

投稿論文 (和文)
PAPERS

岩盤不連続面構造の立体幾何形状のコンピュータ処理に用いるダイレクテッド・ボディについて

池川洋二郎*

ここでは不連続面の方向性など幾何情報を入力データとする岩盤の不連続面構造のコンピュータ処理手法の紹介を行う。3次元空間に分布する多数の不連続面により分割される凹形状を含む3次元ブロックの自動認識、さらに体積、表面積などの計算を4面体への細分割を行わずに可能としている。

Key Words : rock mass structure, discontinuity, three-dimensional, geometrical analysis

1. はじめに

岩盤中では多数の不連続面が互いに交差し、岩盤を多様な形状や大きさの岩石ブロックに分割している。よって、岩盤は多数のブロックが隙間なく積み重なった構造と考えられる。さらに、これらの不連続面は水みちとなるネットワークを形成している。以下では、この岩盤中の不連続面により作られる各岩石ブロックや水みちの幾何形状を簡単に岩盤の“幾何構造”と呼ぶ。岩盤の不連続性は、点や線ではなく、3次元の面の存在により複雑な幾何構造を示し、力学・水理学的に岩盤挙動を大きく支配している。

これまで不連続面の方向性、節理間隔、トレース長などについて確立統計的な処理手法が提案されている。さらに3次元的に幾何構造を採取し、数値シミュレーションに結び付けるためには、幾何構造のコンピュータ処理を行う必要がある。ここでは幾何構造の解析を簡単に“幾何解析”と呼び、幾何解析には各ブロックや水みちの幾何形状の同定・記述など、特に岩盤を不連続体として解析するために必要な幾何データの解析と作成を含む。ここでは岩盤の幾何解析を一般的に行うため、著者らによる立体形状の処理理論であるダイレクテッド・ボディ(directed body)^{3),4)}と、これに基づくコンピュータ・システムの紹介を行う。ここでは亀裂、節理、断層、分離面などを総称して不連続面と呼ぶ。

これまでダイレクテッド・ボディ理論³⁾では岩盤中に形成される3次元ブロックの凹凸形状に関わらない、立体形状の同定を可能としいる。しかし、実際のコンピュータによる同定^{2),3),5),7)}において、幾何構造を記述するデータ・ストラクチャーに問題があり、記述できるブロック数、さらに各ブロックの面、辺、頂点数は各々に配分された配列(メモリー)の大きさに制限を受け、幾何形状の同定処理が行えない。例えば、同定されようとするブ

ロックの面の数が15面であるのに対し、プログラム上で最大10面分の配列しか与えられていない場合、プログラムは誤動作により停止し、このブロックは同定されない。また、岩盤の幾何構造の同定過程では、4節に詳細する次のような特有な段階がある。不連続面を模擬する多数の面をインプットし、それらの面の交点の計算、さらに交点を結んで辺を作り、辺を結んで面、面で形成されるブロックを同定していく。このような同定過程では、事前に分割されるブロック数、各々のブロックの面・辺・頂点数を知ることは無理である。従って、プログラムがコンピュータ上で同定処理を行っている時に、配列の大きさを自動的に変える必要がある。

ここで紹介を行うデータ・ストラクチャーは、プログラムがコンピュータ上で動いている時に、同定されつつある各ブロックに最適な大きさの配列を与え、任意の幾何構造の同定と、記述を可能とした。また、岩石ブロックの途中で止まった不連続面の考慮を行うことを可能とした。これにより不連続面の発生の順序を考慮した解析を可能とできるものと考えられる。水みちの幾何構造は、ブロックの幾何構造の同定後、水みちを体積がゼロである1つのダイレクテッド・ボディとして記述を行うことができる。

2. 岩盤工学における幾何解析

岩盤が示す3次元の幾何構造を知るには膨大な調査とデータが必要であり、現状の技術による完全な調査は不可能である。従って、確立統計手法での基本手法であるサンプル調査により、母集団の推定を推定する方法を取らざるをえない。ここでサンプルは掘削・露頭面やボーリングなどにより計測される方向性、間隔、トレース長である。以上のように、岩盤を不連続体として考慮していくには、その幾何構造に重点をおく必要があり、計測された幾何情報と、数値解析の間を埋めるコンピュータによる立体形状の処理を含む一貫したシステムによる幾何構造の解析の確立が必要であると考える。ここで紹

* 正会員 Ph.D (財)電力中央研究所
(〒270-11 千葉県我孫子市我孫子1646)

