの引数にしておく。整合寸法 M を変えることにより, 任意の鎖数に対応することができる。

呼び出し型式

CALL HIST (R, MOP, NOC, IDR, M)

引数の説明

R:  $R_n$  実数型一次元配列。入力。

MDP: 重合度。整数型。入力。

NOC: 鎖の本数  $N$ 。整数型。入力。

IDR: 区分幅。整数型。入力。

M: R の整合寸法。整数型。入力。

#### (7) INTRND (INT)

機能

グリコシド結合の二面角 ( $\phi$ ,  $\psi$ ) の存在確率のデータを変換し, 存在確率に対応した分布を有する  $\phi$  と  $\psi$  を生成する。整合寸法 K の指定により, 任意の大きさのエネルギー地図に対応できる。

サブルーチン INT は, INTRND を振動鎖の生成に対応できるように改良したものである。改良点は以下の通りである。引数 P を確率  $P_i$  に専用の変数とした。引数 PP を  $\phi$  と  $\psi$  の生成に用いる数値線上の境界値用の変数とした。

呼び出し型式

CALL INTRND (IX, IY, ID, N, P, PP, IPHI, IPSI, NOD, WRT, K)

引数の説明

IX, IY: それぞれ  $\phi$ ,  $\psi$  の最小値 (初期値)  $\phi_0$ ,  $\psi_0$ 。整数型。入力。

ID:  $\phi$ ,  $\psi$  の計算間隔。整数型。入力。

N: 存在確率のデータの個数の平方根。つまり, データを計算した  $\phi$  及び  $\psi$  の個数。整数型。入力。

P: 入力時,  $\phi_i$ ,  $\psi_i$  の存在確率  $P_i$ 。出力時,  $\phi_i$ ,  $\psi_i$  を生成するために用いる変数 ( $\sum P_i$ )。実数型一次元配列。入出力。

PP:  $\phi_i$ ,  $\psi_i$  の存在確率  $P_i$ 。実数型一次元配列。出力。

IPHI, IPSI:  $P_i$  ( $\neq 0$ ) に対応する二面角  $\phi_i$ ,  $\psi_i$ 。整数型一次元配列。出力。

NOD:  $P_i$  ( $\neq 0$ ), 及び  $\sum P_i$  の個数。整数型。出力。

WRT: 書き出し制御用の変数。1.0 の時 IPHI, IPSI, PP を書き出す。0.0 の時は書き出さない。実数型。入力。

K: P, IPHI, IPSI, PP の整合寸法。整数型。入力。

## (8) INVERS

## 機能

3×3の行列  $A$  の逆行列  $A^{-1}$  を計算する。

## 呼び出し型式

CALL INVERS (A, AIN, T)

## 引数の説明

A : 逆行列がほしい行列  $A$ 。A (3, 3) なる実数型二次元配列。入力。

AIN :  $A$  の逆行列。AIN (3, 3) なる実数型二次元配列。出力。

T :  $A$  の行列式。実数型。入力。

## (9) MSETED

## 機能

平均変換マトリックス  $\langle T \rangle$  より末端間距離の二乗平均  $\langle R^2 \rangle$  と特性比  $C_x$  を計算し出力する。特性比については、重合度を0以下あるいは10万以上にすると  $C_\infty$  を計算するように作成されている。また、このサブルーチンは、サブルーチン ADD, SUBTRA, INVERS, PRODAC を使用している。

## 呼び出し型式

CALL MSETED (ATK, MDP)

## 引数の説明

ATK : 平均変換マトリックス  $\langle T \rangle$ 。ATK (3, 3) なる実数型二次元配列。入力。

MDP : 鎖の重合度。実数型。入力。

## (10) PERSIS

## 機能

平均変換マトリックス  $\langle T \rangle$  より持続長  $a$ 、及び持続ベクトル  $\alpha$  を計算し出力する。このサブルーチンは、サブルーチン PRODAC を使用している。

## 呼び出し型式

CALL PERSIS (ATK, VLEN, PSIS, VEC)

## 引数の説明

ATK : 平均変換マトリックス  $\langle T \rangle$ 。ATK (3, 3) なる実数型二次元配列。入力。

VLEN : 仮想ボンド長 (Å)。実数型。入力。

PSIS : 持続長  $a$ 。実数型。出力。

VEC : 終了条件値。重合度による持続長  $a$  の変化が VEC 以下になった時の重合度の持続長  $a$  を真の持続長とみなす。実数型。入力。

## (11) PRODAC

## 機能

行列の積を計算するサブルーチン。

## 呼び出し型式

CALL PRODAC (A, B, PROAB, IC, IL, IS)

## 引数の説明

A, B : 積をもとめようとしている二つの行列  $A$ ,  $B$  の変数名。実数型二次元配列。入力。

PROAB : 行列  $A$ ,  $B$  の積。実数型二次元配列。入力。

IC : 行列  $A$ ,  $A$  と  $B$  の積の行数。整数型。入力。

IL : 行列  $B$ ,  $A$  と  $B$  の積の列数。整数型。入力。

IS : 行列  $A$  の列数。且つ行列  $B$  の行数。整数型。入力。

## (12) RNDGEN (RGEN)

## 機能

サブルーチン INTRND により初期化したデータを使用してグリコシド結合の二面角 ( $\phi$ ,  $\psi$ ) の存在確率に対応した分布を有する  $\phi$  と  $\psi$  をモンテカルロ法により生成する。 $\phi$ ,  $\psi$  の生成に際しては、二分検索法を使用した。発生できる  $\phi$  と  $\psi$  の数は100,000個までである。このサブルーチンでは、SSL II より RANU2 を呼んでいる。

サブルーチン RGEN は、RNDGEN を摂動鎖の生成に対応できるように改良したものである。改良点は、以下の通りである。変数 P を変数 PP に変更する。生成した  $\phi$ ,  $\psi$  の格納されている変数 IPHI, IPSI の配列要素を IPNT に格納する。IPNT は出力用の引数とする。

## 呼び出し型式

CALL RNDGEN (P, IPHI, IPSI, ICNT, NOD, MDP, ISEED, MPHI, MPSI, WRT, K, L)

## 引数の説明

P : INTRND より得られる  $\phi_i$ ,  $\psi_i$  を生成するために用いる ( $\sum P_i$ )。実数型一次元配列。入力。

IPHI, IPSI :  $P_i$  に対応する二面角  $\phi_i$ ,  $\psi_i$ 。整数型一次元配列。入力。

ICNT : 入力時、 $\phi_i$  と  $\psi_i$  の発生頻度の初期値。普通0でよい。出力時、 $\phi_i$  と  $\psi_i$  の発生頻度。整数型一次元配列。入出力。

NOD : 一様乱数の発生区間 (0, 1) の分割点の個数 (1を含む)。整数型。入力。

MDP : 重合度 (乱数の個数)。整数型。入力。

ISEED : 入力時、一様乱数を発生するために用いる初期値。出力時、次の本サブルーチンの呼び出しのための初期値。整数型。入出力。

MPHI, MPSI : 生成した二面角  $\phi_i$  と  $\psi_i$ 。整数型一次元配列。出力。

WRT : 書き出し制御変数。1.0の時出力し、0.0



の時出力しない。実数型。入力。

K : P, IPHI, IPSI, ICNT の整合寸法。

L : MPHI, MPSI の整合寸法。10,000>L>MDP

### (13) STAR

機能

$A=[a_{ij}]$ ,  $B=[b_{ij}]$  のとき  $A \star B=[a_{ij} b_{ij}]$  を計算する。

呼び出し型式

CALL STAR (X, Y, Z, M, N, K, L)

引数の説明

X : 行列  $A$ 。実数型二次元配列。入力。

Y : 行列  $B$ 。実数型二次元配列。入力。

Z : 行列  $[a_{ij} b_{ij}]$ 。実数型二次元配列。出力。

M, N : 行列の行数と列数。

K, L : 変数 X, Y, Z の整合寸法。

### (14) SUBTRA

機能

行列の差を計算する。

呼び出し型式

CALL SUBTRA (A, B, SUBAB, IC, IL)

引数の説明

A, B : 差をもとめようとしている二つの行列  $A$ ,  $B$  の変数名。実数型二次元配列。入力。

SUBAB : 行列  $A$ ,  $B$  の差。  $A-B$ 。実数型二次元配列。出力。

IC, IL : 行列  $A$ ,  $B$  の行数と列数。整数型。入力。

- (15) その他にサブルーチン RNDGEN により発生した  $(\phi_i, \psi_i)$  の頻度の  $\chi^2$  検定のためのサブルーチン CHI も用意されている。

## V アミロース鎖モデルのグラフィック出力用プログラム GRAPHIC, FORT77(GSP3D), (GSPLIB)

使用言語 : FORTRAN (自由型式)

### V-1 概要

本プログラムは、グラフィック・ディスプレイ装置を使って、アミロース鎖を視覚的に表示するためのものである。TAC. FORT77 (MAIN) により算出されるアミロース鎖の仮想ボンドの座標値を入力し座標をプロットする。

京大計算機センターには、グラフィックサブルーチンパッケージ (以下 GSP と略す)<sup>15,16)</sup> と分子構造表示プログラム NAMOD<sup>17)</sup> が用意されており、図型出

力にはこれらのライブラリプログラムを使用した。GRAPHIC. FORT77 (GSP3D) は3次元ディスプレイ装置(F9532A)を、GRAPHIC, FORT77(GSPLIB) は2次元ディスプレイ装置 (F6233L) をそれぞれ使用した。

### V-2 フローチャート

プログラムのフローチャートを Fig. 15 に示す。図より、プログラム中で4つの GDR を作成していることがわかる。内容は次の通りである。

NOGRDR : 画面に表示する鎖の番号の入力を指示する GDR (Fig. 16)。

ANGGDR : 鎖の回転操作を行う時その回転角の入力を指示する GDR (Fig. 17)。

MSGGDR : ファンクションキー(以下 FK と略す)のメニューを表示する GDR (Fig. 18)。

IGDR : 3つの LG から構成され、LG1 はアミロース鎖モデル、LG2 は回転角の表示、LG3 はスケール (10Å) をそれぞれ表示する (Fig. 19)。

次に、フローチャートに付した番号に従って内容を説明する。

1. アミロース鎖の座標値を読み込む。10本以内の鎖のデータが格納される。
- 2, 3. 定数の定義, GSP の初期化及び表示画面のモード指定等を行う。
4. NOCGDR, ANGGDR, MSGGDR を作成する。
5. 第1のアテンションマスクを開設する。これによってキーボード (数値入力用) と FK=1, 2, 3, 4, 5, 32 が有効になる。
6. NOCGDR の表示
7. 10本の鎖の内、表示する鎖の番号 ICHA をキーボードより入力する。
8. 表示する鎖について、NAMOD 用のデータを作成する。
9. 鎖が画面の真中に位置するように、鎖中心を検索する。
10. 表示画面のスクリーンウィンドウとバーチャルウィンドウを設定する。
11. IGDR を作成する。
12. MSGGDR の表示。
13. FK=1の時、MSGGDR を表示する。
14. FK=2の時、表示画面を NLP に出力する。
15. FK=3の時、IGDR を表示する。
16. FK=4の時、IGDR からグラフィックデータを削除し、6にもどる。
17. FK=5の時、第2のアテンションマスクを開設

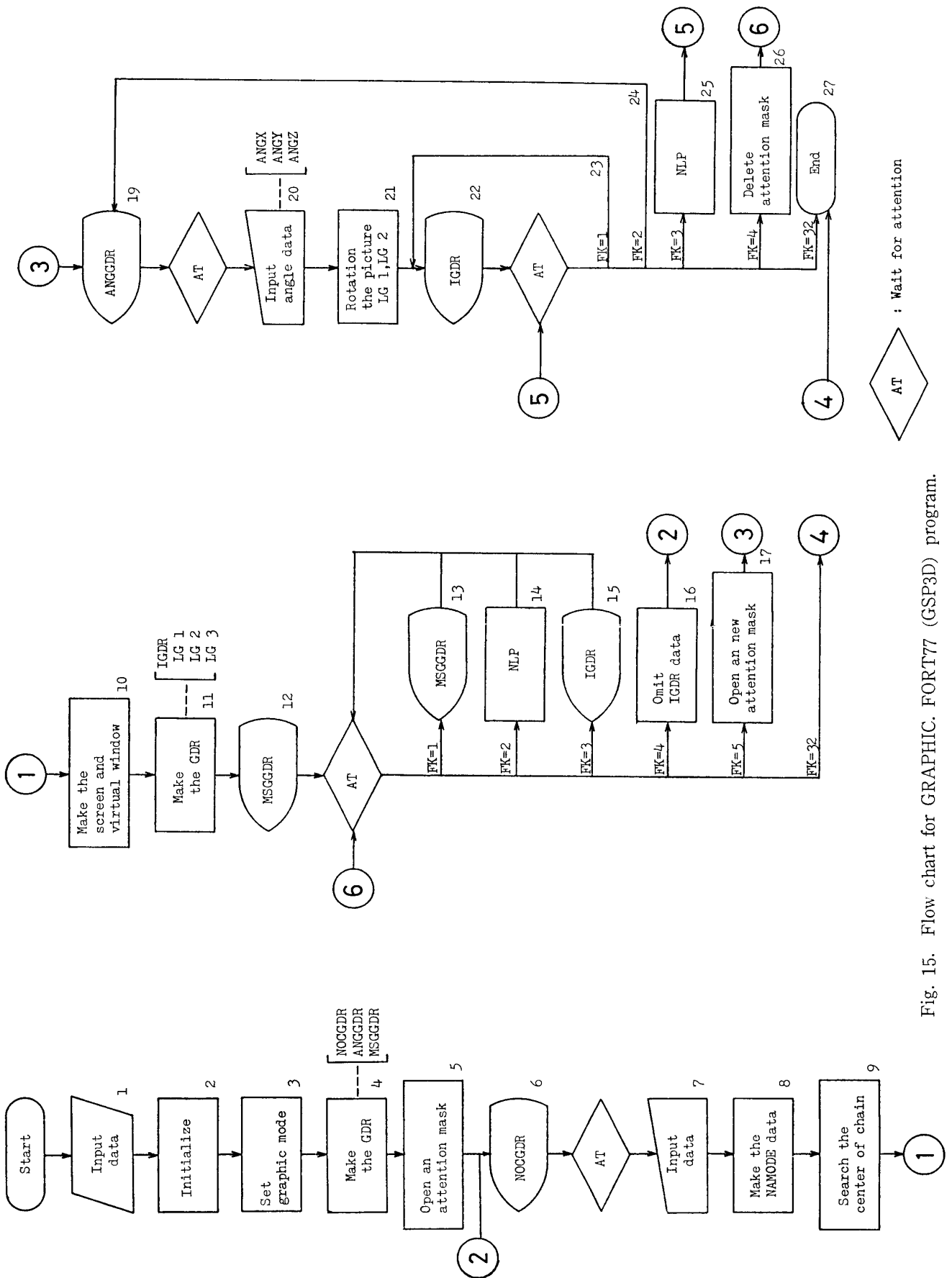


Fig. 15. Flow chart for GRAPHIC. FORT77 (GSP3D) program.

