平成 22 年度
修士論文

iPad によるグラフ理論研究支援ツール “gm standard for iPad” の研究・開発

横浜国立大学　環境情報学府
情報メディア環境学専攻 環境数理解析学コース

学籍番号　09hc002
秋末敦史

指導教官　根上生也
提出日　平成 23 年 1 月 25 日

Contents

第 1 章　はじめに

◇序文、研究の背景

21 世紀のはじめの年、2000 年にアメリカのクレイ数学研究所によって 7 つの数学上の未解決問題、いわゆるミレニアム懸賞問題（millennium prize problems）が発表された。問題の解決者には 100 万ドルの懸賞金が提示されるほどの大難問であるが、そのうちのひとつである『ポアンカレ予想』がロシアの数学者、グリゴリー・ペレルマンによって解決されたことは記憶に新しい。

それはいったいどういう問題だったかというと、「単連結な 3 次元閉多様体は 3 次元球面 S^3 に同相である」というものであった。これは、『フェルマーの最終定理』のような、日本の義務教育を受けたものなら誰でも意味がわかるような問題ではない。そのためにニュース等で広く報道はされても大衆の関心を惹くことはなかった。しかし、この問題の解決がマスメディアにのって広く知れ渡ることで、この難問の解決に有効であろうと思われた“トポロジー”という数学の分野に注目が集まったのも事実である。

そうしてこの問題に関する数多の解説本が書店の数学コーナーを埋め尽くし、トポロジーという数学分野が広く認知されるようになったことに一役買った。そのため、位相幾何学（トポロジー）的グラフ理論を専門とし、且つ私の指導教官である根上生也教授も「トポロジカル宇宙」（日本評論社、1993 年）を問題解決後の 2007 年に[完全版]として増補、再版するとともに、精力的に講演会をこなしている。

一方、根上教授が主催する TGT（“T”opological “G”raph “T”heory）という学会も今年も例年通り開催され、盛況のうちに終わった。

以上のように、世間的な関心もあり、研究者たちの意欲、熱意もますます昂ぶっている今、トポロジー、グラフ理論、そしてそれらを融合した位相幾何学的グラフ理論のより一層の進展のために、研究者の意欲をさらに亢進させるような、思索のための補助ツールがあると非常に便利ではないか、それを作りたい、というのが私の研究の主たるモチベーションである。

グラフ理論とは、“点と線からなる素朴な図形を対象に、組み合わせ理論もしくは離散数学の一分野として 20 世紀後半に急速に成長を遂げた数学”*である。

このグラフ理論を研究するにあたって、思考のツールとして紙と鉛筆があれば大げさな研究施設がなくてもどこでも研究が可能である。しかし、研究者の思考のスピードと量の前には、紙と鉛筆ではあまりに非効率、非経済的であろう。なぜなら、試行錯誤の段階では、一度書いてしまったものを消すのは手間であるし、消している合間に思考の流れを断

*根上生也：位相幾何学的グラフ理論入門、横浜図書（2001）　「はじめに」より抜粋

ち切ってしまう可能性もありうるからである。

さらにいえば、位相幾何学的グラフ、つまり閉曲面―たとえば浮き輪の表面―のグラフ、はたまた三次元空間には収まりきらない曲面上のグラフを考える場合には、その閉じた曲面を切り開いて、展開図という形で二次元平面に落し込み、その上でグラフを描くことで問題を考えていく。切り開いた辺は同一視して考えていかなければならず、その点を考慮しつつグラフを描く必要がある。考えなければいけない要素が増えることで、思考の流れは澱みやすくなる。紙と鉛筆だけで思考を重ねていては、思い違いや思わぬ見落としを誘発しやすくなってしまう。

こうした問題を解決するひとつの方法として、コンピュータを使ったグラフ描画アプリケーションは大変有用であると考えられる。コンピュータ上でグラフを書いたり消したりすることは、ペイントソフトなどを見てもわかるように、紙と鉛筆よりも容易であるし、その結果大量のゴミがでることもない。また、色で塗り分けたり、注目点を設定したりと、グラフの操作をすることもはるかに簡単であろうことは誰もが納得できるところである。さらに言えば、プログラムを組んでコンピュータに必要な計算処理させることが思考の大きな手助けになり、同じ時間でより深い研究が行えるであろう。

そうした背景のもと、1998 年に根上教授が「gm standard」というグラフ描画ソフトを作成した。しかし、10 年以上の月日を経たせいもあってか、ユーザーインターフェースをはじめとして、あまり実用的でない部分がでてきていた。そこでこのアプリケーションの基本的な理念を継承しつつ、改良を加えて、位相幾何学的グラフ理論研究のための実用的な支援ソフト「AnchorRing」を私の卒業論文の成果として作成した。

そして今回は、これら既存のアプリケーションの問題点を検討し、さらに実用的な研究支援ソフトを作成することを目指した。そこで今回私が掲げたテーマは、「研究者間の議論を加速させる、コミュニケーションの媒体としてのアプリケーション」である。

既存のものはどちらも相手との対話におけるツールというよりは、自分一人で思索を巡らすときに効果を発揮するアプリケーションであった。しかしながら、研究は時に様々な人間が共同で行うこともあり、議論もまた必要なことである。そうした man to man のコミュニケートの場においても効力を発揮するアプリケーションというのは未だ開発されていない。そこで今回、最近世間で注目されている iPad に注目した。PC よりも小型で持ち歩きがたやすく、また指を直接触って操作するという tangibility が、コミュニケーションメディアとしてかなり有効であると考えたためである。

そこで本稿は、iPad 上で動くグラフ理論研究支援ツール「gm standard for iPad」の作成にあたって、既存のアプリケーションの問題点を改善、実装した機能の理論的、具体的方法について述べた論文である。

第 2 章　先行事例

本研究の内容に入る前に、既存のアプリケーションについて概論する。それを通してツールとしての利点と問題点を再度整理し、本研究の意義と目的を明確にしたいと思う。

◇2−1「gm standard」概要

図 1.に示すのが「gm standard」の外観である。サブウィンドウ上にマウスでグラフを描けるキャンバスがあり、メインウィンドウのメニューバーから、そのグラフに関して様々な処理（頂点、辺へのラベリング、着色、回転、移動、切り取り、コピー等）を施せるようなユーザーインターフェースになっている。