不連続面の幾何情報の計測

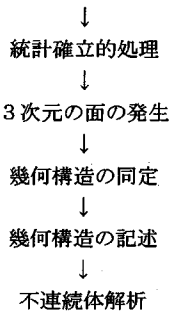


図-2.1 幾何解析の流れ
(ここでは、岩盤中の不連続面により作られる各岩石ブロックや水みちの幾何形状を岩盤の幾何構造と呼ぶ。)

表-3.1 ダイレクテッド・ボディ理論の特徴

	従来 ^{2), 3), 5), 7)} の方法	ダイレクテッド・ボディ理論
入力可能な不連続面形状	円盤あるいは、注目する領域を完全に分割する平面。	凹凸形状に関わらない任意の多角形を入力可能曲面は3角形に分割することによって入力。
立体形状の同定処理能力	凸型ブロックに限定される。凹型ブロックは数個の凸型ブロックの集合として記述する。	凹凸形状に関わらない任意の多面体を処理し、ブロック中で止まった面の処理も可能。
データストラクチャー	静的メモリー配分を用い、配列の最大数が存在し、プログラムの実行中にその最大値の変更が行えない。	動的メモリー配分を用いて、任意の幾何構造の記述が可能。
体積・重心・表面積等の計算	4面体への分割が必要。	凹凸形状に関わらず一般的に計算。

介を行うシステムは、図-2.1に示すように計測データと数値解析の間に位置する幾何解析を行うのに必要なシステムであると考える。

3. ダイレクテッド・ボディ理論

ダイレクテッド・ボディ理論では表-3.1に示すような特徴を持ち、3次元ブロックの凹凸形状に関わらない岩盤不連続面構造のコンピュータ処理を可能としている。ここで凸型形状は領域内の任意の2点を結ぶ線分が完全にその領域に含まれる形状と定義され、凹型形状は完全に含まれない線分が存在する形状である。

このダイレクテッド・ボディを用いたシステムの特徴は適用範囲を凸型ブロックから凹型ブロックまで拡張を行っていることであり、さらに4つに分けられる。入力可能な不連続面の形状は円盤、あるいは正多角形(凸形状)であったが、ここでは穴を持つ任意の形状の多角形まで入力可能としている。幾何構造の同定では凸型ブロックだけで、凹型ブロックは同定された凸型ブロックの集合として記述されていたが、凹型ブロックも凸型ブロック同じ方法で同定が行える。データ・ストラクチャーは、動的メモリー配列を用いることにより同定しようとする幾何構造に適切な配列に自動的に変えることを可能とした。体積や表面積などの数値解析に必要な諸量については、ブロックを4面体ブロックに分割する必要があったが、ここでは形状に関らず分割を行わずに計算を行う。

(1) ダイレクテッド・ボディ

図-3.1.1はダイレクテッド・ボディ(DB)を模式的に示す。DBではブロックを形成する面の法線ベクトルである外向き面ベクトル(external face vector) f^{ext} に、面の大きさを持たせることにより、ブロックの表面積や体積の計算、立体形状の同定処理などを凹凸形状に関らず行うことを可能としている。DBの面はダイレクテッ

ド・フェイス(directed face)DFと呼び、図-3.1.2に示すように分割され面番号を持つ。さらに、頂点 v_i は頂点番号 i を持ち、3次元座標が記述される。DFは外向き面ベクトルに対して反時計回り(図-3.1.3参照)に頂点番号を記述して表現する。DFの各辺は辺ベクトル e として表され、 e と f^{ext} に垂直な外向きと内向きの辺ベクトルをそれぞれ e^{out} と e^{in} と表す。岩盤の幾何構造は、多数のDBが積み重なったものとして記述を行う。

(2) データ・ストラクチャー

コンピュータ・メモリーの利用において、プログラムの実行中に配列の大きさ(要素数)を変化させるのできる動的メモリー配分(dynamic memory allocation)と、ソース・コードで配列の大きさ宣言する静的メモリー配分の2つの方法がある。土木の分野で開発されてきている多くの解析プログラムでは、後者の方法が用いられ、プログラムを動かしているとき、配列の大きさを変える事はできない。従って、配列の大きさを変えるためには、ソースプログラムを書き変える必要がある。

岩盤を不連続面で分割される岩石ブロックの集合体として記述を行う場合、ブロック数、また各ブロックの面・辺・頂点の数は不規則で、静的メモリー配分を用いた場合、十分に大きな配列が必要となり、使用されないメモリー領域が多くなる。例えば、最大20面体を記述できる配列を用いると、4面体を記述する場合、16面分の配列は使われない。また、コンピュータ・メモリーの容量には制限があり、使われないメモリーを減少させることにより、より複雑な幾何構造を扱うことができる。