\*\*\* PLEASE TYPE IN \*\*\*

NO=

(EX,1) NO=03 -----OK

(EX,2) NO=3 -----MISTAKE

Fig. 16. NOCGDR

\*\*\* MENU OF THE SUBROUTINE

\*\*\* MENU OF THE SUBROUTINE \*\*\*

---FK=1 DISPLAY THE MOLECULAR MODEL ---

---FK=2 CHANGE THE X,Y,Z-ANGLES ---

---FK=3 OUTPUT THE NLP ---

---FK=4 END THE SUBROUTINE ---

\*\*\* PLEASE TYPE IN X,Y,Z-ANGLES \*\*\*

ANGLE-X =

ANGLE-Y =

ANGLE-Z =

(EX.1) ANGLE-Y =090

(EX.2) ANGLE-Y =90 ----- MISTAKE !!

Fig. 17. ANGGDR

FK=1 DISPLAY THE MENU

FK=2 OUTPUT NLP

FK=3 DISPLAY THE MOLECULAR MODEL

FK=4 CHANGE THE CHAIN NUMBER

FK=5 INPUT X,Y,Z-ANGLES

FK=23 END

Fig. 18. MSGGDR

する。これによって FK は更新され、キーボード以外の FK=1, 2, 3, 4, 32 が回転操作のための FK として新たに有効となる。

18. FK=32 の時、プログラムは終了する。

19. ANGGDR の表示。

20. キーボードより回転角 ANGX, ANGY, ANGZ を入力する。ANGX, ANGY, ANGZ は、それぞれ x 軸, y 軸, z 軸まわりの回転角である。

21. 表示図形の回転を行なう。

22. 回転したアミロース鎖 (IGDR) を表示する。

23. FK=1 の時、さらに表示図形を ANGX, ANGY, ANGZ だけ回転し、画面に表示する。

24. FK=2 の時、ANGGDR を表示し、回転角の変更が可能になる。

25. FK=3 の時、表示画面を NLP に出力する。

26. FK=4 の時、第2のアテンションマスクが消滅し、第1のアテンションマスクが有効になる。

27. FK=32の時、プログラムは終了する。

具体的な使用方法、あるいは、GSP, NAMOD の内容については、各種マニュアルを参照されたい<sup>15-17)</sup>。

## VI 摂動状態のアミロース鎖作成プログラム

### VI-1 概要

アミロース分子鎖中の隣接したグルコース残基間の相互作用に加えて、複数のグルコース残基を隔てた相互作用、すなわち遠距離相互作用を排除体積効果としてモデルに導入する。まず、アミロース分子鎖を構成するグルコース残基をその重心まわりの半径  $r$  の剛体球とみなし、分子鎖をこれらの剛体球が連結されたものとする。そして、これらの剛体球は互いに重って存在することができないと仮定する。球の半径  $r$  は、遠距離相互作用として考慮する斥力、すなわち排除体積効果の大きさを表わすパラメータであり、溶媒の種類や温度などの外的条件によって変化すると考えられる。

前章で定義した、グルコース残基に固定された座標系において、そのグルコース残基の重心の位置がベクトル  $\mathbf{g}$  で与えられているとすると、アミロース分子鎖を構成する任意のグルコース残基の重心の位置を示すベクトル  $\mathbf{G}_i$  は次式で与えられる。

$$\mathbf{G}_i = \mathbf{T}_1 \mathbf{T}_2 \cdots \mathbf{T}_{i-1} \mathbf{g} + \mathbf{T}_1 \mathbf{T}_2 \cdots \mathbf{T}_{i-2} \mathbf{l} + \cdots + \mathbf{T}_1 \mathbf{T}_2 \mathbf{l} + \mathbf{T}_1 \mathbf{l} + \mathbf{l} \quad (31)$$

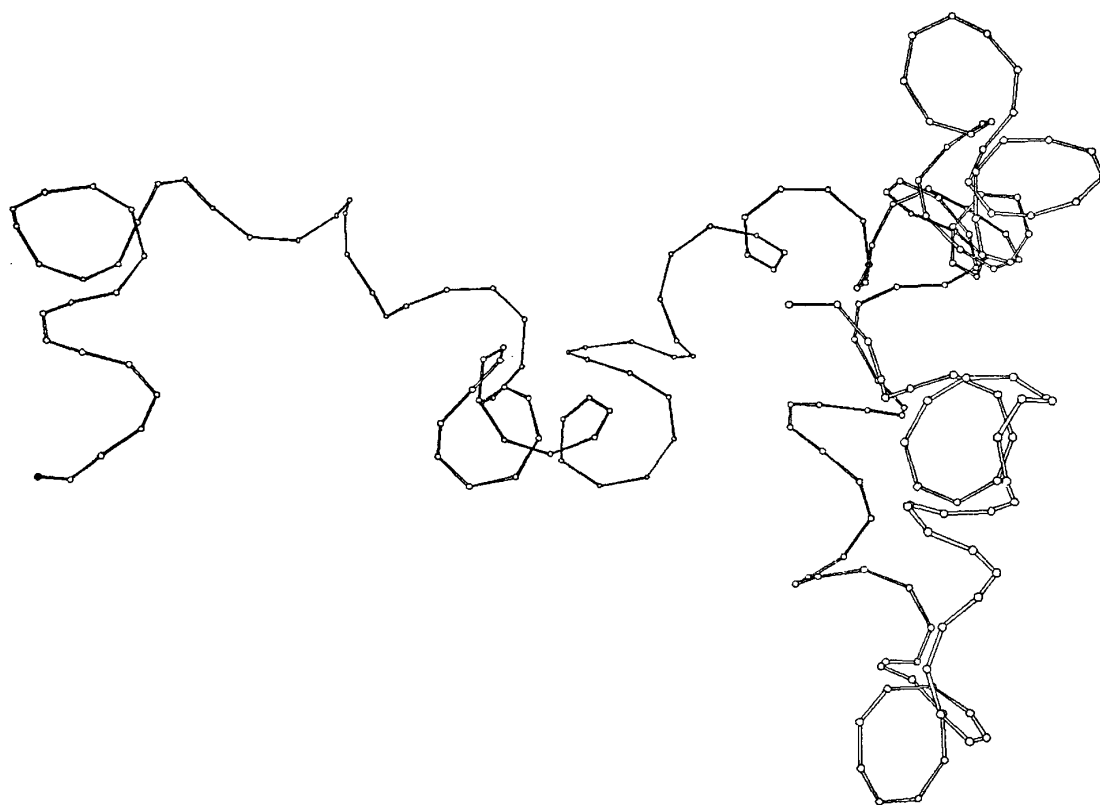
ここで、 $\mathbf{T}_k$  ( $k=1, \dots, i-1$ ) は一様乱数を用いた一連の処理によって得られる座標変換マトリックスであり、また、 $\mathbf{l}$  は仮想ボンドを表わすベクトルである。

(31)式は、(19)式と同様にマトリックス演算の形に書き改めることができる。

$$\mathbf{G}_i = [\mathbf{E} \ \mathbf{0}] \begin{bmatrix} i-1 \\ \prod_{k=1} A_k \end{bmatrix} \begin{bmatrix} \mathbf{g} \\ \mathbf{l} \end{bmatrix} \quad (32)$$

$$A_k = \begin{bmatrix} \mathbf{T}_k & \mathbf{l} \\ \mathbf{0} & 1 \end{bmatrix} \quad (33)$$

ここで、 $\mathbf{E}$  は  $3 \times 3$  の単位マトリックス、 $\mathbf{0}$  は  $0$  ベクトルである。(32)式の演算を  $i=1$  から  $i=m$  まで繰り返すと、重合度  $m$  のアミロース分子鎖のすべてのグルコース残基の重心の座標が計算される。



10.0 ANGSTROM

TANGX= 0.0      TANGY= 0.0      TANGZ= 0.0      NO=  
ANGX= 0.0      ANGY= 0.0      ANGZ= 0.0

Fig. 19. An example of perspective drawing of 200-residues Monte Carlo amylosic chain. Circles represent glycosidic oxygens, and lines are virtual bonds. IGDR.

生成したアミロース鎖中に重なりが生じているか否かは、隣接したグルコース残基を除くすべてのグルコース残基の組み合わせについて、それらの重心間の距離を計算することにより調べることができる。グルコース残基  $i$  の重心とグルコース残基  $j$  の重心の距離  $d_{ij}$  は次式で与えられる。

$$d_{ij} = |\mathbf{G}_i - \mathbf{G}_j| \quad (34)$$

ここで、 $\mathbf{G}_i$ 、 $\mathbf{G}_j$  は、それぞれグルコース残基  $i$  と  $j$  の重心の位置を表わすベクトルである。この場合に  $i > j + 1$  について  $d_{ij} > r$  が満たされていれば、そ

のアミロース分子鎖のサンプルには重なりが生じていないことになる。

モンテカルロ法を用いて、重合度  $m$  の摂動鎖のサンプルを生成する場合の最も簡単なアルゴリズムを以下に示す。

- ①  $x=1$
- ② 第  $x$  番目のグルコース残基の重心の座標を乱数を用いた一連の処理によって決定する。(82, 83 式参照)
- ③ グルコース残基(剛体球)に重なりが生じたら

①にもどる。(84式参照)

④  $x=m$  ならば, 生成した摂動鎖の座標データをサンプルとして登録し, ①にもどる。

⑤ それ以外の場合には  $x$  を 1 増して②にもどる。

このアルゴリズムによって重合度  $m$  の摂動鎖のサンプルが得られる確率  $P_m$  は, 次式で与えられることが報告されている<sup>10,11)</sup>。

$$P_m = \exp(-\lambda m) \quad (35)$$

ここで,  $\lambda$  はサンプル損失定数 (sample attrition constant) と呼ばれるパラメータであり, 今回のアミロース鎖のモデルでは, 隣接残基間の相互作用を表わすエネルギー地図と遠距離相互作用の強さを表わす球の半径  $r$  に依存する。

85式から明らかのように,  $P_m$  は  $m$  の増加とともに指数関数的に減少し, 重合度の大きな摂動鎖のサンプルの生成は非常に困難である。そこでステップ③で①にもどるかわりに②にもどる方法が考えられるが, この方法では, 得られる摂動鎖のサンプルが統計的に偏ってしまうことになる。そこで, 重合度の大きなサンプルを統計学上公平に, かつ効率的に生成するために, Wall らのサンプル倍増法<sup>1)</sup>を用いてサンプル生成の効率化を図った。サンプル倍増法では, 重合度の大きなサンプルを一度に生成することは行わず, 重合度が  $S$  だけ増加するごとに, それまでに生成したサンプルが  $P$  本得られたと考えて, 生成したサンプルを有効に利用する。このとき, 統計的に公平なサンプルを, できる限り効率的に生成するために, パラメータ  $S$  と  $P$  に最適の整数値を与えなければならない。 $S$  と  $P$  の最適値は, 以下の手順で決定した。

- ① 単純モンテカルロ法によって, 重合度  $m$  のサンプルが生成される確率  $P_A$  を測定する。
- ② 測定結果に85式をあてはめてパラメータ  $\lambda$  を決定する。あてはめは, 最小二乗法によって行えばよい。
- ③  $P$  と  $S$  を,  $P \approx \exp(\lambda S)$  を満たす適当な整数として決定する。通常は  $P < \exp(\lambda S)$  であるのが望ましい。

## VI-2 フローチャート

### VI-2-1 TAC. FORT77 (PA)

使用言語: FORTRAN (自由型式)

単純モンテカルロ法によってサンプルを生成し, 重なりのないサンプルの得られる確率  $P_A$  を計算して出

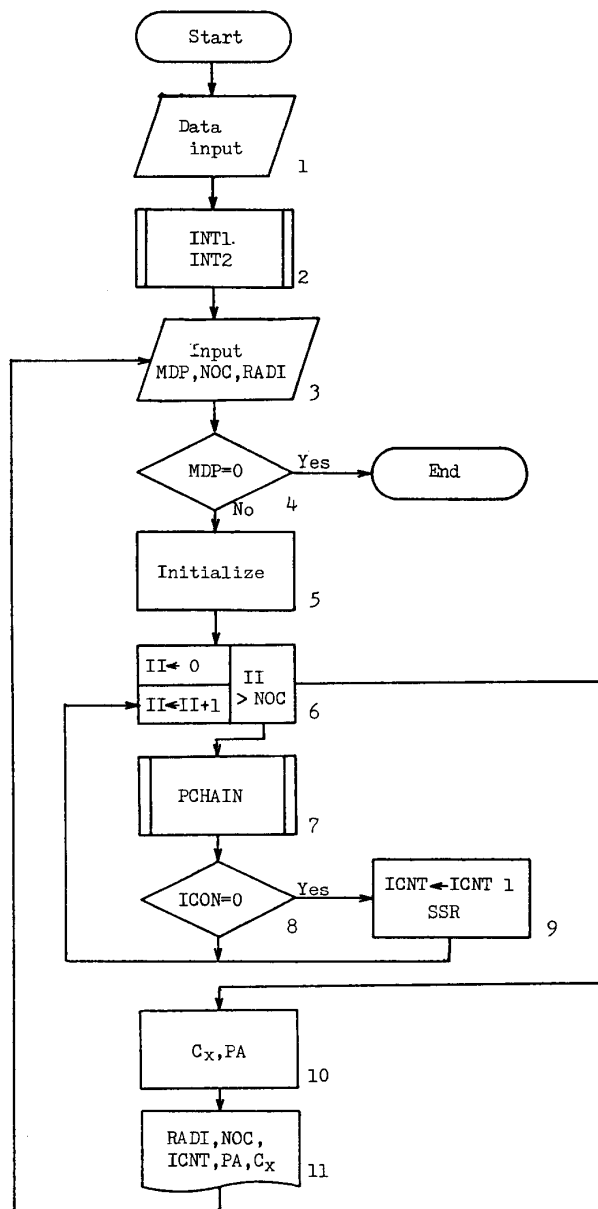


Fig. 20. Flow chart for TAC. FORT77 (PA) program.