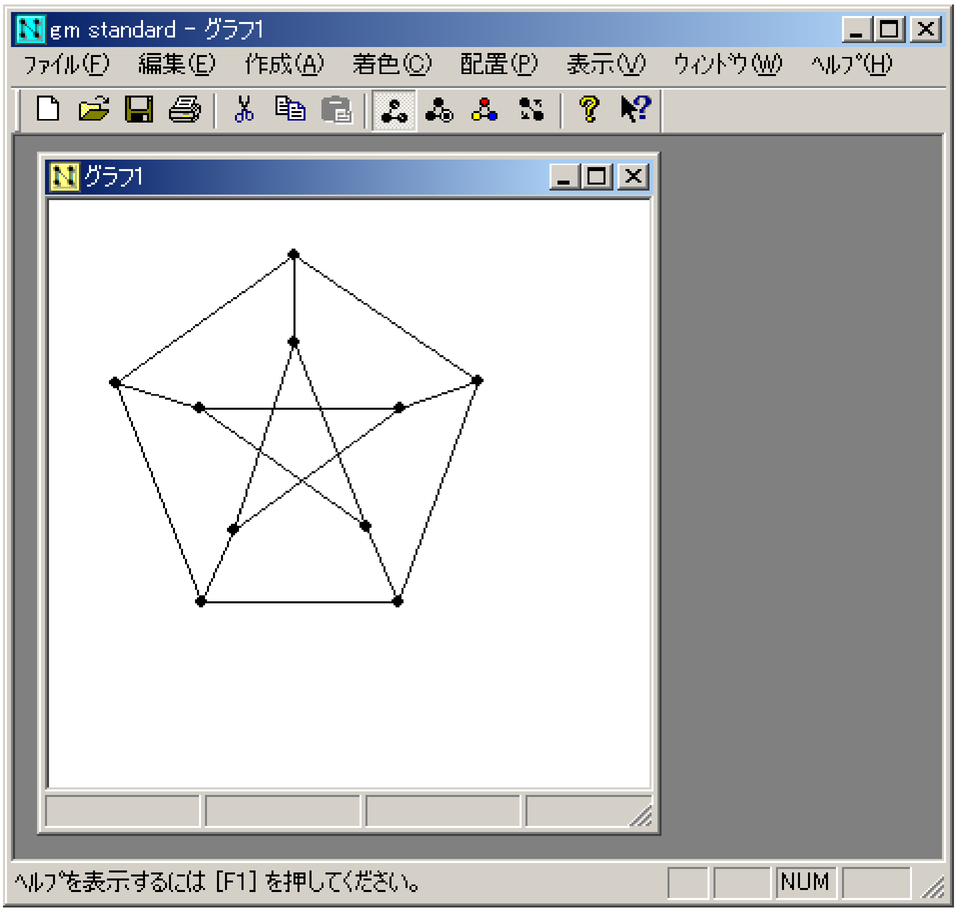

図 1.　「gm standard」外観

マウスを使ってグラフを描くことで、アナログな方法よりも効率的にグラフを描画することを可能にしている。着色や回転などのグラフの処理も同様に可能になっている。

◇2−2　「gm standard」の問題点

◆　2−2−1　ユーザビリティ

マウスが頂点に近づいても何の変化も見て取れないために、ドラッグをしてその頂点が動かせる状態にあるのかどうかという判断がつかない。また、任意の辺を消すためには辺をダブルクリックという設定になっているが、細い線上にカーソルを合わせるのはなかなか神経を使う。以上のように、ユーザビリティにかかわる問題が散見される。

また、グラフに線をなめらかに表現するための処理がされていないために、全体としてグラフが非常にシャギーになってしまっている（**図 2.**）。デザイン上の問題なので本質に関わらない瑣末なことに思えるが、ユーザーを意識したコンテンツを作る以上は無視できない問題である。

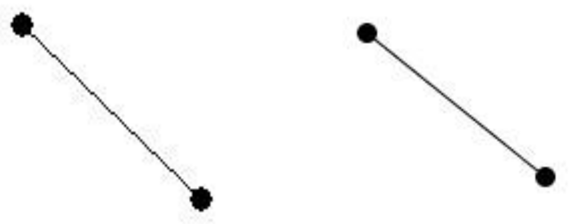

図 2. アンチエイリアス処理前（左）と処理後（右）

◆　2 −2−2　二次元閉曲面へのグラフ描画

「gm standard」がグラフを描画する舞台として提供しているのは「平面」のみである。それはつまり、紙の上に書かれたグラフをコンピュータ上で再現することには成功したということである。しかし、位相幾何学的グラフ理論という研究分野においてグラフを描く舞台は「平面」のみにとどまらない。「球」「トーラス」をはじめとして、「クラインの壺」、「射影平面」などに代表的される、二次元曲面上のグラフについての研究も存在する。

「gm standard」にも、「トーラス上の K７.gm」（**図 3.**）や、「射影平面上の K6.gm」（**図 4.**）といったグラフのデータがデフォルトで用意されているが、それとて曲面を展開して平面上に書き写しただけであり、あくまでも平面上に描かれたグラフとしてしか操作できない。

というのは、図 3.と図 4.で示したグラフの一番外側、四角形と六角形の描かれている部分がそれぞれの閉曲面を平面に展開したときの切れ目となっているため、その切れ目の上に乗っている頂点／辺は、同一視されなければならない別の頂点／辺が１つ以上描かれている。図 3.も図 4.を例にとると、同じ印で囲まれている点が同一視されるべき点である。

そうであるにも関らず、これらはコンピュータの内部では違う頂点として処理されているため、閉曲面の切れ目にある頂点を動かしても、同一視されるべきもう一方の頂点が動くことはない。したがって、閉曲面上にあるグラフを表現することはできても、このグラフの構造―頂点と辺の接続関係―を維持したまま操作することはできない。