そこでダイレクテッド・ボディ理論では動的メモリー配分を用い、配列の大きさの制御・管理を行うVLS(Variable Length String)を複合的に組み合わせることにより作成したデータ・ストラクチャーを用いた。VLSは図-3.2.1に示すように、ヘッダー(header)と配列(data elements)の2つの部分から成る。ヘッダーは配列中の最大要素数(dimlen)と、実際の記述に用いられている要素数(currlen)、各要素の大きさ(esize)、配分され

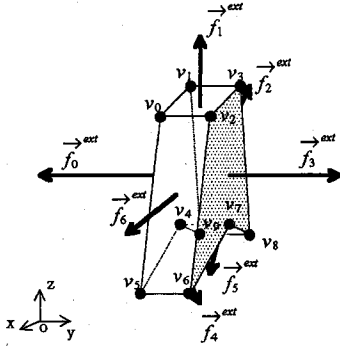


図-3.1.1 ダイレクテッド・ボディ

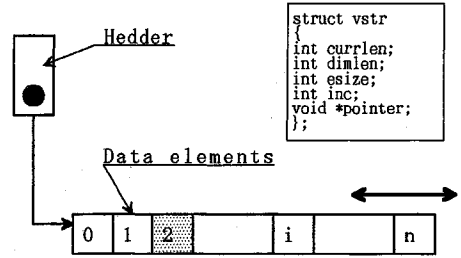


図-3.2.1 Variable length record

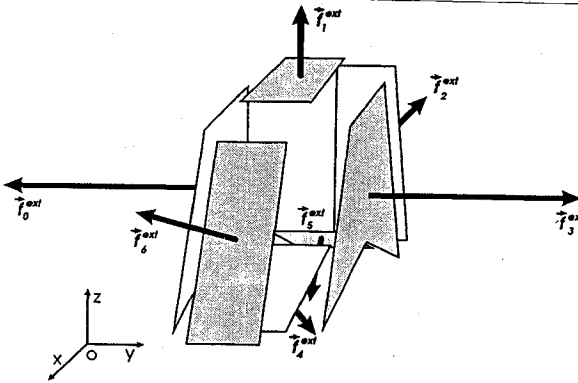


図-3.1.2 ダイレクテッド・ボディの表面分割

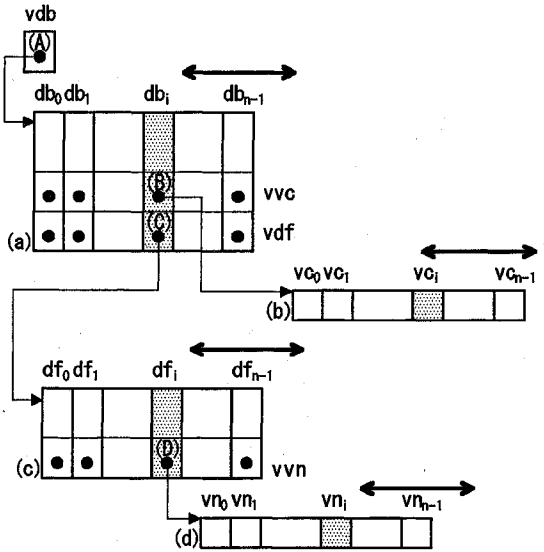


図-3.2.2 ダイレクテッド・ボディのデータ・ストラクチャー

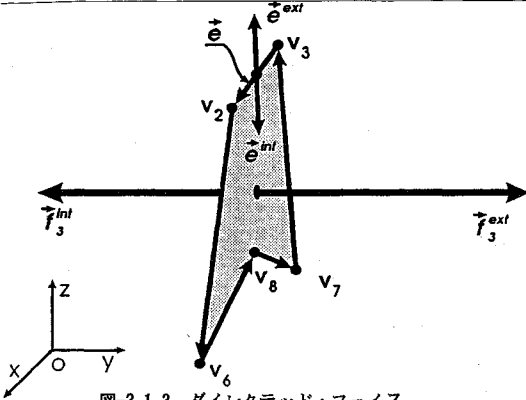


図-3.1.3 ダイレクテッド・フェイス

ている要素数が足りないときの増分(inc), 配列のメモリー上のアドレスを常に記録するポインター(pointer)の5つの情報を持っている。要素数がなくなった場合, VLSはコンピュータ・メモリー上の十分な大きさの空き領域を確保し, そのアドレスをポインターで記録する。図-3.2.1で●から始まる矢印はポインターを模式的に表し, 両端に矢印の付いた記号は配列中の要素数を自動的