力する。以下フローチャート (Fig. 20) に付した番号に従ってその内容を説明する。

1. グルコース残基の重心の座標, 座標変換マトリックス  $T_k$  のパラメータ, 仮想ボンド長, グリコシド結合の二面角 ( $\phi, \psi$ ) とそれに対応する存在確率  $P(\phi, \psi)$  を入力する。
2. サブルーチン INT1, INT2 を呼んで入力データを初期化する。
3. 重合度 (MDP), 鎖の本数 (NOC), 剛体球の半径 (RADI) を入力する。
4.  $MDP=0$  の時, プログラムを終了する。
5. カウンター (ICNT), 二乗半径の総和 (SSR) を初期化する。

6. (7~9) を NOC 回繰り返す。
7. サブルーチン PCHAIN を呼んでサンプル生成し、重なりのない場合には  $ICON=0$ , 重なりのある場合には  $ICON=3000$  を得る。
- 8, 9.  $ICON=0$  の時, ICNT にカウントし, SSR を計算する。それ以外の時は, 計算せず 6 にもどる。
10. 次式により  $C_x$ ,  $P_A$  を計算する。

$$C_x = SSP/MDP/VLEN^{**2}/ICNT \quad (36)$$

$$P_A = ICNT/NOC \quad (37)$$

11.  $RADI$ ,  $NOC$ ,  $ICNT$ ,  $PA$ ,  $C_x$  をプリンタに出力する。

測定結果  $P_A$  に, (37) 式をあてはめて, パラメータ  $\lambda$  を決定する。さらに  $P \approx \exp(\lambda S)$  を満たす整数  $P$  と  $S$  を決定する。

#### VI-2-2 TAC, FORT77 (PCDIM)

使用言語: FORTRAN (自由型式)

サンプル倍増法によってアミロース分子の摂動鎖を生成し, 特性比  $C_x$ , 持続長  $a$ , 動径分布関数等を計算して出力する。 $C_x$  と  $a$  は III 章と同様に次式で計算される。

$$C_x = \langle R_x^2 \rangle / x l^2$$

$$= \langle X_x^2 + Y_x^2 + Z_x^2 \rangle / x l^2 \quad (38)$$

$$a = \lim_{x \rightarrow \infty} \sqrt{\langle X_x \rangle^2 + \langle Y_x \rangle^2 + \langle Z_x \rangle^2} \quad (39)$$

ここで,  $R_x$  は重合度  $x$  の分子鎖の末端間距離,  $X_x$ ,  $Y_x$ ,  $Z_x$  は重合度  $x$  の分子鎖における末端間ベクトルの  $X$ ,  $Y$ ,  $Z$  成分であり,  $\langle \rangle$  は鎖のとり得るあらゆるコンホメーションについての平均を表わす。

以下フローチャート (Fig. 21) に付した番号に従って計算内容を説明する。

1. グルコース残基の重心の座標, 座標変換マトリックス  $T_k$  のパラメータ, 仮想ボンド長, グリコシド結合の二面角 ( $\phi$ ,  $\psi$ ) と, それに対応する存在確率  $P(\phi, \psi)$  を入力する。
2. サブルーチン INT1, INT2 を呼んで入力データを初期化する。
3. サブルーチン ENRICH の為のパラメータ ( $S_0$ ,  $S$ ,  $P$ ,  $JLIM$ ,  $RADI$ ,  $NOC$ ) と, サブルーチン HIST の為のパラメータ ( $ISTEP$ ) を入力する。
4.  $S_0=0$  かつ  $S=0$  の時, プログラムを終了する。
5. サブルーチン ENRICH を呼び, サンプル倍増法を使って摂動鎖を作成する。

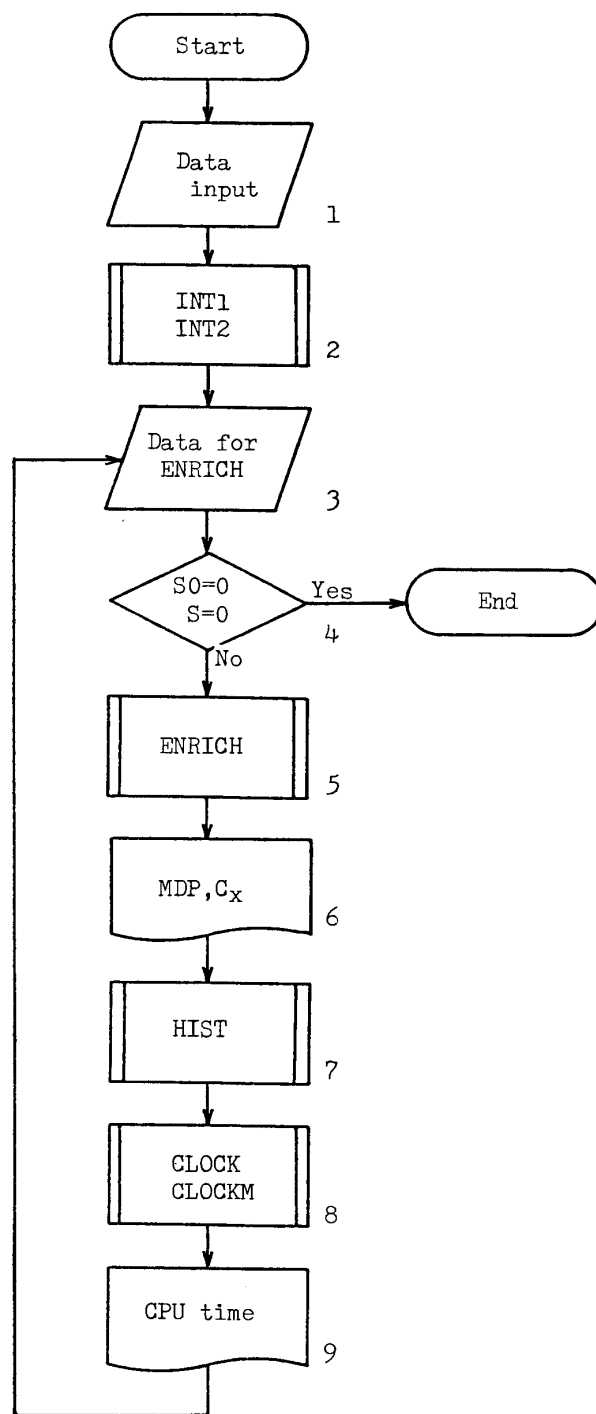


Fig. 21. Flow chart for TAC, FORT77 (PCDIM) program.

6. 摂動鎖の重合度 ( $MDP$ ) と特性比 ( $C_x$ ) を出力する。
7. サブルーチン HIST を呼び, 動径分布関数  $W(R)$  をヒストグラムで出力する。
- 8, 9. CPU 占有時間を測定し出力する。その後 3 へもどる。

#### VI-2-3 TAC, FORT77 (SELECT)

使用言語：FORTRAN（自由型式）

グラフィック表示のための摂動鎖の座標データを生成する。サンプル倍増法によって鎖を生成し、与えられた範囲内の二乗末端間距離を持つサンプルを選択して、そのデータをファイルに格納する。以下フローチャート (Fig. 22) に付した番号に従って計算内容を説明する。

1. サンプル倍増法のためのパラメータ ( $S_0$ ,  $S$ ,  $P$ ,  $JLIM$ ,  $RADI$ ,  $NOC$ ), 選択するサンプルの本数 ( $NOG$ ), 二乗末端間距離の範囲を入力する。次に, グルコース残基の重心の座標, 座標変換マトリックス  $T_k$  のパラメータ, 仮想ボンド長, グリコシド結合の二面角 ( $\phi$ ,  $\psi$ ) とそれに対応する存在確率  $P$  ( $\phi$ ,  $\psi$ ) を入力する。
2. サブルーチン INT1, INT2 を呼んで入力データを初期化する。

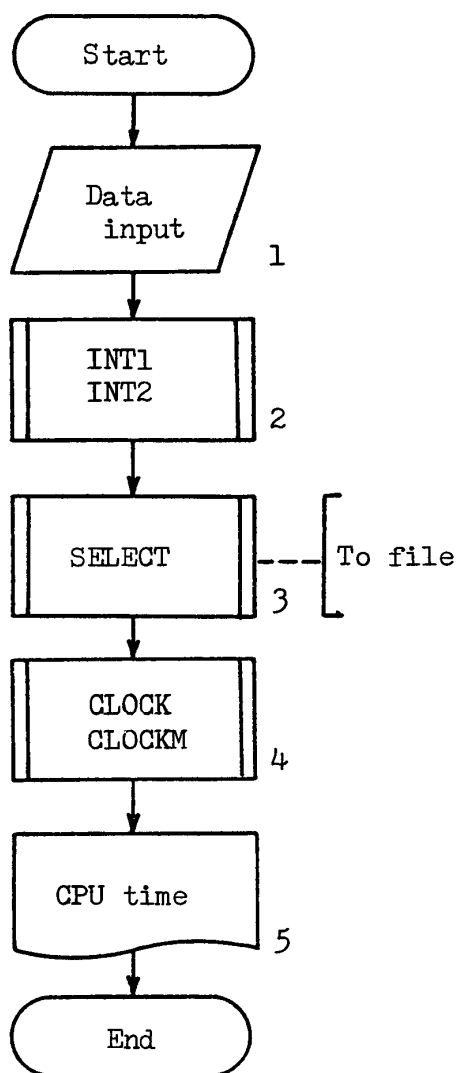


Fig. 22. Flow chart for TAC. FORT77 (SELECT) program.

3. サブルーチン SELECT を呼んで、与えた範囲の二乗末端間距離を持つサンプルを生成し、その座標データをファイルに格納する。
- 4, 5. CPU 占有時間を測定し出力する。

VI-2-4 TAC, FORT77 (TIME)

使用言語：FORTRAN（自由型式）

NOC 本のサンプルの生成に要する CPU 占有時間を測定する。フローチャートは, Fig. 22 中のサブルーチン SELECT を ENRITCH に変えたものに等しく, ここでは, 計算内容の解説は省略する。

VI-3 サブルーチンの説明

メインプログラムとサブルーチンの関係を Fig. 23 に示す。VI-3 に記されていない新しいサブルーチンについて以下説明する。

(1) ENRICH

機能

サブルーチン INT1, INT2 によって初期化されたデータを用い, サンプル倍増法によって重ならないサンプルを指定された数だけ生成し, 生成したサンプルの特性比, 持続長, 慣性半径の二乗平均などを求める。サンプル倍増法の実行に必要なパラメータ  $S$  と  $P$  のいずれか一方, または双方に 0 を指定すれば, 単純モンテカルロ法によって重ならないサンプルを生成する。サンプルのステップ数 (重合度), 確率分布関の大きさ, サンプルの数は任意に指定することができる。

呼び出し型式

CALL ENRICH (PP, IPHI, IPSI, NOD, THE, ETA, XI, OME, VLEN, RADI, G, NOC, NS0, NS, NP, JLIM, R, CX, S2, R2, COOD, NOF, MDP, B, NOT, INCT, OCOX, OCOY, OCOZ, GCOX, GCOY, GCOZ, NSUB, KA, LA, MA, WRT)

引数の説明

PP, IPHI, IPSI, NOD: サブルーチン INT2 に同じ。ただしここでは入力用の変数である。

THE, ETA, XI, OME: 座標変換パラメータ  $\theta$ ,  $\eta$ ,  $\xi$ ,  $\omega$ 。実数型。入力。

VLEN: 仮想ボンド長 (Å)。実数型。入力。

RADI: 剛体球の半径 (Å)。実数型。入力。

G: グルコースの重心ベクトル  $g$ 。G(3)なる実数型一次元配列。入力。

NOC: 必要なサンプル数。整数型。入力。

NS0, NS, NP, JLIM: サンプル倍増法に必要なパラメータ  $S_0$ ,  $S$ ,  $P$ ,  $Jlimit$ 。ただし,  $2 \leq S_0$ ,

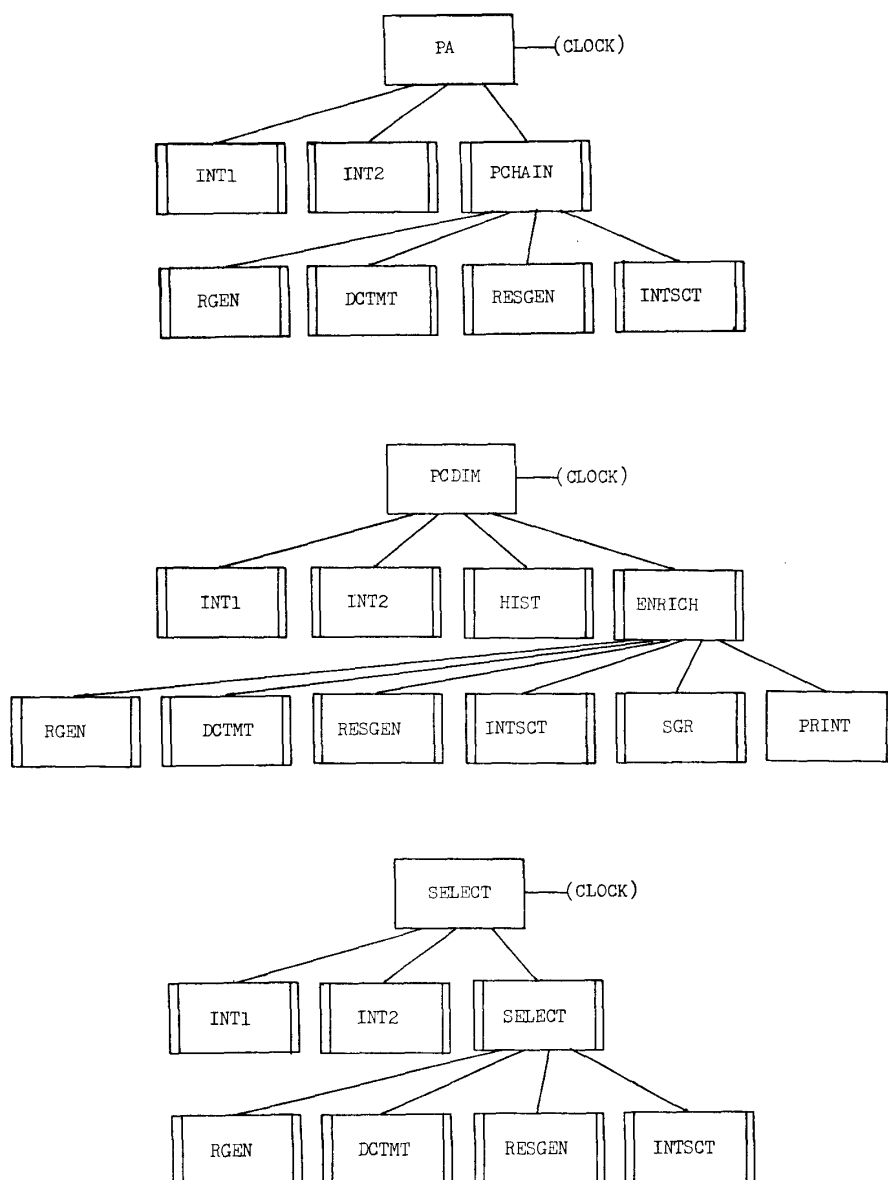


Fig. 23. Relations between main programs (PA, PCDIM, SELECT) and subroutine programs.