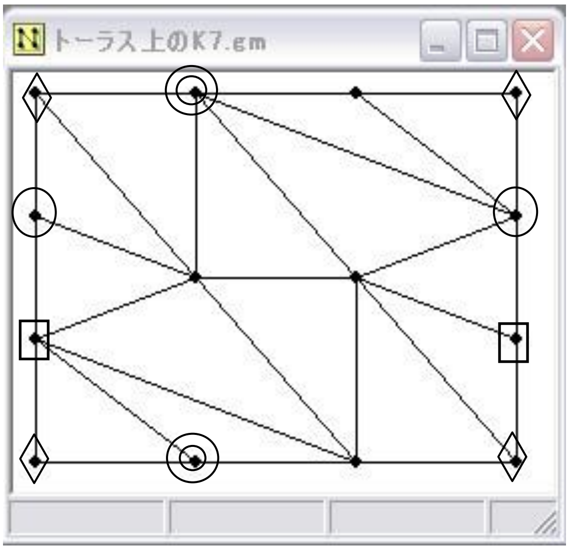

図 3. トーラス上の K７.gm

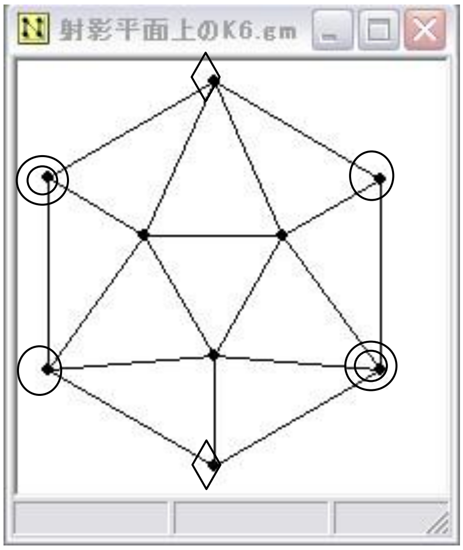

図 4. 射影平面上の K6.gm

◇2－3 「Anchor Ring」概要

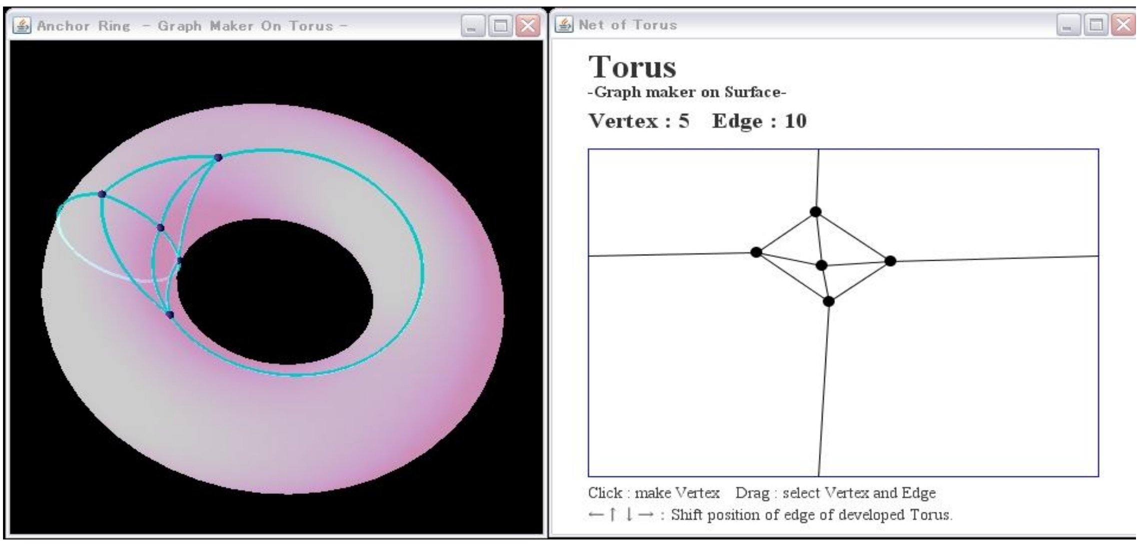

図 5. Anchor Ring 外観

図 5.に示すのが、「gm standard」の改良を目的として作成された研究支援ツール、「Anchor Ring」の外観である。見てわかるとおり、二つのウィンドウを用意し、片方（図 5.左）は３D 表現した閉曲面、片方（図 5.右）は展開図である。二つのウィンドウはリアルタイムに同期していて、展開図に描かれたグラフはすぐさまもう片方のウィンドウに反映される。それは辺の描画中においても同じであるので、辺が実際の閉曲面上ではどの部分を通って描かれるのかも一目瞭然である。

図 5. 右の展開図の境界線に接するように描かれている辺は、上下、左右で同一の辺である。この境界線が閉曲面の切れ目であり、頂点／辺はこの境界線を越えて繋がることができる。また、境界線上に頂点や辺が乗っかった場合は、対応するもう一方の境界線上に自動的に頂点／辺が描かれる。しかしながら、データ構造的には同一の頂点／辺として管理されているので、グラフが動かされた場合にも、グラフの構造が変わらずに表示される。

◇2−4　「Anchor Ring」問題点

「Anchor Ring」が標準装備している閉曲面は、トーラスのみである。これ以外にも閉曲面は無数に存在し、位相幾何学的グラフ理論がそのグラフを描く舞台とする閉曲面はトーラスだけには限らない。トーラスのみに対応できても、実用性は低い。

◇2−5　本研究における改善目標

以上のことがらを踏まえ、本研究で作成するソフトウェアは

- ユーザビリティに関する問題点の修正
- あらゆる閉曲面の展開図上にグラフを描くための理論の考案と実装

この 2 つを主な目的とする。特に 2 つめ、あらゆる閉曲面へのグラフを描画するというアプリケーションは今のところ存在しない。「Anchor ring」のように、3D 描画を伴わせることは、閉曲面が 3 次元空間上に表現できない場合や、表現できたとしても展開図と空間の 1 対 1 対応がうまくできる保証がないため、今回は見送る。しかしながら、専門家は専ら展開図を描いてそれら閉曲面上のグラフを研究しているので、上記の目標を実現することで、より広くグラフ理論研究者のサポートができると考え、本研究ではこれを狙う。