に変化できることを示す。各要素の大きさはバイト(byte)数で表す。例えば, 整数は2バイト, 単精度・倍精度の実数はそれぞれ4・8バイトである。さらに, C言語で用いられるデータ型の宣言(struct)を用いて, 文字, 整数, 実数を任意の組み合わせて, 各要素中に記述するデータを宣言する。(図-3.2.3~5参照)ここで/**/はコメント文, vstrはVLSの宣言を表す。

図-3.2.2はDBのデータ・ストラクチャーを模式的に示し, VLSを複合的に組み合わせたものである。図中の(A)はヘッダーで(a)は配列に相当し, また(a)中の1つの要素db_i中の(B)(C)はそれぞれ(b)(c)部分のヘッダーに相当する。各ブロックの幾何形状の情報は(A)と(a)で構成されるVLSの各要素で記述される。(B)と(b), (C)と(c), (D)と(d)で構成されるVLSで, 頂点座標, ダイレDFの諸量, 面を構成する頂点番号が記述される。

```
struct dirbody
{
  vstr vdf; /* directed faces */
  vstr vvc; /* vertices */
  pt cent; /* gravity center */
  double volume;
}
```

図-3.2.3 ダイレクテッド・ボディのデータ・ストラクチャー。
(図-3.2.2中の(a)の各要素で記述される。)

```
struct pt
{
  double x, y, z;
};
```

図-3.2.4 頂点座標のデータ・ストラクチャー
(図-3.2.2中の(b)で記述される。)

```
struct dirface
{
  pt cent; /* centroid of the face. */
  pt ext; /* external face vector */
  double dadius;
  vstr vvn /* VLS */
}
```

図-3.2.5 ダイレクテッド・フェイスのデータ
ストラクチャー (図-3.2.2の(d)で記述される)

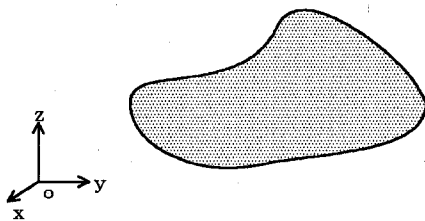


図-3.3.1 空間の閉じた領域

図-3.2.4と図-3.2.5に、頂点座標とDFの各要素中のデータ・ストラクチャーを示す。(d)の各要素にはDFの頂点番号が記述される。

このデータ・ストラクチャーでは、実際にソース・コード上で宣言されるメモリー(静的メモリー配分)は図-3.2.2の(A)のヘッダーだけであり、残りはすべて動的メモリー配分を用いて、プログラム実行中に適時配分される。

(3) 数値解析のための諸量の計算

数値解析への入力データとしては、物性としてブロッ

クの体積、表面積なども必要である。ダイレクテッド・ボディでは、それらの諸量をブロックの形状に関らず同一のアルゴリズムで計算できるように定式化^{3),4)}を行って

a) 外向き面ベクトルと面の面積

閉じた境界(図-3.3.1)は直交座標系において、媒介変数を用いて、以下のようなベクトル関数として記述できる。

$$\vec{v}(t) = v_x(t)\vec{i} + v_y(t)\vec{j} + v_z(t)\vec{k} \dots\dots\dots (1)$$

$$0 \leq t \leq 1, \vec{v}(0) = \vec{v}(1)$$

注目する面が完全に平面に含まれる場合、外向き面ベクトルは、ベクトルの外積を用い、

$$\vec{f}^{ext} = \frac{1}{2} \int (\vec{v}(t) \times d\vec{s}) \dots\dots\dots (2)$$

で計算される。ここで、

$$d\vec{s} = \frac{dv_x(t)}{dt}\vec{i} + \frac{dv_y(t)}{dt}\vec{j} + \frac{dv_z(t)}{dt}\vec{k} \dots\dots\dots (3)$$

さらに、DFの境界は頂点を結ぶ線分で記述されるので、

$$\vec{f}^{ext} = \frac{1}{2} \sum_{i=0}^{n_f-1} (\vec{v}_{i+1} - \vec{p}) \times (\vec{v}_i - \vec{p}) \dots\dots\dots (4)$$