$2 \leq Jlimit$  とする。整数型。入力。

R: 各サンプルの末端間距離。R (LA) なる実数型一次元配列。出力。

CX: 特性比  $C_x$ 。実数型。出力。

S2: 慣性半径の二乗平均  $\langle S^2 \rangle$ 。実数型。出力。

R2: 末端間距離の二乗平均  $\langle R^2 \rangle$ 。実数型。出力。

COOD: 各残基の平均の位置を示す座標値。(bridge oxygen で表わす。) COOD (3, MA) なる実数型二次元配列。出力。

NOF: 重なるの生じたサンプル数。整数型。出力。

MDP: サンプルのステップ数(重合度)。整数型。出力。

B: 作業領域。B(4, 4, JLIM-1) なる実数型三次元配列。入力。

NOT: 作業領域。NOT(JLIM-1) なる実数型一次元配列。入力。

ICNT: 作業領域。ICNT (KA) なる整数型一次元配列。入力。

OCOX, OCOY, OCOZ: 作業領域。OCOX (0: MA), OCOY (0: MA), OCOZ (0: MA) なる実数型一次元配列。入力。

GCOX, GCOY, GCOZ: 作業領域。GCOX (MA), GCOY (MA), GCOZ (MA) なる実数型一次元配列。

NSUB: 作業領域。NSUB (MA) なる整数型一



次元配列。入力。

KA : PP, IPHI, IPSI の整合寸法。KA $\geq$ NOD。整数型。入力。

LA : R の整合寸法。LA $\geq$ NOC。整数型。入力。

MA : COOD, OCOX, OCOY, OCOZ, GCOX, GCOY, GCOZ の整合寸法。MA $\geq$ NS0+NS(JLIM-1)。整数型。入力。

WRT : 書き出し制御変数。0.0の時、書き出さない。1.0の時、最終結果のみ書き出す。2.0の時、重なったサンプルについての情報を書き出す。3.0の時、重なったサンプルについての情報とサンプルの座標値を書き出す。

## (2) INTSCT

機能

与えられた1個の座標と他の  $n$  個の座標の距離  $d_i (i=1 \cdots n)$  を計算し、 $1 \leq i \leq n$  について  $d_i \geq r$  が成立する場合には変数 ICND に1を、それ以外の場合には2を代入し復帰する。サブルーチン SELECT, ENRICH, PCHAIN では本サブルーチンを利用してサンプルに重なりが生じたか否かを判定している。

呼び出し型式

CALL INTSCT (COX, COY, COZ, X, Y, Z, R, ICND, N, NSUB, NOO)

引数の説明

COX, COY, COZ : 座標値 ( $x_i, y_i, z_i$ )。COX (N), COY (N), COZ (N) なる実数型一次元配列。入力。

X, Y, Z : 座標値  $x, y, z$ 。実数型。入力。

R : 剛体球の半径  $r$ 。実数型。入力。

ICND : コンディションコード。整数型。出力。

N : COX, COY, COZ の整合寸法。整数型。入力。

NSUB :  $d_i < r$  になった残基の座標の格納されている配列の添字。整数型一次元配列。出力。

NOO :  $d_i < r$  になった残基の個数。整数型。出力。

## (3) INT1

機能

グリコシド結合の二面角 ( $\phi, \psi$ ) の存在確率に対して、 $\phi$  と  $\psi$  の数値をデータとして追加する。

呼び出し型式

CALL INT1 (IX, IY, ID, N, IPHI, IPSI, K)

引数の説明

IX, IY : それぞれ  $\phi, \psi$  の最小値  $\phi_0, \psi_0$ 。整数型。入力。

ID :  $\phi, \psi$  の計算間隔。整数型。入力。

N : 存在確率のデータの個数の平方根。つまり、データを計算した  $\phi$  及び  $\psi$  の個数。整数型。入力。

IPHI, IPSI : 存在確率を計算した  $\phi$  値,  $\psi$  値。整数型一次元配列。出力。

K : IPHI, IPSI の整合寸法。入力。

## (4) INT2

機能

確率地図に対応した分布を持つ乱数を発生するために、区間 (0, 1) の境界値を生成し、同時に P, IPHI, IPSI に対して圧縮処理を行なう。(P $_i$  = 0 に対応する P, IPHI, IPSI を消去する)

呼び出し型式

CALL INT2 (P, PP, IPHI, IPSI, NOD, WRT, K, N)

引数の説明

P : 入力時は、( $\phi_i, \psi_i$ ) の存在確率 P $_i$ 。出力時は、圧縮処理後の P $_i$ 。実数型一次元配列。入出力。

PP : 区間(0, 1)の境界値。実数型一次元配列。出力。

IPHI, IPSI : 入力時は、P $_i$  に対応する  $\phi_i, \psi_i$ 。出力時は、圧縮処理後の P $_i$  に対応する  $\phi_i, \psi_i$ 。整数型一次元配列。入出力。

NOD : 圧縮処理後の P $_i$  の個数。整数型。出力。

WRT : 書き出し制御変数。1.0の時、IPHI, IPSI, PP を書き出す。0.0の時は書き出さない。実数型。入力。

K : P, PP, IPHI, IPSI の整合寸法。

N : 存在確率のデータの個数の平方根。つまり、データを計算した  $\phi$  及び  $\psi$  の個数。整数型。入力。

## (5) PCHAIN

機能

サブルーチン INT1, INT2 によって初期化されたデータを用い、単純モンテカルロ法によってサンプルを一本生成する。生成過程において剛体球に重なりが生じたら処理を打ち切り、変数 ICND に 3,000 を与えて復帰する。重なりが生じなければ指定されたステップ数に達するまで処理を続行し、ICND に 0 を与えて復帰する。指定できるサンプルのステップ数は 1,000 までである。

呼び出し形式

CALL PCHAIN (PP, IPHI, IPSI, NOD, THE, ETA, XI, OME, VLEN, MDP, RADI, G, ISEED, OCOX, OCOY, OCOZ, GCOX, GCOY, GCOZ, ICON, WRT)

## 引数の説明

PP, IPHI, IPSI, NOD: サブルーチン INT1, INT2 に同じ。ただし, ここでは入力用の変数である。

THE, ETA, XI, OME, VLEN, MDP, RAD, G: サブルーチン ENRICH に同じ。入力。

ISEED: 入力時は, 乱数発生初期値。出力時は, 次の本サブルーチン呼び出しにおける初期値。入出力。

OCOX, OCOY, OCOZ: 各残基の O(1)の座標。OCOX (0:1000), OCOY (0:1000), OCOZ (0:1000) なる実数型一次元配列。出力。

GCOX, GCOY, GCOZ: 各残基の重心の座標。GCOX (1000), GCOY (1000) GCOZ (1000) なる実数型一次元配列。出力。

ICON: コンディションコード。0の時, 生成した鎖に重なりがないことを意味し, 3000の時, 重なりがあることを意味する。整数型。出力。

WRT: 書き出し制御用変数。0.0の時, 書き出さない。1.0の時, 重なりメッセージを書き出す。実数型。入力。

## (6) PRINT

## 機能

サンプルの座標値の出力用サブルーチン。

## 呼び出し型式

CALL PRINT (OCOX, OCOY, OCOZ, MDP, MA)

引数の説明 (サブルーチン ENRICH 参照)

OCOX, OCOY, OCOZ: 各残基の重心の座標。実数型一次元配列。入力。

MDP: 重合度。整数型。入力。

MA: 整合寸法。整数型。入力。

## (7) RESGEN

## 機能

サブルーチン DCTMT で得られたマトリックス  $T_k$  を用いて, 残基数  $(j-2)$  のサンプルを 残基数  $(j-1)$  に伸ばし,  $j$  番目の残基の重心と O(1)の座標を求める。

## 呼び出し型式

CALL RESGEN (A, AA, VLEN, G, GCO, OCO, AAA)

## 引数の説明

A:  $\prod_{i=1}^{j-2} A_i$ . A(4, 4)なる実数型二次元配列。入力。

AA:  $A_{i-1}$ . AA(4, 4) なる実数型二次元配列。入力。

VLEN: 仮想ボンド長 (Å)。実数型。入力。

G: 重心ベクトル  $g$ 。G(3) なる実数型一次元配列。入力。

GCO: 新たに生成した残基の重心の座標。GCO(3) なる実数型一次元配列。

OCO: 新たに生成した残基の O(1)の座標。OCO(3) なる実数型一次元配列。

AAA:  $\prod_{i=1}^{j-1} A_i$ . AAA(4, 4) なる二次元配列。出力。

## (8) SELECT

## 機能

サンプル倍増法によって摂動鎖のサンプルを生成し, 与えられた範囲内の二乗末端間距離を持つサンプルを選択して割り当てられたファイルに出力する。二乗末端間距離の範囲は  $R^2/xl^2$  として与えられるものとする。必要なサンプル数は任意に指定することができる。

## 呼び出し型式

CALL (PP, IPHI, IPSI, NOD, THE, ETA, XI, OME, VLEN, RAD, G, NOC, NOG, NS0, NS, NP, JLIM, RMIN, RMAX, R, MDP, B, NOT, ICNT, OCOX, OCOY, OCOZ, GCOX, GCOY, GCOZ, NSUB, KA, LA, MA)

## 引き数の説明

PP, IPHI, IPSI, NOD, THE, ETA, XI, OME, VLEN, RAD, G: サブルーチン ENRICH に同じ。入力。

NOC, NOG: 生成する重なりのないサンプルの数の上限と, ファイルに出力するサンプル数 (NOG  $\leq 10$ )。ファイルに出力したサンプルの総数が NOC になった場合に復帰する。整数型。入力。

NS0, NS, NP, JLIM: サブルーチン ENRICH に同じ。入力。

RMIN, RMAX: 出力するサンプルの二乗末端間距離の範囲の最小値と最大値,  $R^2/xl^2$  として与える。実数型。入力。

R: 出力するサンプルの末端間距離。R(LA) なる実数型一次元配列。出力。

MDP, B, NOT, ICNT, OCOX, OCOY, OCOZ, GCOX, GCOY, GCOZ, NSUB, KA, LA, MA: サブルーチン ENRICH に同じ。

## (9) SGR

## 機能

関数副プログラムである。1本のサンプルについて, 各グルコース残基の重心の座標値から回転半径

の二乗を計算する。

呼び出し型式

SGR (X, Y, Z, MDP, MA)

引数の説明

X, Y, Z: 鎖の各残基の重心の座標。X (MA),

Y (MA), Z (MA) なる実数型一次元配列。

MDP: 重合度。整数型。

MA: X, Y, Zの整合寸法。整数型。

(10) CLOCK (CLOCKM)

機能

FORTRAN77 サービスルーチン。プログラムの実行開始からのCPU占有時間が秒単位(ミリ秒単位)で*i*に返される

呼び出し型式

CALL CLOCK (*i*)

### 引用文献

- 1) F. T. Wall and J. J. Erpenbeck, *J. Chem. Phys.*, **30**, 634 (1959).
- 2) P. J. Flory: "Statistical Mechanics of Chain Molecules", (1969), Interscience, New York; 日本語訳 安部明廣: 鎖状分子の統計力学, (1971), 倍風館。
- 3) C. V. Gobel, W. L. Dimpfl, and D. A. Brant, *Macromolecules*, **3**, 644 (1970)
- 4) B. A. Burton and D. A. Brant, *Biopolymers* **22**, 1769 (1983).
- 5) R. C. Jordan, D. A. Brant and A. Cesàro, *Biopolymers*, **17**, 2617 (1978)
- 6) D. A. Brant and W. L. Dimpfl, *Macromolecules*, **3**, 655 (1970).
- 7) M. E. Gress and G. A. Jefferey. *Acta Cryst.*, **33**, 2490 (1977).
- 8) S. S. C. Chu and G. A. Jefferey, *Acta Cryst.*, **23**, 1038 (1967).
- 9) A. Hybl, R. E. Rundle and D. E. Williams, *J. Am. Chem. Soc.*, **87**, 2779 (1965).
- 10) A. I. Kitaygorodski, *Tetrahedron*, **14**, 230 (1961).
- 11) W. Bruns and L. Vogel, *Colloid & Polymer Sci.*, **260**, 303 (1982).
- 12) 京都大学大型計算機センター, 利用の手引き(XYプロッタ編), (1981).
- 13) 上田光三郎, J6/CONTOR, 京都大学大型計算機センター・プログラム・ライブラリ (1980).
- 14) 富士通, FACOM FORTRAN SSL II 使用手引書, (1976).
- 15) 富士通, FACOM OS IV/F4 MSP GSP—3D 手引書, (1982).
- 16) 富士通, FACOM OS IV GSP 文法書(高級型用), (1982).
- 17) 別府良考, NAMOD (Nagoya Molecular Display), 名古屋大学大型計算機センターニュース Vol. 9, No. 2) (1979).

### Abstract

A series of computer programs in FORTRAN, which is made up six interrelated programs, was constructed to theoretically study the conformation of amylosic chain. The first program calculates conformational geometries of  $\alpha$ -D-maltose using structural models based on the crystal structures of its related compounds. The second program estimates conformational energies of the  $\alpha$ -maltose using semienpirical potential energy functions in the usual manner. The program involves the drawing of conformational energy contour map and the calculation of probability of conformations with respective conformational energies. The third program generates amylosic chains with various conformations so as to distributed consistent with the probability of respective conformations using a Monte Calro method, and estimates their average conformational properties. Outputs include the radi-

al distribution function of end to end length and the average value of characteristic ratio. Perspective drawing of the representative Monte Calro chains can be performed using another program. The fourth program takes the statistical mechanical theory (ref. 2,6) to calculate the average conformational properties of amylosic chains. Outputs include characteristic ratios at given chain lengths, components of the persistence vector of vertual bond, and persistence length. The last program dials with the problem of the excluded volume effect in the amylosic chain sequence. The program takes the Wall-Erpenbeck s-p method (ref. 1) of chain enrichment. Programs can be run together without re-entry of data. Most of programs may be available for the theoretical investigation of other polysaccharides conformations.