第 3 章　位相幾何学的グラフ理論

位相幾何学的グラフ理論とは、グラフを曲面や空間に埋め込むことで起こる様々な現象と、それに付随するグラフの構造を研究する分野である。ここで、本稿を読む上で前提となる、それに関する知識を簡潔にまとめておく。

◇3－1　グラフとは

本研究の要であるグラフを明確に定義する。グラフ(G)とは、「点」を「線」で結ぶことによって作られる極めて単純な図形のことである。この「点」を「頂点」と呼び、「線」を「辺」と呼ぶ。G におけるそれぞれの集合は$V(G)$、$E(G)$とあらわす。

辺を描く場合に、曲線で描いたり、同じ場所になんども辺を描いたり（多重辺）、同じ頂点に帰ってくるように描いたり（ループ）という書き方も可能であるが、今後グラフという言葉を使う際は、辺は直線、しかも多重辺とループを許さないグラフ（単純グラフ）のことをさすことにする。

◇3－2　種数・向き付け可能／不可能

グラフを描く舞台は、必ずしも平面だけとは限らない。なぜなら、「閉曲面」という舞台を設定することで、グラフは新しい構造や性質を表すことがあり、深く探求していくだけの魅力を備えているからである。「閉曲面」の定義とは「境界のないコンパクトで連結な 2 次元多様体」であるが、とりあえずここでは「球面」や「トーラス」を思い起こしてほしい。「トーラス」とは浮き輪のような形の表面のことである。その他代表的な閉曲面として、「クラインの壷」や「射影平面」などが挙げられる。**図.6、図.7** に挙げるのは、「トーラス」と「射影平面」の二つの射影平面である。

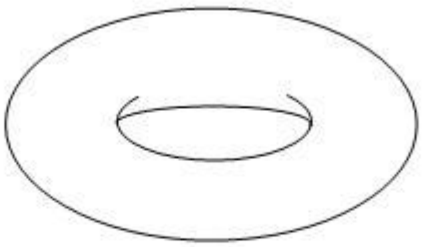

図 6. トーラス(T^2)

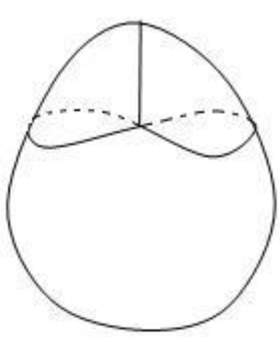

図 7. 射影平面(P^2)

トーラス(T^2)は見たとおりの形をしているが、射影平面(P^2)は、三次元空間にはその真の姿を投影することができないため、あくまで擬似的な形である。実際には、円盤の直径の両端を同一視して作られる閉曲面である。

位相幾何学において、全ての閉曲面はこの閉曲面を連結したものと位相同型、つまり同じ閉曲面であるととらえる。つまり、これらがいくつ連なって構成された閉曲面であるかということから、閉曲面を分類することが可能である。

閉曲面の連結は一般的に($F_1^2 \# F_2^2 \# \ldots \# F_n^2$)という形で表すこととして($F^2$は一般的な閉曲面)、($T_1^2 \# T_2^2 \# \ldots T_g^2$)という形の閉曲面を、「種数 g の向き付け可能(orientable)な閉曲面」と呼び、S_gで表す。同様に($P_1^2 \# P_2^2 \# \ldots P_k^2$)という形の閉曲面を、「種数 k の向き付け不可能(non-orientable)な閉曲面」とよび、N_kで表す。

ここで“向き付け”とは、曲面全体で時計回りの方向を一致させることができるか、ということである。図 8.に示すとおり、メビウスの輪のように、表裏の区別がつけられないように短冊を捻って両端をくっつけた閉曲面の上では、一周回ってきたところで時計回りの方向が逆になってしまうという現象が起こる。そのため、このような閉曲面は向き付け不可能な閉曲面と呼ばれる。

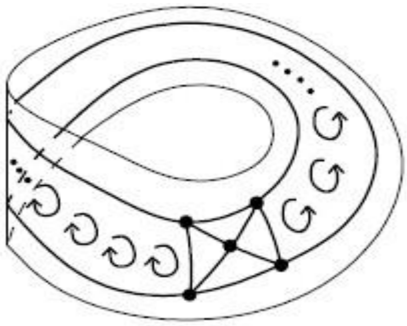

図 8. 向き付け不可能な閉曲面（メビウスの輪）

位相幾何学において全ての閉曲面は、この「向き付け可能／不可能」と「種数」によって分類することとしている。

◇3−3　閉曲面の展開多角形

閉曲面上のグラフを考える際には、紙面上で考えられるようにその閉曲面に切り込みをいれて展開し、平面上で考えられるようにすることがほとんどの場合である。その際、周囲を矢印の方向に従って同一視する。

図 9.を例にとった場合、トーラスは上辺と下辺、右辺と左辺を同一の境界とみなす。また、射影平面の場合は、円盤の直径の両端を同一視すればよいので、かならずしも図 9.のように六角形で描く必要はないが、慣例的にこの形で描かれることが多い。