となる。ここで、 \vec{v}_i は頂点の位置ベクトル、 \vec{p} は注目するDFを含む平面上の任意点の位置ベクトルである。内向き面ベクトルは、

$$\vec{f}^{int} = -\vec{f}^{ext} \dots\dots\dots (5)$$

であり、DFの面積は、

$$A = |\vec{f}^{ext}| = |\vec{f}^{int}| \dots\dots\dots (6)$$

のように外向き・内向き面法線ベクトルの大きさである。

b) 辺ベクトル

DFの辺は、図-3.1.3に示すように辺ベクトル(edge vector)と、外向き・内向き辺ベクトルの3種類のベクトルで考慮できる。これらの辺ベクトルは2つの面が作る角度や、トポロジー的な線分の連結を表すのに用いる。

$$\vec{e}_i = \vec{v}_{i+1} - \vec{v}_i \quad i = 0, 1, \dots, n_f - 1 \dots\dots\dots (7)$$

ここで n_f は面を作る辺あるいは頂点の数で、式(7)において $i = n_f - 1$ のとき、 $i + 1 = 0$ である。外向き、内向き法線辺ベクトルは外向き面法線ベクトルを用いると、

$$\vec{e}_i^{ext} = \vec{e}_i \times \vec{f}^{ext} \dots\dots\dots (8)$$

$$\vec{e}_i^{int} = \vec{f}^{ext} \times \vec{e}_i \dots\dots\dots (9)$$

である。

c) **ダイレクテッド・ボディの体積**

ブロックの体積はベクトルの3重積をもとに外向き面ベクトルを用いて、以下のように求められる。

$$V = \frac{1}{3} \sum_{i=0}^{N-1} (f_i^{ext} \cdot \vec{p}) \dots\dots\dots (10)$$

ここで \vec{p} は任意の点の位置ベクトルを表す。

4. 立体の幾何構造の形状同定処理

岩盤幾何構造の同定では、各不連続面の3次元形状の入力を行い、それらの面によって分割されるブロック形状を同定する方法が、不連続面の発生過程を考慮する上で適当であると考えられる。つまり不連続面群の発生の順序によっては、既存の不連続面上で進展の止まったものが存在し、これらの過程を一般的に処理するには、コンピュータ処理においても各不連続面の立体形状を順次入力し、分割されるブロック形状を同定する方法をとる必要がある。

これまで不連続面を円盤と仮定し、位置、半径、方向を確立統計的に発生させている。以下では、不連続面の発生方法をより一般的に行えることを考慮してシステムの構築を行っており、具体的な不連続面の発生方法については、このシステムを用いて研究を進めていくものとする。

(1) **解析領域の入力**

立体の解析領域は、ダイレクテッド・ボディとして記述を行う。図-4.1.1では六面体が解析領域として示す。

(2) **不連続面の入力**

不連続面は任意の多角形のDFとして入力を行う。具体的な形状は、今後の研究を必要とするものである。

また、図-4.2.1に示すような曲面は有限要素法における形状関数として用いられているラグランジュ関数を用いて表すことが、有効な方法である。これらの曲面は直接、同定過程に入力できないので、小さな平面に分割することによって入力する。システムでは、(11)式中の u と v をランダム、あるいは等分割することにより、面内に点を発生させ、それらを結んで三角形に分割する2つの方法をとることができる。以上の三角形分割では、4節の方法を用いて各三角形の同定を行う。

$$\vec{r}(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 c_{ij} v^i u^j, \quad 0 \leq u, v \leq 1 \dots\dots\dots (11)$$

図-4.1.1には、簡単な例として2組の平行な4角形の面の入力を行っている。

(3) **交点の同定**

平面の交点には図-4.3.1に示す3面の交点と、図-4.3.2に示す2面(面と面の辺)の交点の2種類がある。3面の交点は各面を含む平面の方程式を連立して解き、その交点が同時に3面の中に存在すればブロックの