## PROGRAM LIST

```

DATA SET NAME : AKH0112.MALTOSE.FORT(PLCTEPCXY)

00010 C      MALTOSE ENERGY CALCULATION (L-J FUNCTION + COULOMB + TORSION)
00020        DIMENSION A0(3,18),B0(3,18),A(3,18),B(3,18)
00030        DIMENSION RHB(9,100),VHB(9,100),TOTALV(100,100),NN(100)
00040        DIMENSION CA(15),CC(15)
00050        DATA QH/0.076/,QC/0.031/,QCI/0.139/,QO/-0.216/,QOH/-0.108/,
00060        *      QCHH/0.0/
00070        DATA CA/0.0718,0.5599,0.2920,0.2726,1.457,3.929,2.256,2.114,9.550,
00080        *      1.309,1.231,5.512,1.159,5.159,22.97/
00090        DATA CC/46.48,126.2,98.19,91.64,225.7,361.0,292.1,273.7,634.4,
00100        *      243.8,229.4,506.5,215.8,474.0,1121.0/
00110        READ(5,500) A0,B0
00120 500     FORMAT(6F10.0)
00130        READ(5,501) ROCC,ROCO,ROCH,ROOO,ROOH,ROHH
00140 501     FORMAT(6F10.0)
00150        READ(5,502) PI,C1,C2,C3,R,T
00160 502     FORMAT(6F10.0)
00170        READ(5,*) DPHI,DPSI
00180 C
00190        WRITE(6,600) A0
00200 600     FORMAT(1H1,4HDATA//1H ,25HNON REDUCED GLUCOPYRANOSE/1H ,21HC1-C6,0
00210        *2-06,H1-H6,H6'/1H ,1HR,7X,5HTHETA,3X,3HPHI/
00220        *(1H ,5(F6.3,1X,F6.3,1X,F6.3,4X)))
00230        WRITE(6,601) B0
00240 601     FORMAT(1H //1H ,21HREDUCED GLUCOPYRANOSE/1H ,27HC1-C6,01-03,05,06
00250        * ,H1-H6,H6'/1H ,1HR,7X,5HTHETA,3X,3HPHI/
00260        *(1H ,5(F6.3,1X,F6.3,1X,F6.3,4X)))
00270        WRITE(6,609)
00280 609     FORMAT(1H //1H ,42HLEAVE H01-H04 AND H06 OUT OF CONSIDERATION)
00290        WRITE(6,618)
00300 618     FORMAT(1H /1H ,32HLENNARD-JONES FUNCTION PARAMETER)
00310        WRITE(6,610)
00320 610     FORMAT(1H ,85H      H-H      H-C      H-O      H-OH      H-CHH
00330        *      C-C      C-O      C-OH      )
00340        WRITE(6,611) (CA(I),I=1,8)
00350 611     FORMAT(1H ,5H CA ,8G10.4)
00360        WRITE(6,612) (CC(I),I=1,8)
00370 612     FORMAT(1H ,5H CC ,8G10.4)
00380        WRITE(6,613)
00390 613     FORMAT(1H ,75H      C-CHH      O-O      O-OH      O-CHH      OH-OH
00400        *      OH-CHH      CHH-CHH      )
00410        WRITE(6,614) (CA(I),I=9,15)
00420 614     FORMAT(1H ,5H CA ,7G10.4)
00430        WRITE(6,615) (CC(I),I=9,15)
00440 615     FORMAT(1H ,5H CC ,7G10.4)
00450        WRITE(6,619)
00460 619     FORMAT(1H /1H ,24HPARTIAL ELECTRONIC CHARGE)
00470        WRITE(6,620) QH,QC,QCI,QO,QOH,QCHH
00480 620     FORMAT(1H ,2X,3HQH=G11.5,2X,3HQC=G11.5,2X,4HQCI=G11.5,2X,
00490        *3HQO=G11.5,2X,4HOH=G11.5,2X,5HQCHH=G11.5/)
00500        WRITE(6,621)
00510 621     FORMAT(1H /1H ,28HPARAMETERS OF TORSION ENERGY/1H ,52HPHIKI= 1.8,
00520        * PHIKG= 1.1, PSIKI= 1.8 --- (KCAL/MOL)/)
00530        WRITE(6,603) PI,C1,C2,C3
00540 603     FORMAT(1H //1H ,3HPHI=F9.6/1H ,4HTAU=F 7.1/1H ,14HTURN INTERVAL=,
00550        *F 7.2/1H ,23HRANGE OF CALCULATION=+,-,F7.1/)
00560        WRITE(6,650)
00570 650     FORMAT(1H1/1H ,7HRESULTS//1H ,6HENENERGY/1H ,27HPHI(DEG)=,ENERGY(KCL
00580        */MOL)=/)
00590 C
00600 6      C=C2/180.0*PI
00610        D=(180-C1)/180*PI
00620        C4=117.0
00630        L=1
00640        H=IFIX(C3)*(0-1)
00650 1      DO 10 JJ=1,18
00660        B(1, JJ)=B0(1, JJ)*SIN(B0(2, JJ))*COS(B0(3, JJ)+C*M/C2)*COS(D)
00670        *-B0(1, JJ)*COS(B0(2, JJ))*SIN(D)
00680        B(2, JJ)=B0(1, JJ)*SIN(B0(2, JJ))*SIN(B0(3, JJ)+C*M/C2)
00690        B(3, JJ)=B0(1, JJ)*SIN(B0(2, JJ))*COS(B0(3, JJ)+C*M/C2)*SIN(D)
00700        *+B0(1, JJ)*COS(B0(2, JJ))*COS(D)
00710 10     CONTINUE
00720        N=1
00730 2      NH(N)=IFIX(C3)*(0-1)+IFIX(C2)*(N-1)
00740        DO 11 II=1,18
00750        A(1, II)=A0(1, II)*SIN(A0(2, II))*COS(A0(3, II)-C*NN(N)/C2)
00760        A(2, II)=A0(1, II)*SIN(A0(2, II))*SIN(A0(3, II)-C*NN(N)/C2)
00770        A(3, II)=A0(1, II)*COS(A0(2, II))
00780 11     CONTINUE
00790        TOTALV(N,L)=0.0
00800        CALL LJF (12,16,12,16,CA(1),CC(1),A,B,TOTALV,N,L)
00810        CALL LJF (12,16,1,5,CA(2),CC(2),A,B,TOTALV,N,L)
00820        CALL LJF (1,5,12,16,CA(2),CC(2),A,B,TOTALV,N,L)
00830        CALL LJF (12,16,10,10,CA(3),CC(3),A,B,TOTALV,N,L)
00840        CALL LJF (10,10,12,16,CA(3),CC(3),A,B,TOTALV,N,L)
00850        CALL LJF (12,16,7,9,CA(4),CC(4),A,B,TOTALV,N,L)
00860        CALL LJF (7,9,12,16,CA(4),CC(4),A,B,TOTALV,N,L)
00870        CALL LJF (12,16,6,6,CA(5),CC(5),A,B,TOTALV,N,L)
00880        CALL LJF (6,6,12,16,CA(5),CC(5),A,B,TOTALV,N,L)
00890        CALL LJF (1,1,1,3,CA(6),CC(6),A,B,TOTALV,N,L)
00900        CALL LJF (1,1,5,5,CA(6),CC(6),A,B,TOTALV,N,L)
00910        CALL LJF (2,5,1,5,CA(6),CC(6),A,B,TOTALV,N,L)
00920        CALL LJF (1,5,10,10,CA(7),CC(7),A,B,TOTALV,N,L)
00930        CALL LJF (10,10,1,5,CA(7),CC(7),A,B,TOTALV,N,L)
00940        CALL LJF (1,5,7,9,CA(8),CC(8),A,B,TOTALV,N,L)
00950        CALL LJF (7,9,1,5,CA(8),CC(8),A,B,TOTALV,N,L)

```

```

00960 CALL LJF (1,5,6,6,CA(9),CC(9),A,B,TOTALV,N,L)
00970 CALL LJF (6,6,1,5,CA(9),CC(9),A,B,TOTALV,N,L)
00980 CALL LJF (10,10,10,10,CA(10),CC(10),A,B,TOTALV,N,L)
00990 CALL LJF (10,10,7,9,CA(11),CC(11),A,B,TOTALV,N,L)
01000 CALL LJF (7,9,10,10,CA(11),CC(11),A,B,TOTALV,N,L)
01010 CALL LJF (10,10,6,6,CA(12),CC(12),A,B,TOTALV,N,L)
01020 CALL LJF (6,6,10,10,CA(12),CC(12),A,B,TOTALV,N,L)
01030 CALL LJF (7,9,7,9,CA(13),CC(13),A,B,TOTALV,N,L)
01040 CALL LJF (7,9,6,6,CA(14),CC(14),A,B,TOTALV,N,L)
01050 CALL LJF (6,6,7,9,CA(14),CC(14),A,B,TOTALV,N,L)
01060 CALL LJF (6,6,6,6,CA(15),CC(15),A,B,TOTALV,N,L)
01070 CALL COULOM (1,1,1,1,QCI,QCI,A,B,TOTALV,N,L)
01080 CALL COULOM (1,1,2,3,QCI,QC,A,B,TOTALV,N,L)
01090 CALL COULOM (1,1,5,5,QCI,QC,A,B,TOTALV,N,L)
01100 CALL COULOM (1,1,7,9,QCI,QOH,A,B,TOTALV,N,L)
01110 CALL COULOM (1,1,10,10,QCI,QO,A,B,TOTALV,N,L)
01120 CALL COULOM (1,1,12,16,QCI,QH,A,B,TOTALV,N,L)
01130 CALL COULOM (2,5,1,1,QC,QCI,A,B,TOTALV,N,L)
01140 CALL COULOM (2,5,2,5,QC,QC,A,B,TOTALV,N,L)
01150 CALL COULOM (2,5,7,9,QC,QOH,A,B,TOTALV,N,L)
01160 CALL COULOM (2,5,10,10,QC,QO,A,B,TOTALV,N,L)
01170 CALL COULOM (2,5,12,16,QC,QH,A,B,TOTALV,N,L)
01180 CALL COULOM (7,9,1,1,QOH,QCI,A,B,TOTALV,N,L)
01190 CALL COULOM (7,9,2,5,QOH,QC,A,B,TOTALV,N,L)
01200 CALL COULOM (7,9,7,9,QOH,QOH,A,B,TOTALV,N,L)
01210 CALL COULOM (7,9,10,10,QOH,QO,A,B,TOTALV,N,L)
01220 CALL COULOM (7,9,12,16,QOH,QH,A,B,TOTALV,N,L)
01230 CALL COULOM (10,10,1,1,QO,QCI,A,B,TOTALV,N,L)
01240 CALL COULOM (10,10,2,5,QO,QC,A,B,TOTALV,N,L)
01250 CALL COULOM (10,10,7,9,QO,QOH,A,B,TOTALV,N,L)
01260 CALL COULOM (10,10,10,10,QO,QO,A,B,TOTALV,N,L)
01270 CALL COULOM (10,10,12,16,QO,QH,A,B,TOTALV,N,L)
01280 CALL COULOM (12,16,1,1,QH,QCI,A,B,TOTALV,N,L)
01290 CALL COULOM (12,16,2,5,QH,QC,A,B,TOTALV,N,L)
01300 CALL COULOM (12,16,7,9,QH,QOH,A,B,TOTALV,N,L)
01310 CALL COULOM (12,16,10,10,QH,QO,A,B,TOTALV,N,L)
01320 CALL COULOM (12,16,12,16,QH,QH,A,B,TOTALV,N,L)
01330 CALL TORSIO (M,NN,DPHI,DPSI,TOTALV,N,L,PI)
01340 IF(NN(N).GE.IFIX(C3)) GO TO 3
01350 N=N+1
01360 GO TO 2
01370 3 WRITE(6,660) C1,M,(NN(K),TOTALV(K,L),K=1,N)
01380 660 FORMAT(1H //1H ,4HTAU=,F7.1/1H ,4HPSI=,I4/(1H ,6(I4,1X,E12.4,4X))
01390 IF(M.GE.IFIX(C3)) GO TO 4
01400 M=M+IFIX(C2)
01410 L=L+1
01420 GO TO 1
01430 C
01440 4 IX=IFIX(C3*(-1.0))
01450 IY=IX
01460 ID=IFIX(C2)
01470 WRITE(8,*) IX,IY,ID
01480 WRITE(8,*) N
01490 C
01500 L1=L
01510 N1=N
01520 WRITE(6,670) R,T
01530 670 FORMAT(1H //1H ,14HPROBABILITY(%)1H ,2HR=,E11.4,1H ,2HT=,F7.2)
01540 CALL PIP1 (L1,N1,TOTALV,C1,C2,C3,R,T)
01550 IF(C1.GE.C4) GO TO 5
01560 C1=C1+2.0
01570 GO TO 6
01580 C
01590 5 WRITE(7,700) C1,C2,C3
01600 700 FORMAT(3F10.6)
01610 WRITE(7,710) ((TOTALV(I,J),I=1,N),J=1,L)
01620 710 FORMAT(4E15.7)
01630 C
01640 STOP
01650 END
01660 C
01670 SUBROUTINE LJF (I1,I2,J1,J2,CA,CC,A,B,TOTALV,N,L)
01680 DIMENSION A(3,18),B(3,18),TOTALV(100,100)
01690 DO 100 J=J1,J2
01700 DO 101 I=I1,I2
01710 R=SQRT((A(1,I)-B(1,J))**2+(A(2,I)-B(2,J))**2+(A(3,I)-B(3,J))**2)
01720 V=CA*100000.0/R**12-CC/R**6
01730 TOTALV(N,L)=TOTALV(N,L)+V
01740 101 CONTINUE
01750 100 CONTINUE
01760 RETURN
01770 END
01780 C
01790 SUBROUTINE COULOM (I1,I2,J1,J2,Q1,Q2,A,B,TOTALV,N,L)
01800 DIMENSION A(3,18),B(3,18),TOTALV(100,100)
01810 DO 100 J=J1,J2
01820 DO 101 I=I1,I2
01830 R=SQRT((A(1,I)-B(1,J))**2+(A(2,I)-B(2,J))**2+(A(3,I)-B(3,J))**2)
01840 V=332*Q1*Q2/4.0/R
01850 TOTALV(N,L)=TOTALV(N,L)+V
01860 101 CONTINUE
01870 100 CONTINUE
01880 RETURN
01890 END
01900 C
01910 SUBROUTINE TORSIO (M,NN,DPHI,DPSI,TOTALV,N,L,PI)
01920 DIMENSION TOTALV(100,100),NN(100)
01930 PHIT=FLOAT(NN(N))+DPHI
01940 PSIT=FLOAT(M)+DPSI
01950 IF(PHIT.GT.0.0.AND.PHIT.LT.120.0) PHIKG=1.1
01960 IF(PHIT.LE.0.0.OR.PHIT.GE.120.0) PHIKG=0.0

```

```

01970      PHIKI=1.8
01980      PSIKI=1.8
01990      PHIT=PHIT/180.0*PI
02000      PSIT=PSIT/180.0*PI
02010      V=(PHIKI/2.0)*(1.0+COS(3.0*PHIT))+
02020      *(PHIKG/2.0)*(1.0-COS(3.0*PHIT))+
02030      *(PSIKI/2.0)*(1.0+COS(3.0*PSIT))
02040      TOTALV(N,L)=TOTALV(N,L)+V
02050      RETURN
02060      END
02070 C
02080      SUBROUTINE PIPI (L1,N1,TOTALV,C1,C2,C3,R,T)
02090      DIMENSION TOTALV(100,100),S(100,100),P(100,100),NN(100)
02100      TOTALS=0.0
02110      DO 204 L=1,L1
02120      DO 203 N=1,N1
02130      IF(ABS(TOTALV(N,L)).GE.6.000) GO TO 201
02140      S(N,L)=EXP(TOTALV(N,L)*(0.0-1.0)/R/T)
02150      GO TO 202
02160 201  S(N,L)=0.0
02170 202  TOTALS=TOTALS+S(N,L)
02180 203  CONTINUE
02190 204  CONTINUE
02200      L=1
02210      M=IFIX(C3)*(0-1)
02220 205  N=1
02230 206  NN(N)=IFIX(C3)*(0-1)+IFIX(C2)*(N-1)
02240      P(N,L)=S(N,L)/TOTALS*100.0
02250      IF(NN(N).GE.IFIX(C3)) GO TO 207
02260      N=N+1
02270      GO TO 206
02280 207  WRITE(6,680) C1,M,(NN(K),P(K,L),K=1,N)
02290 680  FORMAT(1H //1H ,4HTAU=,F7.1/1H ,4HPSI=,I4/(1H ,7(I4,1X,FB.5,3X)))
02300      IF(M.GE.IFIX(C3)) GO TO 208
02310      M=M+IFIX(C2)
02320      L=L+1
02330      GO TO 205
02340 C
02350 208  WRITE(8,*) ((P(I,J),I=1,N),J=1,L)
02360 C
02370      RETURN
02380      END

      DATA SET NAME : AHH3703.TAC.FORT77(MAIN)

00010 "      *****
00020 "      *
00030 "      *          MAIN PROGRAM          *
00040 "      *
00050 "      *****
00060 "
00070 DIMENSION P(0:2000),IPHI(2000),IPSI(2000),PP(2000),ICNT(2000),MPHI(5000),MPSI(5000),-
00080 COOX(0:5000),COOY(0:5000),COOZ(0:5000),SR(5000),R(5000),CSR(11,5000),-
00090 W(20,5000),CSSR(11),CX(11),NTH(17),NFO(17),NFI(17),NSI(17),NSE(17),-
00100 TH(17),FO(17),FI(17),SI(17),SE(17),WR(5000),SW(20,5000),CCKR(20),CK(20)
00110 "
00120 "      ***** DATA READ *****
00130 "
00140 READ(8,*) THE,ETA,XI,OME,VLEN
00150 READ(8,*) IX,IY,ID
00160 READ(8,*) N
00170 READ(8,*) ( P(I),I=1,N**2 )
00180 "
00190 WRITE(6,*) '          << DP= ? >> '
00200 READ(5,*) MDP
00210 WRITE(6,*) '          << NOC=? >> '
00220 READ(5,*) NOC
00230 WRITE(6,*) '          << DO YOU WANT GRAPHIC DATA ?? >> '
00240 WRITE(6,*) '          YES= 1.0 NO=0.0'
00250 READ(5,*) XG
00260 "
00270 "      ***** CHAIN GENERATION *****
00280 "
00290 CALL INTRND(IX,IY,ID,N,P,PP,IPHI,IPSI,NOD,0.0,2000)
00300 ISEED=0
00310 IXG=1
00320 NOCG=5
00330 IF(XG.EQ.1.0) THEN
00340     IF(NOC.LT.NOCG) NOCG=NOC
00350     WRITE(7,*) NOCG
00360 END IF
00370 DO 10 I=1,NOC
00380     CALL RNDGEN(P,IPHI,IPSI,ICNT,NOD,MDP,ISEED,MPHI,MPSI,0.0,2000,5000)
00390     CALL CHAIN(MPHI,MPSI,MDP,VLEN,COOX,COOY,COOZ,THE,ETA,XI,OME,0.0,5000)
00400     SR(I)=COOX(MDP)**2+COOY(MDP)**2+COOZ(MDP)**2
00410     R(I)=SQRT(SR(I))
00420     JR=10
00430     DO 20 J=1,9
00440         JR=JR+10
00450         CSR(J,I)=COOX(JR)**2+COOY(JR)**2+COOZ(JR)**2
00460         20 CONTINUE
00470         CSR(10,I)=COOX(150)**2+COOY(150)**2+COOZ(150)**2
00480         CSR(11,I)=COOX(200)**2+COOY(200)**2+COOZ(200)**2
00490 "
00500     DO 30 K=1,20
00510         SW(K,I)=COOX(K)**2+COOY(K)**2+COOZ(K)**2
00520         W(K,I)=SQRT(SW(K,I))
00530         30 CONTINUE
00540 "

