

双対原理を利用したソリッドモデリング

6AD-2

早稲田大学 ○津金尚志, 吉田典正, 土井淳, 金俊赫, 山内俊哉, 山口富士夫

1. はじめに

射影幾何学において, ある命題(あるいは概念)中のすべての「点」および「面」という語をそれぞれ「面」および「点」に変えて得られる命題を, もとの命題の空間双対命題という[1]. もとの命題が成り立てばその双対にあたる命題も常に成り立つという性質を双対原理(Principle of Duality)という. 本稿は, 完全4次元同次処理[2]による境界表現(B-rep)のソリッドモデラに, 双対原理を利用した一例を示したものである.

双対原理を定理・証明・演算・アルゴリズムに適用することは大変有効である. 双対原理をアルゴリズムに適用する場合の主な利点は,

- 1つのプログラムが互いに双対な2つの用途に使用できること
- 双対変換により, 変換前よりアルゴリズムが効率的になる場合があること[3]

である. また1つめの利点に関しては, その片方の用途について正確な動作が確認されれば(デバッグをおこなえば), その双対についても正確な動作が保証されるという更なる利点がある. ここではこの1つめの利点をソリッドモデリングに積極的に利用していく. この利点は双対な幾何計算などにも利用できるが, 本稿では特に立体の位相を記述するデータ構造と, その操作をおこなうオペレータについて述べる.

2. 双対なデータ構造

2.1. Quad Edge データ構造

形状の位相を記述するデータ構造に双対原理を利用したものとして Quad Edge データ構造[4]がある(図1).

図1において十字形は稜線を表わし, それに接続する実線と点線がそれぞれ稜線の両端点, 稜線の両側の面分を表わす. このデータ構造では頂点と面分が抽象

化されており, 図1の実線のループと点線のループをそれぞれ, 頂点・面分として解釈するか(図2左), 面分・頂点として解釈するか(図2右)によって, このデータ構造は双対な2つの形状を表現できる.

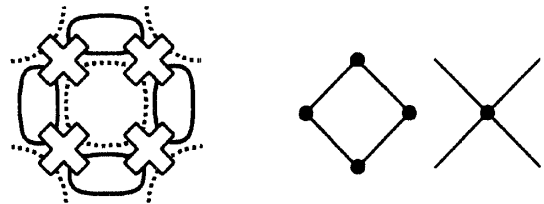


図1. Quad Edge データ構造 図2. 双対な2つの形状

2.2. Quarter Edge データ構造での双対原理の利用

この概念をソリッドモデラの Quarter Edge データ構造とオイラーオペレータ[5]に応用する. Quarter Edge データ構造は, Quad Edge データ構造と同様に空間双対なデータ構造である(図3).

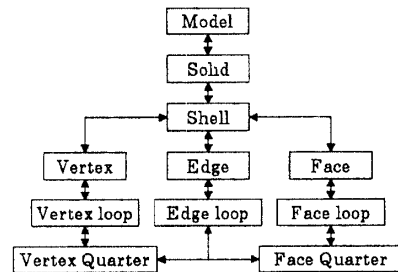


図3. Quarter Edge データ構造の接続関係

ここで互いに双対な Face と Vertex, Face loop と Vertex loop, Face Quarter と Vertex Quarter には DFaceVertex, DFaceVertexLoop, DQuarter ('D'は"Duality"の意)というそれぞれ共通の構造体(実際はオブジェクト指向によるクラス)を使用する. 例えば DQuarter は表1に示すような位相情報を保持する.

表1. DQuarter が保持する位相情報

*Prev	*Next	*Pair
*FaceVertexLoop	*EdgeLoop	
*Model()	*Solid()	*Shell()
*FaceVertex()	*Edge()	*Other()

表1で括弧がついているのはオブジェクト指向によってメソッドとして与える仮想的なポインタで, 例えば *FaceVertex()は FaceVertexLoop を介してポインタを

Modelling Solid Objects Based on the Principle of Duality

Hisashi TSUGANE, Norimasa YOSHIDA,
Jun DOI, Junhyuk KIM, Toshiya YAMAUCHI,
Fujio YAMAGUCHI

School of Science and Engineering, Waseda University
3-4-1, Ohkubo, Shinjuku-ku, Tokyo 169 Japan

得る。

実際のプログラムではこれらの位相情報以外にメソッドとしてオイラーオペレータなどの位相操作を与え、これらを継承した上位のクラスで頂点の座標値や面分の平面式係数といった幾何情報を保持・操作する。そうすることで双対な位相情報の保持・操作を完全に 1 つのプログラムで実行できる。