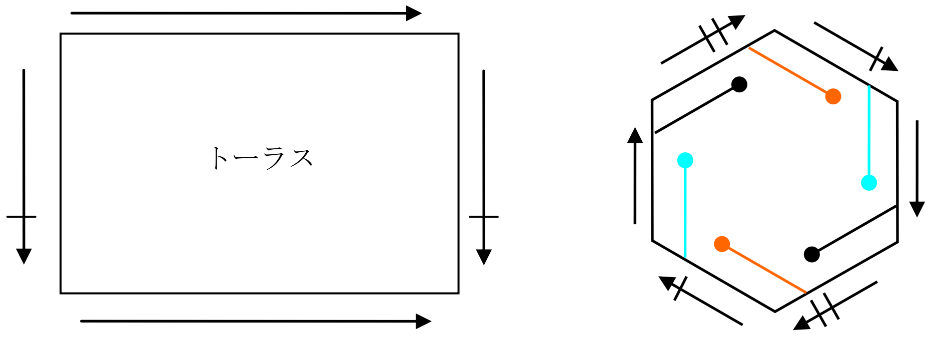

図 9. トーラス($S_1 = T^2$)、射影平面($N_1 = P^2$)の展開図

また、展開図の切り開き方には一定の法則があり、種数 g の向き付け可能な展開図の場合は、4g 角形の辺を、

$$a_1 b_1 a_1^{-1} b_1^{-1} \dots a_g b_g a_g^{-1} b_g^{-1}$$

という形で切り開くことができる。–1 という指数は、矢印の方向が逆になることを示す。図 9.のトーラスを例に考えていただければわかるであろう。

一方、種数 k の向き付け不可能な閉曲面の場合は、2k 角形の辺を

$$a_1 a_1 \dots a_k a_k$$

という形で切り開くことができる。射影平面の場合は k が 1 なので、多角形を構成することはできない。そこで先程も述べた通り慣例的に六角形で表すことになるが、その際は辺の対応関係がすぐ次の辺ではなく、向かい側の辺と対応していることに注意する必要がある。

第 4 章　研究計画

前章までの話をふまえて、実際にどのようなコンテンツを、どうやって開発していくかをここでまとめる。

◇4−1　任意の閉曲面の展開図へのグラフ描画

「gm standard」は、平面上のグラフしか扱うことができない。というのは、展開図上でのグラフの構造—展開図の境界線上の頂点／辺は、対応するもう一方の頂点／辺と同一視する等—が、閉曲面上のものではなく、あくまで平面に描かれたグラフとしてのデータ構造になってしまっているのである。

また、「Anchor Ring」では、データ構造的に閉曲面上のグラフとして扱うことができるエディタとなったが、いかんせん閉曲面の展開図がトーラスしか装備されておらず、未だ研究支援ツールとしての汎用性が非常に低い。より有用なエディタであるためには、あらゆる閉曲面に対応している必要がある。

そこで今回は、展開図の切り開き方にある一定の法則があることを足がかりに、種数と向き付け可能／不可能というパラメータから任意の閉曲面の展開図を描き、同一視するべき境界線が繋がっているようにグラフを表示／編集させることができるシステムを考える。

◇4−2　デバイスをよりハンドヘルドに　-iPad で動作するアプリケーション-

「gm standard」も「AnchorRing」も、動作するハードウェアは PC であった。そのためアプリケーションの使用を想定している人数は 1 人であり、1 人で思索を深めるための有効な補助ツールであったといえる。もしこれを使って人と議論しようとすると、操作するためのマウスを他人と取り合いながら議論するという形になり、あまりスマートな方法であるとは言えない。また、PC のモニターを眺めながらのコミュニケーションというのも、顔と顔、体と体が向きあってない状態で、対話しにくいと考えられる。人間同士が話合うときというのは､体が向き合っている方が自然である｡体が同じ方向を向いていても、同じモニターを眺めているのであればさして苦痛を感じないかもしれない。しかし、先述したとおり議論の対象となっているグラフを操作するためのデバイスである、マウスなりキーボードなりは、PC という性格上、だいたいにおいてひとり分しか用意されていないのが普通である。これを取り合って議論をするのはなかなかストレスフルであると考えられる。

そこで、今回は研究者間の議論をよりスムーズにするためのツールを開発することとしたい。これまでも、学会や研究集会の場で、研究者同士が大きなホワイトボードを前に、

図をいくつも書いたり消したりしながら議論をしている場面を幾度となく見たことがある。その際のホワイトボード的な役割を、iPad というデバイスが担うことができると思われる。そして iPad でアプリケーションを作ることのメリットが他にもいくつかあると思われる。その利点は、下記の 3 つにおおよそまとめられるであろう。

・ハンドヘルド PC であることのユビキタス性
・タッチパネルによるフィジカルな操作性
・研究者間のコミュニケーション促進

iPad は持ち運びが普通のノート PC などよりも格段にたやすい。どこにでも持っていけるし、いつでも好きなときに動かすことが出来るという意味においてユビキタス性が高いと言える。

また、ご存知の通り iPad はタッチパネルによる操作であるので、マウスで操作する PC 上でのアプリケーションよりも更にフィジカルにグラフを操作することが可能となる。これはつまり、自分の体を動かすという「行動」とその「結果」がよりダイレクトに結びつくために、操作方法が体に馴染みやすくなり、とっつきやすいアプリケーションとなるであろう。

そして、研究者が体と体を向け合った状態で、その中央に、iPad を配し、双方からグラフを編集しあいながら、思索を深めていくということが可能になる。このことが、頭の中で考えていることがより早く形として現れるために、より短い時間で多くのことを話しあうことができる。さらにいえば、研究者間のコミュニケーションの量を増加させるための起爆剤となることが期待できるかもしれない。

◇4−3　開発言語・開発環境・動作環境

現在、iPad ならびに iPhone でのアプリケーションの開発には、Objective-C という言語を使うしかない。この言語は、既存の C 言語に現在のプログラム言語の主流な構造である「オブジェクト指向」を取り入れて作られた言語である。この言語を使用した、一連の API 群(フレームワーク)である Cocoa と、iPhone/iPad 用の SDK を導入することで今回のアプリケーションを開発した。

開発環境には、Mac に標準で装備されている Xcode という統合開発環境を使用する。また、動作環境は言うまでもなく iPad であるが、Xcode には iPad のシミュレータが搭載されているので、Mac があれば動作状況を確認することができる。また、スタンドアロンなアプリケーションであるので、インターネットへの接続環境は必要ない。

第5章　位相幾何学的グラフ理論研究支援ソフト「gm standard for iPad」の概要

今回作成した「gm standard for iPad」の外観を図 10.に示す。4 章で述べたとおり、iPad のデバイスにて動作する。開発途中のもののため、インターフェースに若干の不備が見受けられるが、主要なシステム自体は完成しているので、展開多角形の設定とグラフの表示／編集のシステムについてここでは説明する。