```

```

00550      IF(XG.EQ.1.0) THEN
00560      HG=MDP
00570      IF(IXG.LE. 5) THEN
00580      IXG=IXG+1
00590      IF(MG.GT.200) MG=200
00600      WRITE(7,*) MG
00610      WRITE(7,*) ( COOX(II),II=0,MG )
00620      WRITE(7,*) ( COOY(II),II=0,MG )
00630      WRITE(7,*) ( COOZ(II),II=0,MG )
00640      END IF
00650      END IF
00660      10 CONTINUE
00670 "
00680 "      ***** WRITEING OF DATA *****
00690 "
00700      WRITE(6,*) ' '
00710      WRITE(6,*) ' '          ***** INPUT DATA *****
00720      WRITE(6,*) ' '
00730      WRITE(6,*) ' THETA= ',THE,' ETA= ',ETA,' XI= ',XI,' OMEGA= ',OME
00740      WRITE(6,*) ' DP= ',MDP,' NUMBER FO CHAINS= ',NOC,' BOND LENGTH= ',VLEN
00750 "
00760 "      ***** << CALCULATION FO CHARACTERISTIC RATIO >> *****
00770 "
00780      SSR=0.0
00790      DO 40 J=1,11
00800      CSSR(J)=0.0
00810      40 CONTINUE
00820      DO 41 K=1,19
00830      CKKR(K)=0.0
00840      41 CONTINUE
00850 "
00860      DO 50 I=1,NOC
00870      SSR=SSR+SR(I)
00880      50 CONTINUE
00890      DO 61 I=1,NOC
00900      DO 60 J=1,11
00910      CSSR(J)=CSSR(J)+CSR(J,I)
00920      60 CONTINUE
00930      DO 62 K=1,19
00940      CKKR(K)=CKKR(K)+SW(K,I)
00950      62 CONTINUE
00960      61 CONTINUE
00970 "
00980      CXM=SSR/NOC/MDP/VLEN**2
00990      JR=10
01000      DO 70 J=1,9
01010      JR=JR+10
01020      CX(J)=CSSR(J)/NOC/JR/VLEN**2
01030      70 CONTINUE
01040      KK=0
01050      DO 71 K=1,19
01060      KK=KK+1
01070      CK(K)=CKKR(K)/NOC/KK/VLEN**2
01080      71 CONTINUE
01090      CX(10)=CSSR(10)/NOC/150/VLEN**2
01100      CX(11)=CSSR(11)/NOC/200/VLEN**2
01110 "
01120      WRITE(6,*) ' '
01130      WRITE(6,*) ' '
01140      WRITE(6,*) ' '          ***** << CHARACTERISTIC RATIO CX >> *****
01150      WRITE(6,*) ' DP= ',MDP,' C(DP)= ',CXM
01160      WRITE(6,*) ' '
01170      KK=0
01180      DO 81 K=1,19
01190      KK=KK+1
01200      WRITE(6,*) ' DP= ',KK,' C(DP)= ',CK(K)
01210      81 CONTINUE
01220      JR=10
01230      DO 80 J=1,9
01240      IF(CX(J).GT. 0.0) THEN
01250      JR=JR+10
01260      WRITE(6,*) ' DP= ',JR,' C(DP)= ',CX(J)
01270      END IF
01280      80 CONTINUE
01290      WRITE(6,*) ' DP= 150 C(DP)= ',CX(10)
01300      WRITE(6,*) ' DP= 200 C(DP)= ',CX(11)
01310 "
01320 "      ***** HISTGRAM FO DISTRIBUTION FUNCTION W(R) *****
01330 "
01340 "
01350      DO 140 I=1,NOC
01360      WR(I)=R(I)
01370      140 CONTINUE
01380      IMDP=MDP
01390      999 CMAX=WR(1)
01400      CMIN=WR(1)
01410      DO 130 I=1,NOC
01420      IF(WR(I).GT.CMAX) CMAX=WR(I)
01430      IF(WR(I).LT.CMIN) CMIN=WR(I)
01440      130 CONTINUE
01450      B=CMAX-CMIN
01460      IF(B.GT.0.0 .AND. B.LE.102400.0) THEN
01470      IF(B.LT.60.0) IN=1
01480      IF(B.GE.60.0 .AND. B.LT.100.0) IN=2
01490      IF(B.GE.100.0 .AND. B.LT.200.0) IN=4
01500      IF(B.GE.200.0 .AND. B.LT.400.0) IN=8
01510      IF(B.GE.400.0 .AND. B.LT.800.0) IN=16
01520      IF(B.GE.800.0 .AND. B.LT.1600.0) IN=32
01530      IF(B.GE.1600.0 .AND. B.LT.3200.0) IN=64
01540      IF(B.GE.3200.0 .AND. B.LT.6400.0) IN=128
01550      IF(B.GE.6400.0 .AND. B.LT.12800.0) IN=256

```

```

01560         IF(B.GE.12800.0 .AND. B.LT.25600.0)   IN=512
01570         IF(B.GE.25600.0 .AND. B.LT.51200.0)   IN=1042
01580         IF(B.GE.51200.0 .AND. B.LT.104200.0)  IN=2048
01590 "
01600         CALL HIST(WR,IMDP,NOC,IN,5000)
01610     END IF
01620 888 WRITE(6,*)' '
01630     WRITE(6,*)' '
01640     WRITE(6,*)'         *** DO YOU WANT THE HISTGRAM W(R) (4<R<20) ?? ***'
01650     WRITE(6,*)'         DP= ?'
01660     READ(5,*) IDP
01670     IF(IDP.GE.4 .AND. IDP.LE.20) THEN
01680         DO 150 I=1,NOC
01690             WR(I)=0.0
01700             WR(I)=W(IDP,I)
01710             150 CONTINUE
01720     IMPD=IDP
01730     GO TO 999
01740     END IF
01750 "
01760     DO 90 K=4,20
01770         NTH(K)=0
01780         NFO(K)=0
01790         NFI(K)=0
01800         NSI(K)=0
01810         NSE(K)=0
01820     90 CONTINUE
01830 "
01840     DO 100 K=4,20
01850         DO 110 I=1,NOC
01860             IF(W(K,I).GT.0.0 .AND. W(K,I).LE.3.0) NTH(K)=NTH(K)+1
01870             IF(W(K,I).GT.0.0 .AND. W(K,I).LE.4.0) NFO(K)=NFO(K)+1
01880             IF(W(K,I).GT.0.0 .AND. W(K,I).LE.5.0) NFI(K)=NFI(K)+1
01890             IF(W(K,I).GT.0.0 .AND. W(K,I).LE.6.0) NSI(K)=NSI(K)+1
01900             IF(W(K,I).GT.0.0 .AND. W(K,I).LE.7.0) NSE(K)=NSE(K)+1
01910         110 CONTINUE
01920     100 CONTINUE
01930 "
01940     DO 120 K=4,20
01950         TH(K)=FLOAT(NTH(K))/FLOAT(NOC)*100.0
01960         FO(K)=FLOAT(NFO(K))/FLOAT(NOC)*100.0
01970         FI(K)=FLOAT(NFI(K))/FLOAT(NOC)*100.0
01980         SI(K)=FLOAT(NSI(K))/FLOAT(NOC)*100.0
01990         SE(K)=FLOAT(NSE(K))/FLOAT(NOC)*100.0
02000     120 CONTINUE
02010     WRITE(6,511) NOC
02020     WRITE(6,501) ( TH(K),K=4,20 )
02030     WRITE(6,502) ( FO(K),K=4,20 )
02040     WRITE(6,503) ( FI(K),K=4,20 )
02050     WRITE(6,504) ( SI(K),K=4,20 )
02060     WRITE(6,505) ( SE(K),K=4,20 )
02070 511 FORMAT(/ 15X,35H***** << PROBABILITY P(R) (NOC=,IS,12H) >> ***** / 5X-
02080     ,8H (%) 4,5X,1H5,5X,1H6,5X,1H7,5X,1H8,5X,1H9,4X,2H10,4X,2H11,4X,-
02090     2H12,4X,2H13,4X,2H14,4X,2H15,4X,2H16,4X,2H17,4X,2H18,4X,2H19,4X,2H20)
02100 501 FORMAT(3X,7HP( 3 ),20(F5.2,1X))
02110 502 FORMAT(3X,7HP( 4 ),20(F5.2,1X))
02120 503 FORMAT(3X,7HP( 5 ),20(F5.2,1X))
02130 504 FORMAT(3X,7HP( 6 ),20(F5.2,1X))
02140 505 FORMAT(3X,7HP( 7 ),20(F5.2,1X))
02150 "
02160 STOP
02170 END

```

DATA SET NAME : AKH0112.TACLIB.FORT77(INTRND)

```

00010 "
00020 "***** SUBROUTINE INTRND (INITIALIZATION FOR RANDOM GENERATION) *****
00030 "
00040 SUBROUTINE INTRND(IX,IY,ID,N,P,PP,IPHI,IPSI,NOD,WRT,K)
00050 DIMENSION P(0:K),IPHI(K),IPSI(K),PP(K)
00060 "
00070 " ----- GENERATION OF PHI AND PSI -----
00080 "
00090     DO 10 J=1,N
00100         IXX=IX
00110         DO 20 I=1,N
00120             IPHI((J-1)*N+I)=IXX
00130             IPSI((J-1)*N+I)=IY
00140             IXX=IXX+ID
00150         20 CONTINUE
00160         IY=IY+ID
00170     10 CONTINUE
00180 "
00190 " ----- INITIALIZATION FOR RANDOM GENERATION -----
00200 "
00210     P(0)=0.0
00220     J=1
00230     DO 30 I=1,N**2
00240         IF(P(I).EQ.0.0) GO TO 30
00250         PP(J)=P(I)
00260         P(J)=P(J-1)+P(I)
00270         IPHI(J)=IPHI(I)
00280         IPSI(J)=IPSI(I)
00290         J=J+1
00300     30 CONTINUE
00310     NOD=J-1
00320 "
00330 " ----- NOMALIZATION -----
00340 "

```



```

00350     PN=P(NOD)
00360     DO 40 I=1,NOD
00370         P(I)=P(I)/PN
00380         PP(I)=PP(I)/PN
00390     40 CONTINUE
00400 "
00410 " ----- WRITING OF RESULTS -----
00420 "
00430     IF (WRT.EQ.1.0) THEN
00440         WRITE(6,*) ' '
00450         WRITE(6,*) ' *** INITIALIZED DATA FOR RANDOM GENERATION ***'
00460         WRITE(6,*) ' '
00470         WRITE(6,*) ' I IPHI IPSI PP P'
00480         WRITE(6,*) ' '
00490         WRITE(6,100) (I,IPHI(I),IPSI(I),PP(I),P(I),I=1,NOD)
00500     100 FORMAT(3X,I4,3X,I4,3X,I4,3X,I4,3X,E15.7,3X,E15.7)
00510     END IF
00520     RETURN
00530     END

```

DATA SET NAME : AKH0112.TACLIB.FORT77(RNDGEN)

```

00010 "
00020 " ***** SUBROUTINE RNDGEN (RANDOM GENERATION) *****
00030 "
00040 SUBROUTINE RNDGEN(P,IPHI,IPSI,ICNT,NOD,MDP,ISEED,MPHI,MPSI,WRT,K,L)
00050 DIMENSION P(0:K),IPHI(K),IPSI(K),ICNT(K),MPHI(L),MPSI(L),RND(100000)
00060 "
00070 " ----- GENERATION OF UNIFORMLY RANDOM NUMBERS -----
00080 "
00090     CALL RANU2(ISEED,RND,MDP,ICON)
00100 "
00110 " --- TRANSFORMATION OF UNIFORMLY RANDOM NUMBERS TO ROTATION ---
00120 " --- ANGLES, PHI AND PSI (BINARY SARCHING) ---
00130 "
00140     DO 10 I=1,MDP
00150         L=0
00160         IH=NOD
00170     110     M1=(L+IH)/2
00180         M2=M1+1
00190         IF (L.GT.IH) THEN
00200             WRITE(6,*) ' * ERROR1'
00210             STOP
00220         ELSE IF (P(M1).LE.RND(I).AND.P(M2).GT.RND(I)) THEN
00230             MPHI(I)=IPHI(M2)
00240             MPSI(I)=IPSI(M2)
00250             GO TO 100
00260         ELSE IF (P(M1).GT.RND(I)) THEN
00270             IH=M1
00280             GO TO 110
00290         ELSE IF (P(M2).LE.RND(I)) THEN
00300             L=M2
00310             GO TO 110
00320         ELSE
00330             WRITE(6,*) ' * ERROR2'
00340             STOP
00350         END IF
00360     100     ICNT(M2)=ICNT(M2)+1
00370     10 CONTINUE
00380 "
00390 " ----- WRITING OF RESULTS -----
00400 "
00410     IF (WRT.EQ.1.0) THEN
00420         WRITE(6,*) ' '
00430         WRITE(6,*) ' **** FREQUENCY OF GENERATED PHI AND PSI ****'
00440         WRITE(6,*) ' '
00450         WRITE(6,*) ' ----- FREQUENCY -----'
00460         WRITE(6,*) ' '
00470         WRITE(6,*) ('* PHI=',IPHI(I),' PSI=',IPSI(I),' FREQ.=',ICNT(I),
00480             ' ',I=1,NOD)
00490     END IF
00500     RETURN
00510     END

```

DATA SET NAME : AKH0112.TACLIB.FORT77(CHAIN)

```

00010 " ***** SUBROUTINE CHAIN (FOR CHAIN GENERATION) *****
00020 "
00030 SUBROUTINE CHAIN(MPHI,MPSI,MDP,VLEN,COOX,COOY,COOZ,THE,ETA,XI,OME,WRT,L)
00040 DIMENSION MPHI(L),MPSI(L),COOX(0:L),COOY(0:L),COOZ(0:L),
00050     A(4,4),B(4,4),E(4,4),TK(3,3),VW(4)
00060 DATA E/1.0,0.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,1.0/
00070 "
00080 " ----- INITIALIZATION -----
00090 "
00100     DO 10 I=1,4
00110         DO 10 J=1,4
00120             B(I,J)=E(I,J)
00130     10 CONTINUE
00140     A(1,4)=VLEN
00150     A(2,4)=0.0
00160     A(3,4)=0.0
00170     A(4,4)=1.0
00180     A(4,1)=0.0
00190     A(4,2)=0.0
00200     A(4,3)=0.0
00210     COOX(0)=0.0