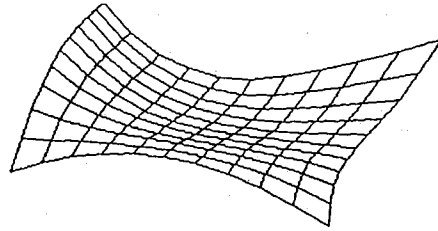


図-4.2.1 曲面の例

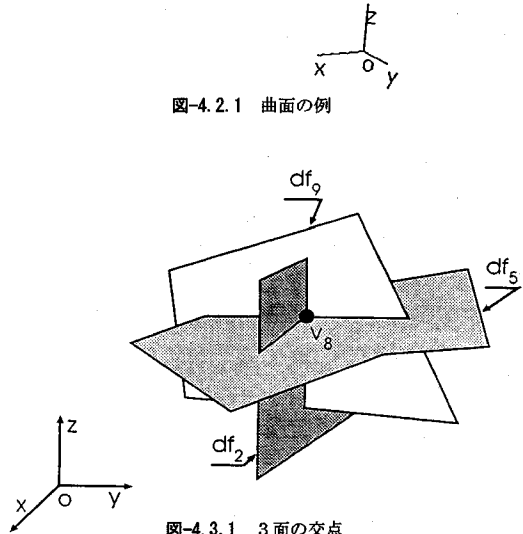


図-4.3.1 3面の交点

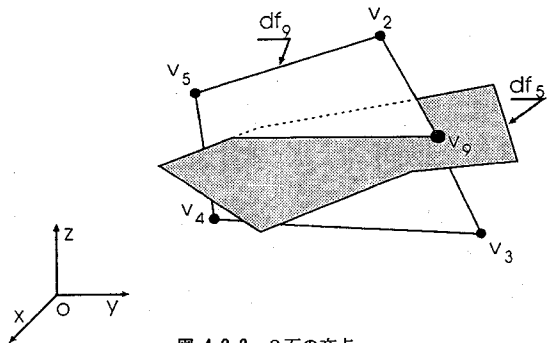


図-4.3.2 2面の交点

頂点となる。また、2面の交点は面を含む平面の方程式と辺を含む直線の方程式を連立して解き、その交点が2面上に存在すれば交点となる。これまでは2面の交点を考慮していなかったが、ここで考慮することによりブロック内で止まっている面を同定することを可能としている。