図 10.に示している 4 つの展開図は、左から順番に、「種数 1 の向き付け可能な閉曲面」、「種数 1 の向き付け不可能な閉曲面」、「種数 2 の向き付け可能な閉曲面」、「種数 4 の向き付け不可能な閉曲面」である。これらの多角形は「種数と向き付け可能／不可能」というパラメータのみから自動生成されている。

また、展開図の周りの矢印が境界線の対応関係を表している。ところどころに展開図の頂点から直線が伸びている部分があるが、直線と直線の間の展開図の辺が種数 1 つ分を表していることを示すためのものである。図 10.の右側 2 つは、境界線の対応関係を示すために展開図の周囲の矢印に「●◯」のマークが付随しているが、同じマークであっても直線で区切られた間以外の部分とは何ら対応関係がないことを展開図から伸びている直線が明示している。

グラフの頂点はダブルタップで生成／削除できる。当然ではあるが、展開図の中でのみ生成され、展開図の辺に近づいていくと、その辺上に載せることができ、対応するもう一方の展開図の辺上に自動的に同一視すべきグラフの頂点を描く。展開図の頂点にグラフの頂点が乗った場合は、全ての展開図の頂点に同一視するグラフの頂点が描かれる。（図.10 の一番左を参照）同一の展開図の辺上にあるグラフの頂点間のグラフの辺の場合も同様である。境界線上にある頂点の削除は、どちらかをダブルタップしても、両方を一度ずつタップすることでも削除することが出来る。

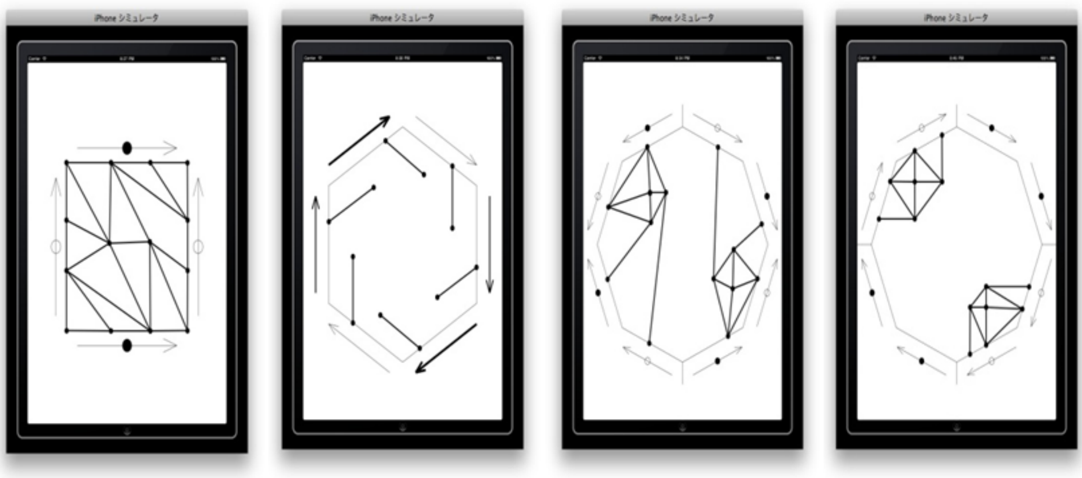

図 10. gm standard for iPad 外観

第 3 章で、グラフは多重辺、ループを許さないこととする、と述べたが、グラフの頂点が展開図の辺上にあるために 2 箇所以上同一視するグラフの頂点が描画されており、そのそれぞれへ向けた辺であればシステムとしては実現可能である(**図.11**)。ただし、7 章で詳述するが、グラフの辺は展開図の辺を通過しないシステムになっているので、多重辺、ループした辺をもつ展開図の辺上のグラフの頂点が展開図内に動かされた場合には、多重辺の場合はひとつの辺として展開図内に描画され、内部データとしても辺はひとつとする仕様になっている。ループした辺の場合も描画ができないのでデータ的にも消滅する。

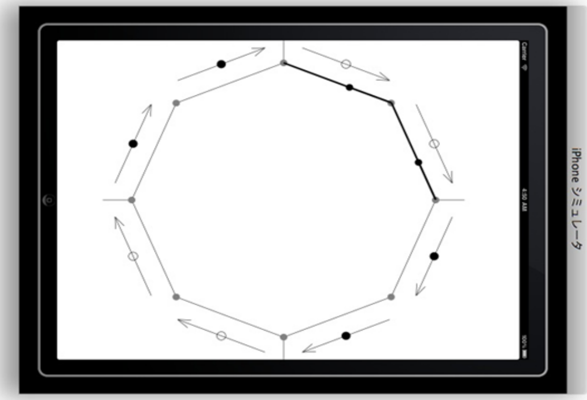

図 11. 展開図の頂点におかれたグラフの頂点と、境界線上の頂点の間の辺
(灰色で描かれた頂点は全て同一頂点)

何も描かれていないところを、辺を通過するようにタッチパネルをドラッグすることで**図 12.**のように辺の選択が可能である。選択された辺は赤色で表現される。選択された辺は、もういちど同じように選択することで削除される。逆に辺を作成したい場合は、結び合う頂点同士をタップするか、どこかの頂点を一度タップした状態で、頂点が存在しないところをダブルタップすることで頂点の作成と同時に辺も作成される。

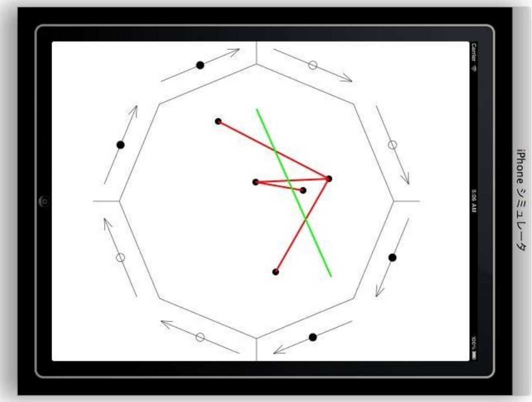

図 12. ドラッグによる辺の選択(緑色のラインが自分でドラッグした通り道)