```

```

00220      COOY(0)=0.0
00230      COOZ(0)=0.0
00240      "
00250      " ----- COORDINATES OF GLYCOSIDIC LINKAGE OXYGENS -----
00260      "
00270      DO 20 I=1,MDP
00280          PHI=FLOAT(MPHI(I))
00290          PSI=FLOAT(MPSI(I))
00300          CALL DCTMT(PHI,PSI,THE,ETA,XI,OME,TK)
00310          DO 30 J=1,3
00320              DO 30 K=1,3
00330                  A(J,K)=TK(J,K)
00340          30 CONTINUE
00350          CALL MGGM(B,4,A,4,B,4,4,4,4,VW,ICON)
00360          COOX(I)=B(1,4)
00370          COOY(I)=B(2,4)
00380          COOZ(I)=B(3,4)
00390      20 CONTINUE
00400      "
00410      " ----- WRITING OF RESULTS -----
00420      "
00430      IF(WRT.EQ.1.0) THEN
00440          WRITE(6,*) ' '
00450          WRITE(6,*) ' ***** COORDINATES OF GLYCOSIDIC LINKAGE OXYGENS *****'
00460          WRITE(6,*) ' '
00470          WRITE(6,*) '      NO.          X          Y          Z'
00480          WRITE(6,*) ' '
00490          WRITE(6,100) (I+1,COOX(I),COOY(I),COOZ(I),I=0,MDP)
00500      100  FORMAT(3X,I4,3X,E15.7,5X,E15.7,5X,E15.7)
00510      END IF
00520      RETURN
00530      END

```

DATA SET NAME : AHH3703.TAC.FORT77(CXPER)

```

00010 "*****
00020 "*"
00030 "*" << CX , AND , PERSISTENCE > > "*"
00040 "*"
00050 "*****
00060 "
00070 DIMENSION ATK(3,3),P(0:2000)
00080 "
00090 " ***** DATA READ *****
00100 "
00110 READ(8,*) THE,ETA,XI,OME,VLEN
00120 READ(8,*) IX,IY,ID
00130 READ(8,*) N
00140 READ(8,*) (P(I),I=1,N**2)
00150 "
00160 WRITE(6,*) ' *** << DP= ? >> ***'
00170 READ(5,*) MDP
00180 "
00190 " ***** CALCULATE *****
00200 "
00210 CALL ACTMT(IX,IY,ID,THE,ETA,XI,OME,N,P,ATK,1.0)
00220 CALL MSETED(ATK,-1)
00230 IF(MDP.GT. 0) THEN
00240     DO 10 M=1,9
00250         CALL MSETED(ATK,M)
00260     10 CONTINUE
00270     DO 20 N=10,MDP,10
00280         CALL MSETED(ATK,N)
00290     20 CONTINUE
00300 END IF
00310 "
00320 CALL PERSIS(ATK,VLEN,PSIS,0.001)
00330 STOP
00340 END

```

DATA SET NAME : AKH0112.TACLIB.FORT77(ACTMT)

```

00010 "
00020 " ***** SUBROUTINE FOR AVERAGED MATRIX , <TK> *****
00030 "
00040 SUBROUTINE ACTMT(IX,IY,ID,THE,ETA,XI,OME,N,P,ATK,WRT)
00050 DIMENSION P(0:2000),T(3,6000),V(6000,3),A(3,6000),PP(2000),IPHI(2000),IPSI(2000) -
00060     ,E(3,3),EE(3,3),VW(6000),W(1,2000),TK(3,3),ATK(3,3)
00070 DATA E/1.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,1.0/
00080 DATA EE/1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0/
00090 "
00100 " ----- GENERATION OF MATRIX , W -----
00110 "
00120 CALL INTRND(IX,IY,ID,N,P,PP,IPHI,IPSI,NOD,0.0,2000)
00130 DO 10 I=1,NOD
00140     W(1,I)=PP(I)
00150 10 CONTINUE
00160 "
00170 " ----- GENERATION OF MATRIX , T -----
00180 "
00190 DO 20 J=1,3*NOD,3
00200     PHI=FLOAT(IPHI((J-1)/3+1))
00210     PSI=FLOAT(IPSI((J-1)/3+1))
00220     CALL DCTMT(PHI,PSI,THE,ETA,XI,OME,TK)
00230     DO 30 I=1,3
00240         DO 30 K=0,2
00250             T(I,J+K)=TK(I,K+1)

```

```

                                30 CONTINUE
00260          20 CONTINUE
00270
00280 "
00290 " ----- GENERATION OF MATRIX , V -----
00300 "
00310 "       DO 40 K=1,3*NOD,3
00320 "           DO 40 I=0,2
00330 "               DO 40 J=1,3
00340 "                   V(I+K,J)=E(I+1,J)
00350 "       40 CONTINUE
00360 "
00370 " ----- CALCULATION OF AVERAGED MATRIX , <TK> -----
00380 "
00390 "       CALL DIRECT(W,1,NOD,EE,3,3,A,MA,NA,1,2000,3,3)
00400 "       CALL STAR(A,T,A,MA,NA,3,6000)
00410 "       CALL HGGH(A,3,V,6000,ATK,3,MA,NA,MA,VW,ICON)
00420 "
00430 " ----- WRITING OF RESULTS -----
00440 "
00450 "       IF(WRT.EQ.1.0) THEN
00460 "           WRITE(6,*) ' '
00470 "           WRITE(6,*) ' **** AVERAGED MATRIX , <TK> ****'
00480 "           WRITE(6,*) ' '
00490 "           WRITE(6,*) '           1 COLUMN           2 COLUMN           3 COLUMN'
00500 "           WRITE(6,*) ' '
00510 "           DO 100 I=1,3
00520 "               100 WRITE(6,1000) I,(ATK(I,J),J=1,3)
00530 "           WRITE(6,*) ' '
00540 "       ENDIF
00550 "       RETURN
00560 "
00570 " ----- FORMAT STATEMENTS -----
00580 "
00590 " 1000 FORMAT(I2,' ROW ',3(E15.7,5X))
00600 "       END

                                DATA SET NAME : AKH0112.TACLIB.FORT77(MSETED)

00010 " *****
00020 " * * * * *
00030 " * SUBROUTINE MSETED *
00040 " * * * * *
00050 " *****
00060 "
00070 SUBROUTINE MSETED(ATK,MDP)
00080 "
00090 DIMENSION ATK(3,3),E(3,3),C(3,3),ADDET(3,3),SUBET(3,3),EXPTN(3,3),SUBIN(3,3),SU2IN(3,3)
00100 DIMENSION SUBTT(3,3),PROAI(3,3),SU2ET(3,3),PROS2(3,3),PRO22(3,3),X(3,3)
00110 "
00120 DATA E/1.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,1.0/
00130 "
00140 "       CALL ADD(E,ATK,ADDET,3,3)
00150 "       CALL SUBTRA(E,ATK,SUBET,3,3)
00160 "       CALL INVERS(SUBET,SUBIN,T1)
00170 "       CALL PRODAC(ADDET,SUBIN,PROAI,3,3,3)
00180 "
00190 "       IF(MDP.LT.1 .OR. MDP.GT.10000) THEN
00200 "           DO 40 I=1,3
00210 "               DO 40 J=1,3
00220 "                   40 C(I,J)=PROAI(I,J)
00230 "                   GO TO 222
00240 "       END IF
00250 "
00260 "       DO 10 I=1,3
00270 "           DO 10 J=1,3
00280 "       10 X(I,J)=ATK(I,J)
00290 "       N=1
00300 "
00310 " 111 CALL PRODAC(X,ATK,EXPTN,SUBTT,3,3)
00320 "       N=N+1
00330 "       DO 30 I=1,3
00340 "           DO 30 J=1,3
00350 "       30 X(I,J)=EXPTN(I,J)
00360 "       IF (N.LT.MDP+1) GO TO 111
00370 "
00380 "       CALL SUBTRA(ATK,EXPTN,SUBTT,3,3)
00390 "
00400 "       CALL PRODAC(SUBET,SUBET,SU2ET,3,3,3)
00410 "       CALL INVERS(SU2ET,SU2IN,T2)
00420 "
00430 "       CALL PRODAC(SUBTT,SU2IN,PROS2,3,3,3)
00440 "
00450 "           DO 20 I=1,3
00460 "               DO 20 J=1,3
00470 "       20 PRO22(I,J)=2*PROS2(I,J)/MDP
00480 "
00490 "       CALL SUBTRA(PROAI,PRO22,C,3,3)
00500 " 222 CHARA=C(1,1)
00510 "
00520 "       IF(MDP.LT.1 .OR. MDP.GT.10000) THEN
00530 "           WRITE(6,*) ' ***** << CHARACTERISTIC RATIO >> *****'
00540 "           WRITE(6,100) CHARA
00550 "           100 FORMAT( // 16X,14HMDP= INFINITY / 10X,10H***** CN=,F12.6,7H ***** //)
00560 "           GO TO 333
00570 "       END IF
00580 "
00590 "       WRITE(6,120) MDP,CHARA
00600 " 120 FORMAT(16X,4HMDP=,I8,10X,10H***** CX=,F12.6,7H ***** )
00610 "
00620 " 333 RETURN
00630 "       END

```

```

DATA SET NAME : AKH0112.TACLIB.FORT77(PERSIS)

00010 "*****
00020 "*"
00030 "*"          SUBROUTINE PERSIS          *
00040 "*"          *                          *
00050 "*****
00060 "
00070 SUBROUTINE PERSIS(ATK,VLEN,PSIS,VEC)
00080 "
00090 DIMENSION  ATK(3,3),TATK(3,3),PT(3),DATA(3,3)
00100 "
00110 DATA      TATK/1.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,1.0/
00120          PX=0.0
00130          PY=0.0
00140          PZ=0.0
00150          D=0.0
00160          DJ=0.0
00170 "
00180          N=2
00190          PT(1)=VLEN
00200 "
00210          WRITE(6,*) ' '
00220          WRITE(6,*) '          ***** << PERSISTENCE LENGTH >> *****'
00230          WRITE(6,*) ' '
00240          WRITE(6,*) '          *** DO YOU WANT X,Y,Z-COORDINATES ***'
00250          WRITE(6,*) '          YES=1.0 NO=0.0'
00260          READ(5,*) COOR
00270          IF(COOR.EQ.1.0)-
00280          WRITE(6,*) '          NOC          X          Y          Z          PER-V          DIS'
00290 999 D=SQRT(PT(1)**2+PT(2)**2+PT(3)**2)
00300          DJ=SQRT((PT(1)-PX)**2+(PT(2)-PY)**2+(PT(3)-PZ)**2)
00310          IF(COOR.EQ.1.0) THEN
00320              WRITE(6,501) N,PT(1),PT(2),PT(3),D,DJ
00330              501 FORMAT(5X,I4,7X,3(F7.3,3X),3X,F7.4,5X,F7.3)
00340              IF(N.GT.50) COOR=0.0
00350          END IF
00360          IF(DJ.GT.VEC) THEN
00370              CALL PRODAC(TATK,ATK,DATA,3,3,3)
00380              DO 23 I=1,3
00390                  DO 23 J=1,3
00400                      23 TATK(I,J)=DATK(I,J)
00410              N=N+1
00420              PX=PT(1)
00430              PY=PT(2)
00440              PZ=PT(3)
00450              DO 10 I=1,3
00460                  PT(I)=PT(I)+TATK(I,1)*VLEN
00470              10 CONTINUE
00480              GO TO 999
00490          END IF
00500          PSIS=D
00510          WRITE(6,*) ' '
00520          WRITE(6,*) '          ***** << PERSIS= ',PSIS,' >> *****'
00530          WRITE(6,*) '          ** X=',DT(1),' Y=',DT(2),' Z=',PT(3),' **'
00540 "
00550          STOP
00560          END

```

```

DATA SET NAME : AKH0112.TAC.FORT77(PCDIM)

00010 "
00020 "
00030 "          ++++++
00040 "          +
00050 "          + MAIN PROGRAM FOR ESTIMATION OF PURTURBED CHAIN +
00060 "          + DIMENSIONS. +
00070 "          +          1984/2/6 +
00080 "          +
00090 "          ++++++
00100 "
00110 DIMENSION IPHI(2000),IPSI(2000),P(2000),PP(2000),G(3),R(10000),COOD(3,0:1000), -
00120          NOT(99),GCOX(0:1000),GCOY(0:1000),GCOZ(0:1000),GCOX(0:1000),GCOY(0:1000), -
00130          GCOZ(0:1000),NSUB(1000),ICNT(2000),B(4,4,99)
00140 "
00150 "          ----- DATA READING -----
00160 "
00170          CALL CLOCK(ITIME)
00180          CALL CLOCKM(INTIME)
00190          READ(1,*) G(1),G(2),G(3)
00200          READ(1,*) THE,ETA,XI,OME,VLEN
00210          READ(1,*) IX,IY,ID
00220          READ(1,*) N
00230          READ(1,*) (P(I),I=1,N**2)
00240 "          ----- INITIALIZING FOR ENERGY MAP DATA -----
00250 "
00260          CALL INT1(IX,IY,ID,N,IPHI,IPSI,2000)
00270          CALL INT2(P,PP,IPHI,IPSI,NOD,0.0,2000,N)
00280 "
00290 "          ----- ESTIMATION OF PURTURBED CHAIN DIMENSIONS -----
00300 "
00310          WRITE(6,*) ' '
00320          WRITE(6,*) ' *****'
00330          WRITE(6,*) ' *****'
00340          WRITE(6,*) ' ** **'
00350          WRITE(6,*) ' **          CHARACTERIZATION OF CHAIN DIMENSIONS          **'
00360          WRITE(6,*) ' ** **'
00370          WRITE(6,*) ' *****'
00380          WRITE(6,*) ' *****'

```

```

00390 WRITE(6,*) ' '
00400 1000 READ(5,*) NS0,NS,NP,JLIM,RADI,NOC,ISTEP
00410 IF(NS0.EQ.0.AND.NS.EQ.0) STOP
00420 WRITE(6,*) ' '
00430 WRITE(6,*) ' *** PARAMETERS FOR SAMPLE ENRICHMENT METHOD ***'
00440 WRITE(6,*) ' '
00450 WRITE(6,*) ' S0=' ,NS0, ' S=' ,NS, ' P=' ,NP, ' JLIM=' ,JLIM, ' RADI=' ,RADI
00460 WRITE(6,*) ' '
00470 CALL ENRICH(PP,IPHI,IPSI,NOD,THE,ETA,XI,OME,VLEN, -
00480 RADI,G,NOC,NS0,NS,NP,JLIM, -
00490 R,CX,S2,R2,COOD,NOF,MDP, -
00500 B,NOT,ICNT,OCOY,OCOZ,GCOX,GCOY,GCOZ,NSUB, -
00510 2000,1000,1000,0.0)
00520 WRITE(6,*) ' '
00530 WRITE(6,*) ' **** RESULTS ***'
00540 WRITE(6,*) ' '
00550 WRITE(6,*) ' DP=' ,MDP, ' CX=' ,CX
00560 WRITE(6,*) ' '
00570 CALL HIST(R,MDP,NOC,ISTEP,10000)
00580 CALL CLOCK(MTIME)
00590 CALL CLOCKM(MMTIME)
00600 JTIME=MTIME-ITIME
00610 IM=JTIME/60
00620 IS=JTIME-IM*60
00630 WRITE(6,*) ' '
00640 WRITE(6,*) ' * CPU TIME = ' ,IM,'MIN. ' ,IS,'SEC. ( = ' ,MMTIME-IMTIME,'MSEC. )'
00650 WRITE(6,*) ' '
00660 GO TO 1000
00670 END