(4) **辺の同定**

一つの面に注目すると面は図-4.4.1の模式的に示すように多数の面と交差している。辺ベクトルは前節で同定した隣接する2点を始点と終点を持つベクトルとして表

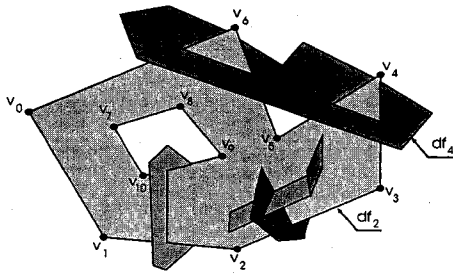


図-4.4.1 交差する面

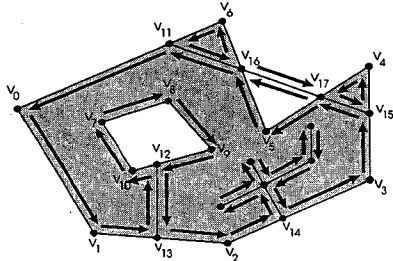


図-4.4.2 辺・ベクトル

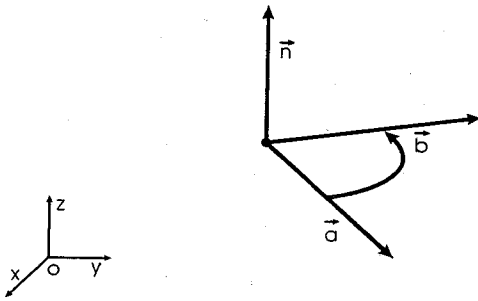


図-4.5.1 3次元における右手回りの角度

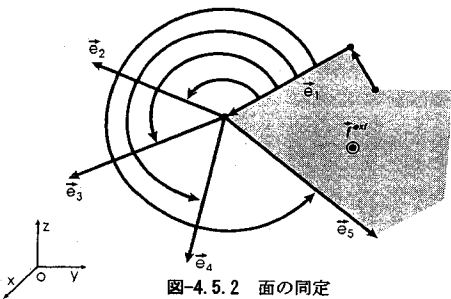


図-4.5.2 面の同定

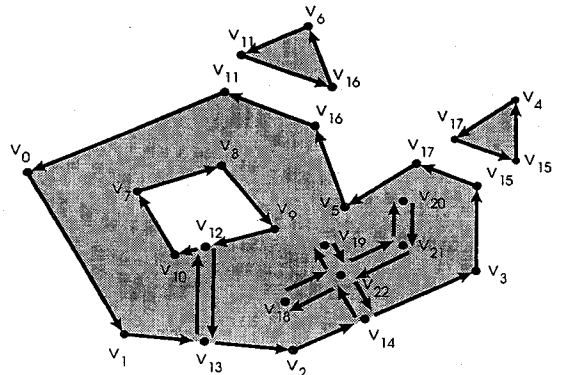


図-4.5.3 同定された面

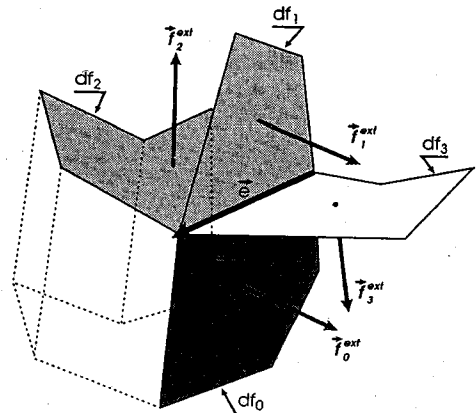


図-4.6.1 ブロックの同定

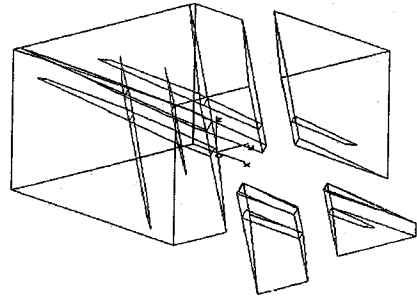


図-4.6.2 ブロック・システムへの分割

す。ここでは面の同定のため、他の面との交差によって作られる辺は、互いに反対を向く2つの辺ベクトルとして表す。図-4.4.1の交差の例から得られる辺ベクトルを図-4.4.2に示す。凹形状の面の場合、隣接する2点間の辺ベクトルを考慮すると、面上に存在しない辺を辺ベクトルとして記述することがあるため、面内に含まれない辺ベクトルは消去する。例えば、図-4.4.2では v_{16} と v_{17} の間の2つの反対向きの辺ベクトルはもとの面に含まれて

いないので消去しなければならない。

(5) 面の同定

面の同定では、まず入力した各々面について順次行う。1つの面は図-4.4.2のように、他の面との交差により分割されており、辺の同定で得られた辺ベクトルの中から注目する面上にある辺ベクトルをすべて選ぶ。注目する面の外向き面ベクトルを基準にして、ある辺ベクトルから順次、右手回りの最大の角度を持つ辺ベクトルをつないでいき、閉じた面を同定していく。右手回りの角度は図-4.5.1に示すように、 \vec{a} に対し、 \vec{a} から \vec{b} への角度とする。ある辺ベクトルに注目すると、このベクトルの終

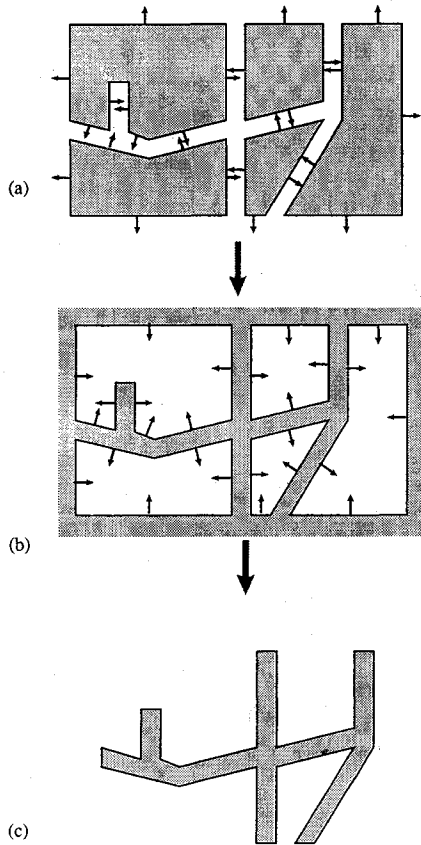


図-4.7.1 ブロック・システムから水みちへの変換

点を始点とする辺ベクトルが、まず候補として挙げられる。この中から、図-4.5.2に示すように最大の右手回りの角度を作る辺ベクトルを同定しようとする面を形成する辺として記述する。例えば図-4.5.2では、辺ベクトル \vec{e}_1 に注目したとき、この辺ベクトルの終点を始点を持つ辺ベクトルが4つあり、右手回りの最大角を作る \vec{e}_2 が実際の辺である。さらにこの操作を繰り返す、得られた辺ベクトルがサイクリックに結ばれたとき、これを1つのDFとして記述する。図-4.5.3は図-4.4.2で得られた辺ベクトルから同定されるDFを模式的に示し、面内の途中で止まった面、穴の空いた面なども以上の規則を用いれば同定できることを示す。