◇5－1　システム全体の概略

システム全体の構造は**図 13.**のとおりのような構成になっている。全体を統括する reportView.m というクラスが主にグラフの編集を行う関数などを保持しているクラスであり、必要なデータは、種類ごとにパーツとしてまとまっており、それぞれを呼び出し／編集／参照することによってシステムが成り立っている。

データは 4 つの種類に別れている。ひとつは展開図を表現するために必要なデータが格納された Field.m、そして、今の状況がどういう状態であるのか—頂点が選択された状態か、辺が選択された状態かなど—を管理する Flag.m、最後に、実際にグラフのデータを保持している Gpoint.m（頂点）と Edge.m（辺）である。

Gpoint.m と Edge.m のデータは、それぞれひとつの頂点、辺を管理しているので、グラフの編集による頂点数、辺数の変化に応じてデータの数は可変する。

また、Gpoint.m は、頂点の座標情報を保持するとともに、展開図の辺上に乗った場合に、同一視すべき頂点として描かれるもう一方の頂点の座標を、Imaginary vertex として可変データ配列で保存している。可変である理由は、頂点が展開多角形の頂点に乗った場合にも対応するためである。

Edge.m クラスは、どの頂点とどの頂点が結びついているのか、そして結びついているのはその頂点の original であるか imaginary であるか、imaginary であるならば何番目の imaginary Vertex と結びついているのかのみを把握している。具体的な座標情報は、Edge.m が持っている情報から Gpoint.m の情報を取得するという方法をとっている。

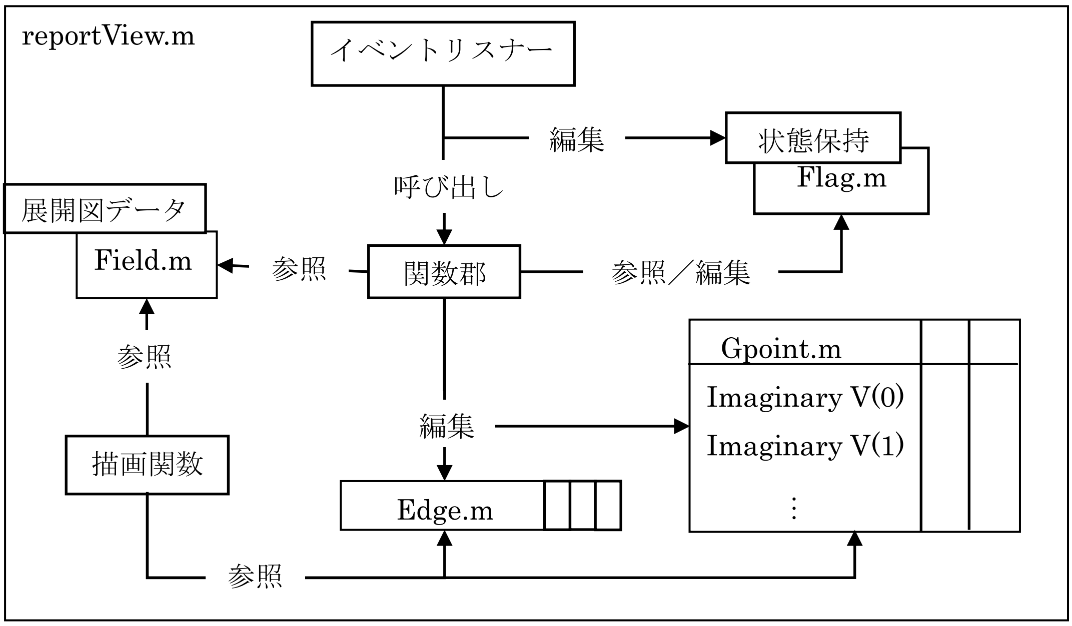

図 13. システムの全体図

◇5－2　閉曲面の展開図の境界線判定

頂点を移動する際、当然ながら頂点は展開図の範囲を越えて描かれてはならない。ところが、展開図は種数と向き付け可能／不可能のパラメータから生成される任意の閉曲面であるので、全ての展開図において通用する展開図内外の判定方法でなければならない。それがどのような理論の基に実現可能にしているかを、ここで詳述する。

◆5－2－1　閉曲面の展開図の描画方法

第 3 章でみた通り、閉曲面は 4g 角形もしくは 2k 角形として切り開くことができる。そのため、全ての展開図はかならず正 2n 角形として描くことができる。このことを利用して、展開図の各頂点の座標を定めていく。始点となる座標は、$(x, y) = (0, field_radius)$とする。field_radius はその展開図と外接する円の半径である。次に、その外接する円の展開図の頂点間の円弧をラジアンで示すと、π/nである。よって、後は始点の座標をπ/nずつ$2n-1$回だけ回転させれば、閉曲面の展開図の各頂点の座標を求めることができ、これらの座標間を繋げれば、展開図を描くことが出来る。(図.14)

二次元座標の原点を中心にした座標の回転は、下記の行列にて求められる。なおプログラミング上はθをラジアンの値で与えることで求まる。

$$\mathrm{Rotate}(x, y) = \begin{pmatrix} s \\ t \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

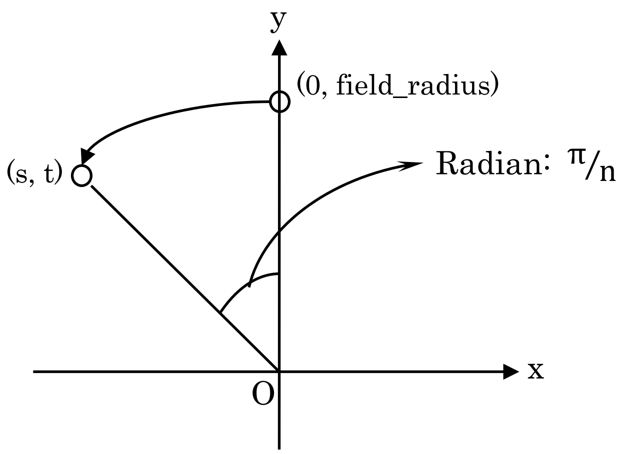

図 14. 展開図の描画方法

◆5－2－2　判定方法

こうして描かれた展開図は、偶数角形であるので、必ず y 軸を対称軸として線対称となることがわかる。この事から、展開図の内部かどうかを判定する座標点の y 座標がとる値に応じて、展開図のどの辺と比較すればよいのかがわかる。(図 15.)