```

DATA SET NAME : AKH0112.TACLIB.FORT77(ENRICH)

```

00010 "
00020 " *****
00030 " *****
00040 " ** **
00050 " ** THIS SUBROUTINE WAS CONSTRUCTED FOR PURTURBED CHAIN **
00060 " ** GENERATION BY WALL AND ERPENBECK ENRICHMENT METHOD. **
00070 " ** **
00080 " ** COPY RIGHT : T. OKAHOTO , LABORATORY OF BIOPOLYMERS , **
00090 " ** FACULTY OF AGRICULTURE , KYOTO PREFECTURAL **
00100 " ** UNIVERSITY , KYOTO 606 , JAPAN **
00110 " ** **
00120 " *****
00130 " *****
00140 "
00150 SUBROUTINE ENRICH(PP,IPHI,IPSI,NOD,THE,ETA,XI,OME,VLEN,RADI,G,NOC,NS0,NS,NP,JLIM, -
00160 R,CX,S2,R2,COOD,NOF,MDP,B,NOT,ICNT,OCOY,OCOZ,GCOX,GCOY,GCOZ,NSUB, -
00170 KA,LA,MA,WRT)
00180 DIMENSION PP(0:KA),IPHI(KA),IPSI(KA),G(3),R(LA),COOD(3,0:MA),NOT(JLIM-1),OCOY(0:MA), -
00190 OCOZ(0:MA),GCOX(MA),GCOY(MA),GCOZ(MA),NSUB(MA),GCO(3),OCO(3),A(4,4),AA(4,4),E(4,4), -
00200 TK(3,3),MPSI(10),MPNT(10),ICNT(KA),B(4,4,JLIM-1)
00210 CHARACTER LOOP*8
00220 DATA E/1.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,1.0/
00230 "
00240 " ----- INITIALIZING -----
00250 "
00260 IF (WRT.GE.1.0) THEN
00270 WRITE(6,*) ' '
00280 WRITE(6,*) ' *****'
00290 WRITE(6,*) ' * * * * *'
00300 WRITE(6,*) ' * MESSAGES FROM SUBROUTINE "ENRICH" *'
00310 WRITE(6,*) ' * * * * *'
00320 WRITE(6,*) ' *****'
00330 WRITE(6,*) ' '
00340 WRITE(6,*) ' '
00350 ENDIF
00360 MDP=NS0+NS*(JLIM-1)
00370 NOF=0
00380 ICC=0
00390 ISEED=0
00400 OCOX(0)=0.0
00410 OCOY(0)=0.0
00420 OCOZ(0)=0.0
00430 OCOX(1)=VLEN
00440 OCOY(1)=0.0
00450 OCOZ(1)=0.0
00460 GCOX(1)=G(1)
00470 GCOY(1)=G(2)
00480 GCOZ(1)=G(3)
00490 AA(1,4)=VLEN
00500 AA(2,4)=0.0
00510 AA(3,4)=0.0
00520 AA(4,4)=1.0
00530 AA(4,1)=0.0
00540 AA(4,2)=0.0
00550 AA(4,3)=0.0
00560 R2=0.0
00570 S2=0.0
00580 DO 5 I=1,NOC
00590 R(I)=0.0
00600 5 CONTINUE
00610 DO 7 I=1,3
00611 DO 7 J=0,MDP
00620 COOD(I,J)=0.0
00630 7 CONTINUE
00640 "
00650 " ----- CHAIN SAMPLING LOOP -----
00660 "
00670 DO 10 WHILE(ICC.LT.NOC)

```

```

00680 "
00690 "
00700 "
00710 ICND=2
00720 DO 20 WHILE(ICND.EQ.2)
00730   DO 30 I=1,4
00740     DO 30 J=1,4
00750       A(I,J)=E(I,J)
00760     30 CONTINUE
00770     DO 40 I=2,MDP
00780       OCOX(I)=0.0
00790       OCOY(I)=0.0
00800       OCOZ(I)=0.0
00810       GCOX(I)=0.0
00820       GCOY(I)=0.0
00830       GCOZ(I)=0.0
00840     40 CONTINUE
00850     ICND=1
00860     IDP=2
00870     DO 50 WHILE(ICND.EQ.1.AND.IDP.LE.NS0)
00880       CALL RGEN(PP,IPHI,IPSI,ICNT,N0D,1,ISEED,MPHI,MPSI,MPNT,0.0,KA,10)
00890       PHI=FLOAT(MPHI(1))
00900       PSI=FLOAT(MPSI(1))
00910       CALL DCTMT(PHI,PSI,THE,ETA,XI,OME,TK)
00920       DO 60 I=1,3
00930         DO 60 J=1,3
00940           AA(I,J)=TK(I,J)
00950       60 CONTINUE
00960       CALL RESGEN(A,AA,VLEN,G,GCO,OCO,A)
00970       IF(IDP.NE.2) THEN
00980         CALL INTSCT(GCOX,GCOY,GCOZ,GCO(1),GCO(2),GCO(3),RADI,ICND,IDP-2,NSUB,N0D)
00990         IF(ICND.EQ.2) THEN
01000           NOF=NOF+1
01010           IF(WRT.GE.2) THEN
01020             WRITE(6,*) ' ----- '
01030             DO 62 I=1,N0D
01040               DIST=SQRT((GCO(1)-GCOX(NSUB(I)))**2+(GCO(2)-GCOY(NSUB(I)))**2+(GCO(3)-GCOZ(NSUB(I)))**2)
01050               WRITE(6,*) ' ',NSUB(I),' - ',IDP,' INTERSECTED. DISTANCE = ',DIST
01060               WRITE(6,*) ' NO. GCOX GCOY GCOZ'
01070               WRITE(6,1000) NSUB(I),GCOX(NSUB(I)),GCOY(NSUB(I)),GCOZ(NSUB(I))
01080               WRITE(6,1000) IDP,GCO(1),GCO(2),GCO(3)
01090               1000 FORMAT(3X,I4,2X,3E15.7)
01100             62 CONTINUE
01110             OCOX(IDP)=OCO(1)
01120             OCOY(IDP)=OCO(2)
01130             OCOZ(IDP)=OCO(3)
01140             GCOX(IDP)=GCO(1)
01150             GCOY(IDP)=GCO(2)
01160             GCOZ(IDP)=GCO(3)
01170             IF(WRT.GE.3.0) CALL PRINT(OCOX,OCOY,OCOZ,GCOX,GCOY,GCOZ,IDP,MA)
01180           ENDIF
01190         ENDIF
01200       ELSE
01210         ICND=1
01220       ENDIF
01230       OCOX(IDP)=OCO(1)
01240       OCOY(IDP)=OCO(2)
01250       OCOZ(IDP)=OCO(3)
01260       GCOX(IDP)=GCO(1)
01270       GCOY(IDP)=GCO(2)
01280       GCOZ(IDP)=GCO(3)
01290       IDP=IDP+1
01300     50 CONTINUE
01310   20 CONTINUE
01320 "
01330 "
01340 "
01350 JPNT=1
01360 LOOP='CONTINUE'
01370 DO 65 I=1,4
01380   DO 65 J=1,4
01390     B(I,J,1)=A(I,J)
01400 65 CONTINUE
01410 DO 70 I=1,JLIM
01420   NOT(I)=0
01430 70 CONTINUE
01440 "
01450 "
01460 "
01470 DO 80 WHILE(LOOP.NE.'END'.AND.NP.NE.0.AND.NS.NE.0.AND.ICC.LT.NOC)
01480   IF(NOT(JPNT).EQ.NP) THEN
01490     NOT(JPNT)=0
01500     JPNT=JPNT-1
01510     IF(JPNT.EQ.0) THEN
01520       LOOP='END'
01530     ELSE
01540       LOOP='CONTINUE'
01550     ENDIF
01560   ELSE
01570     "
01580     "
01590     "
01600     DO 85 I=1,4
01610       DO 85 J=1,4
01620         A(I,J)=B(I,J,JPNT)
01630       85 CONTINUE
01640     "
01650     "
01660     "
01670     IDP=NS0+NS*(JPNT-1)+1
01680     ICND=1
01690     DO 90 WHILE(ICND.EQ.1.AND.IDP.LE.NS0+NS*JPNT)

```

```

01700      CALL RGEN(PP,IPHI,IPSI,ICNT,NOD,1,ISEED,MPHI,MPSI,MPNT,0.0,KA,10)
01710      PHI=FLOAT(MPHI(1))
01720      PSI=FLOAT(MPSI(1))
01730      CALL DCTMT(PHI,PSI,THE,ETA,XI,OME,TK)
01740      DO 100 I=1,3
01750          DO 100 J=1,3
01760              AA(I,J)=TK(I,J)
01770      100 CONTINUE
01780      CALL RESGEN(A,AA,VLEN,G,GCO,OCO,A)
01790      CALL INTSCT(GCOX,GCOY,GCOZ,GCO(1),GCO(2),GCO(3),RADI,ICND,IDP-2,NSUB,NOO)
01800      OCOX(IDP)=OCO(1)
01810      OCOY(IDP)=OCO(2)
01820      OCOZ(IDP)=OCO(3)
01830      GCOX(IDP)=GCO(1)
01840      GCOY(IDP)=GCO(2)
01850      GCOZ(IDP)=GCO(3)
01860      IDP=IDP+1
01870      90 CONTINUE
01880 "
01890 " ----- AFTER INTERSECTION -----
01900 "
01910      IF(ICND.EQ.2) THEN
01920          IF(WRT.GE.2.0) THEN
01930              WRITE(6,*) ' -----
01940                  DO 105 I=1,NOO
01950                      DIST=SQRT((GCO(1)-GCOX(NSUB(I)))**2+(GCO(2)-GCOY(NSUB(I)))**2+(GCO(3)-GCOZ(NSUB(I)))**2)
01960                      WRITE(6,*) ' ',NSUB(I),' - ',IDP-1,' INTERSECTED. DISTANCE = ',DIST
01970                      WRITE(6,*) ' NO. GCOX GCOY GCOZ'
01980                      WRITE(6,1000) NSUB(I),GCOX(NSUB(I)),GCOY(NSUB(I)),GCOZ(NSUB(I))
01990                      WRITE(6,1000) IDP-1,GCO(1),GCO(2),GCO(3)
02000                  105 CONTINUE
02010                  IF(WRT.GE.3.0) CALL PRINT(OCOX,OCOY,OCOZ,GCOX,GCOY,GCOZ,IDP-1,MA)
02020              ENDIF
02030              NOF=NOF+1
02040              NOT(JPNT)=NOT(JPNT)+1
02050              LOOP='CONTINUE'
02060 "
02070 " ----- AFTER NO INTERSECTION -----
02080 "
02090      ELSE
02100          NOT(JPNT)=NOT(JPNT)+1
02110          JPNT=JPNT+1
02120          IF(JPNT.EQ.JLIM) THEN
02130              ICC=ICC+1
02140              IF(WRT.GE.2.0) THEN
02150                  WRITE(6,*) ' -----
02160                  WRITE(6,*) ' * CHAIN GENERATION HAS BEEN ACCOMPLISHED. DP=',MDP
02170              ENDIF
02180              IF(WRT.GE.3.0) CALL PRINT(OCOX,OCOY,OCOZ,GCOX,GCOY,GCOZ,MDP,MA)
02190 "
02200 " ----- ACCUMULATE QUANTITIES -----
02210 "
02220          R2=R2+OCOX(MDP)**2+OCOY(MDP)**2+OCOZ(MDP)**2
02230          R(ICC)=SQRT(OCOX(MDP)**2+OCOY(MDP)**2+OCOZ(MDP)**2)
02240          S2=S2+SGR(GCOX,GCOY,GCOZ,MDP,MA)
02250          DO 130 I=0,MDP
02260              COOD(1,I)=COOD(1,I)+OCOX(I)
02270              COOD(2,I)=COOD(2,I)+OCOY(I)
02280              COOD(3,I)=COOD(3,I)+OCOZ(I)
02290          130 CONTINUE
02300 "
02310 " -----
02320 "
02330          JPNT=JPNT-1
02340          LOOP='CONTINUE'
02350      ELSE
02360          DO 140 I=1,4
02370              DO 140 J=1,4
02380                  B(I,J,JPNT)=A(I,J)
02390          140 CONTINUE
02400          LOOP='CONTINUE'
02410      ENDIF
02420      ENDIF
02430      ENDIF
02440      80 CONTINUE
02450      LOOP='CONTINUE'
02460      IF(NP.EQ.0.OR.NS.EQ.0) THEN
02470          ICC=ICC+1
02480          IF(WRT.GE.2.0) THEN
02490              WRITE(6,*) ' -----
02500              WRITE(6,*) ' * CHAIN GENERATION HAS BEEN ACCOMPLISHED. DP=',MDP
02510          ENDIF
02520          IF(WRT.GE.3.0) CALL PRINT(OCOX,OCOY,OCOZ,GCOX,GCOY,GCOZ,MDP,MA)
02530 "
02540 " ----- ACCUMULATE QUANTITIES -----
02550 "
02560          R2=R2+OCOX(MDP)**2+OCOY(MDP)**2+OCOZ(MDP)**2
02570          R(ICC)=SQRT(OCOX(MDP)**2+OCOY(MDP)**2+OCOZ(MDP)**2)
02580          S2=S2+SGR(GCOX,GCOY,GCOZ,MDP,MA)
02590          DO 145 I=0,MDP
02600              COOD(1,I)=COOD(1,I)+OCOX(I)
02610              COOD(2,I)=COOD(2,I)+OCOY(I)
02620              COOD(3,I)=COOD(3,I)+OCOZ(I)
02630          145 CONTINUE
02640          ENDIF
02650      10 CONTINUE
02660 "
02670 " ----- ESTIMATION OF CHAIN DIMENSIONS -----
02680 "
02690      R2=R2/NOC
02700      S2=S2/NOC
02710      DO 150 I=1,3
02720          DO 150 J=0,MDP

```

```

02730          COOD(I,J)=COOD(I,J)/NOC
02740          150 CONTINUE
02750          CX=R2/MDP/VLEN**2
02760          PA=FLOAT(NOC)/(NOC+NOF)
02770 "
02780 " ----- WRITEING OF RESULTS -----
02790 "
02800          IF(WRT.GE.1.0) THEN
02810              WRITE(6,*) ' '
02820              WRITE(6,*) ' ++++++ RESULTS IN THE SUBROUTINE , "ENRICH" ++++++'
02830              WRITE(6,*) ' '
02840              WRITE(6,*) ' DP=',MDP,' <S2>=',S2,' <R2>=',R2,' CX=',CX
02850              WRITE(6,*) ' '
02860              WRITE(6,*) ' PA=',PA,' NOC=',NOC,' NOF=',NOF
02870              WRITE(6,*) ' '
02880              WRITE(6,*) ' ----- PERSISTENCE VECTOR -----'
02890              WRITE(6,*) ' '
02900              WRITE(6,*) '          NO.          X          Y          Z'
02910              WRITE(6,2000) (J,(COOD(I,J),I=1,3),J=1,MDP)
02920              2000 FORMAT(5X,I4,2X,3E15.7)
02930          ENDIF
02940          RETURN
02950          END

```