3. 双対なオペレータ

3.1. 双対原理を利用したオイラーオペレータ群

オイラーオペレータは様々な位相操作をおこなう関数群である。そのなかで頂点に関する操作と面分に関する操作は双対であり、Quad Edge データ構造のように頂点と面分の概念を抽象化して操作することにより、双対な操作は 1 つのプログラムで実行できる。表 2 にこれまでのオイラーオペレータ群と、双対原理を利用した新たなオペレータ群を示す。双対なオペレータは 1 つにまとめられ、16 個あったオペレータは 10 個に減少した。

表 2. 新旧オイラーオペレータ群

旧オペレータ群	双対オペレータ群
mef, mev (kef, kevv)	mefv (kefv)
mekif, mekliv (kemif, kemlv)	mekifv (kemifv)
mfkifh, mvklvh (kfmifh, kvmlvh)	mfvklfvh (kfvmlfvh)
mvfss (kvfss)	mvfss (kvfss)
mvfs (kvfs)	mvfs (kvfs)

また、Edge loop の導入により、新たなオペレータ $mehkl_e$ と $kehml_e$ [6] を追加したが、双対原理の利用による比較という意味でここでは省略した。

3.2. 抽象化されたオペレータの操作

双対な操作は抽象化され 1 つのプログラムで実行できる。ここではその例として、mef と mev を実現する共通のオペレータ mefv の操作を示す。図 4 はその抽象化された位相操作である。

ここでは稜線と点線で表わされたループが加えられ、それに応じて接続が更新されている。実線のループと点線のループをそれぞれ、頂点・面分として解釈するか(図 5 左の mef)、面分・頂点として解釈するか(図 5 右の mev)によって、このオペレータは双対な 2 つの操作を実現できる。実際に関数は、引数として FaceQuarter が渡されれば Face に関する処理 (mef) を、VertexQuarter が渡されれば Vertex に関する処理 (mev) をおこなう。関数内では "FaceQuarter" や "Vertex" というように、それが面分に関するものなのか、頂点に関するものなのかを明示して記述することは

ない。それらはすべて、"Quarter" や "FaceVertex" というように抽象化された形で記述される。

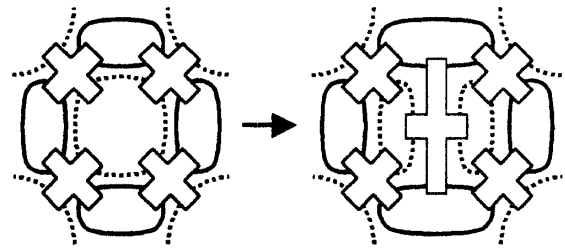


図 4. 抽象化された位相操作(mefv)

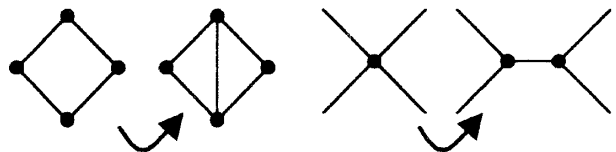


図 5. 双対な 2 つの操作(mef と mev)

このように頂点と面分を抽象化したアルゴリズムを記述することによって、双対な 2 つのオペレータを 1 つのプログラムで実行できるようになった。

また、これらのクラスを継承してブラックボックス化し、視覚的にわかりやすい操作ができるライブラリを制作した。これにより、表面上は複雑な Quarter Edge や双対原理を感じずに、様々な操作をおこなうことができる。

4. まとめ

ソリッドモデラに双対原理を利用し、データ構造とオペレータにおいて双対な処理を 1 つのプログラムで実現した。これにより、①プログラムサイズが小さくなる、②アルゴリズムが簡潔になる、③信頼性が容易に得られる、といった利点を得た。双対原理は概念はやや複雑であるが、問題が単純化されるということが確認された。

参考文献

[1] 弥永昌吉, 平野鉄太郎: 数学全書射影幾何学, 朝倉書店, (1959).
 [2] 山口富士夫: 4 次元理論による図形・形状処理工学, 日刊工業新聞社, (1996).
 [3] B. Chazelle, L. J. Guibas and D. T. Lee: The Power of Geometric Duality, *BIT*, Vol.25, pp.76-90(1985).
 [4] L. Guibas and J. Stolfi: Primitives for the Manipulation of General Subdivision and the Computation of Voronoi Diagrams, *ACM Transactions on Graphics*, Vol.4, No.2, pp.74-123(1985).
 [5] 新関雅俊, 小西史和, 吉田誠, 山口富士夫: データ構造およびオイラーオペレータの双対性, *精密工学会誌*, Vol.59, No.8, pp.1233-1238(1993).
 [6] 津金, 吉田, 井口, 北原, 土井, 山口: クォーターエッジデータ構造への稜線ループの導入, 1997 年度精密工学会春季大会学術講演会講演論文集, pp.9-10(1997).