(6) ブロックの同定

ブロックの同定では、図-4.6.1に示すようにある面 f_{int}^{ext} の1つの辺ベクトル \vec{e}_1 に注目し、この負の辺ベクトル $-\vec{e}_1$ を持つ面を同定しようとするブロックの面を形成する面の候補として探します。次に注目した辺ベクトルに対して、内向き辺ベクトルが右手回りの最大角度を作る面がブロックを形成する。図-4.6.1の模式的な例では、面の辺ベクトル \vec{e}_1 に注目したとき、3つの面が候補として挙げられ、実際の面は f_{int}^{ext} となっています。この操作を繰り返す、閉じたブロックを形成したとき1つのブロックとして記述する。

図-3.6.2には、このシステムによって六面体が2群の四

角形によって分割される4つのブロックを示し、ブロックの途中で分割が止まっているもの、さらにそれらがブロックの中で交差している状態も同定されていることがわかる。また、他の不連続面上で進展の止まった不連続面を表現するには、ブロック内に残っている面を消去することによって行うことができる。ここで、ブロック内に残っている面は、互いに反対向きの外向き面ベクトルを持つ2つのDFとして記述されている。

(7) 水みちの同定

DFの頂点の記述を反対に記述すると、外向きフェイスベクトルの方向は反対方向となる。従って、岩盤中に含まれ、解析領域の境界上にない同定されたすべての面を持つ1つのDBと記述することにより、水みちの幾何構造が記述される。図-4.7.1(a)の模式図に示すように、入力した不連続面により分割されるブロックの同定を行い、さらに(b)のようにすべてのブロックのダイレクテッド・フェイスを反転し、境界上の面を消去し、残ったすべての面により形成されるブロックを記述することにより、水みちが1つのDBとして記述できる。

5. 最後に

この論文では岩盤中の不連続面により分割される、各岩石ブロックの形状などの幾何構造を、一般的にコンピュータ処理を行うための立体形状処理理論であるダイレクテッド・ボディと、動的メモリー配分を用いたデータ・ストラクチャーを用いることにより形状処理システムの開発を行うことにより、岩盤の幾何構造の解析を3次的に行うことを可能とした。

参考文献

- 1) Cundall, P.A. : Formulation of Three-dimensional Distinct Element Model - Part I. A Scheme to Detect and Represent Contacts in System Composed of Many Polyhedral Blocks. Int. J. Rock Mech. Min. Sci. & Geomech. Abstr. Vol.25, No.3, pp107-116, 1988.
- 2) Heliot, D. : Generating a Blocky Rock Mass. Int. J. Rock Mech. Sci. Min & Geomech. Abstr. Vol.25, No.3, pp.127-138, 1988.
- 3) Ikegawa, Y. and Hudson, J.A. : A Novel Automatic Identification System for Three-dimensional Multi-block Systems. Engineering Computings, Vol.9 pp169-179, 1992.
- 4) Ikegawa, Y. : Three-dimensional geometrical analysis of rock mass structure. Ph.D thesis, Univ. of London, 1992.
- 5) Lin, D., Fairhurst, C. and Starfield, A.M. : Geometrical Identification of Three-dimensional Rock Block System Using Topological Techniques. Int. J. Rock Mech. Sci. Min & Geomech. Abstr. Vol.25, No.3, pp.139-147, 1988.

- 6) Mosich, D., Shamas, N. and Flamig, B. : Advanced Turbo C Programmer's Guide. Wiley, 1988.
- 7) Warburton, P. M. : Application of a New Computer Model for Reconstructing Blocky Block Geometry,

Analysis Single Block Stability and Identifying Keystones. Proc. 5th Int. Congr. on Rock Mech. ppF225-230. Melbourne, 1983.

(1992. 11. 10受付)

A DIRECTED BODY THEORY FOR COMPUTATIONAL MANAGEMENT OF SOLID GEOMETRY OF ROCK MASS STRUCTURE

Yojiro IKEGAWA

Rock masses are fractured structure and consist of rock blocks in three-dimensional space : the nature and distribution of the fractures is termed "rock mass structure". Although the 3-D geometry of the structure is one of the primary parameters that dominates rock mass behavior, the subject has not been studied adequately. This paper introduces a new computational 3-D geometry management method for rock mass structure, i.e., directed body theory. A computational system based on the directed body concept has been developed to the state where it can handle the terminated discontinuities in rock blocks and also non-convex blocks, together with calculation of quantities, such as volumes and surface area. Thus, the system successfully produces a geometrical database of a representative 3-D rock mass structure in a generic manner.