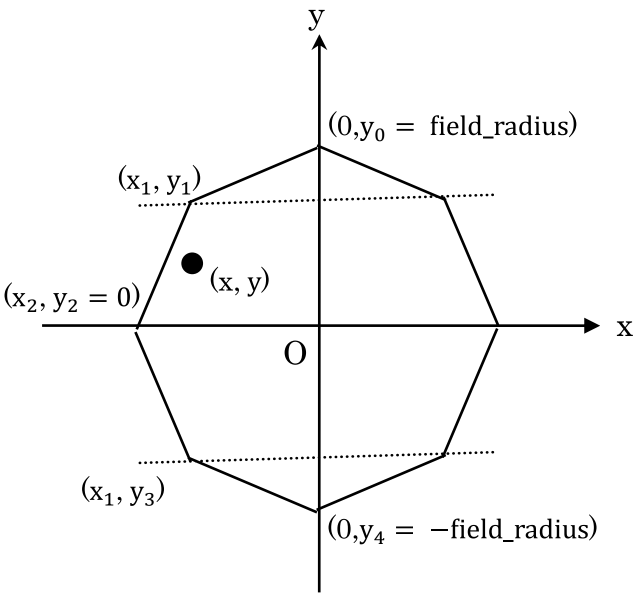

図.15 比較する辺の判定(2g, 4k の閉曲面の展開多角形の場合)

図.15 を例にとると、この場合は、$(y_1 > y > y_2) \cap (x < 0)$であるので、$(x_1, y_1)$ と$(x_2, y_2 = 0)$を通る一次関数 $f(x) = ax + b$(a, b は定数)よりも大きい値をとる、つまり$f(x) < x$ならば、展開図の内部であると判定することが出来る。

Field.m には、(0, field_radius)を始点として、反時計回りの順番に、展開図の頂点情報が格納されている。同時に、各辺の一次関数の情報（傾きと切片）も(0, field_radius)を始点として反時計回りの順番に Field.m は保持しているので、その値を使うことで判定することが出来る。

具体的なアルゴリズムとしては次に述べる通りである。2n 角形の始点の y 座標をy_0として、反時計回りに各頂点の y 座標y_1、y_2 …としていき、y_nの頂点まで、(y_0, y_1)、(y_1, y_2) …(y_{n-1}, y_n)、と 2 つずつピックアップして、その間に判定する y 座標の値が含まれるかを判定する。判定した回数(i)と、判定する点の x 座標が正の値か負の値かで、何番目に格納された辺の一次関数情報をピックアップすればよいかが決まる。展開図の各辺を E として、データが格納されている順番にE_0、E_1 … E_nとするとき、もし x が負の値ならばE_i、もし正の値ならばE_{2n-1-i}となる。(2n は展開図の頂点数)

判定する点の x 座標が正の時、比較する E の一次関数 f(x)との関係が f(x)>x ならば展開

図内、逆に負の時、f(x)との関係が f(x)<x ならば展開図の範囲内と判定することができる。

ちなみに、種数 1 の向き付け可能な閉曲面、つまりトーラスだけは 5-2-1 で説明したような展開図の描き方では、図形が菱形になってしまう。無論そのままでも展開図の構造としてはまったく問題ないのだが、慣例的にトーラスは正方形の形で描かれることが多い。そのために、始点を-45° だけずらした状態から書き始めることで、正方形として描くことにしている。(図.17)。上述した展開図の内外の判定は、この図形に対しても有効に機能することがわかるであろう。上辺と下辺は y 座標が同じであるので、判定に関与する辺ではないが、右辺と左辺の y 座標が異なるので、この両辺から内外判定が可能である。

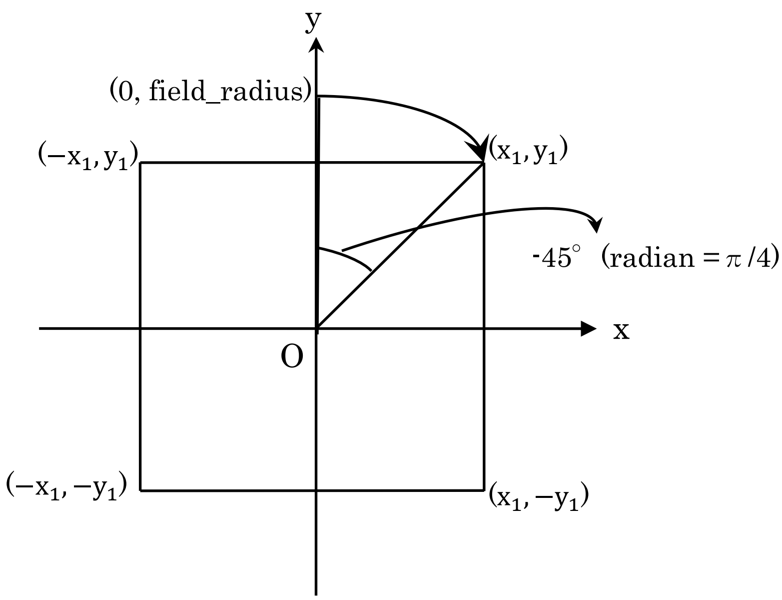

図.17 トーラスの場合の展開図の描き方

◇5－3　境界線上の頂点／辺と同一視すべき対応する境界線上の頂点／辺の決定方法

展開図の辺上にグラフの頂点／辺が乗っかる場合には、自動的に対応するもう一方の展開図の辺上にグラフの頂点／辺が描かれるシステムは、このアプリケーションの最も有益と考えられる機能である。ここで、どのような閉曲面であってもこの機能を実現するための理論と、その実装方法をここで説明する。

◆ 5－3－1　理論的解説

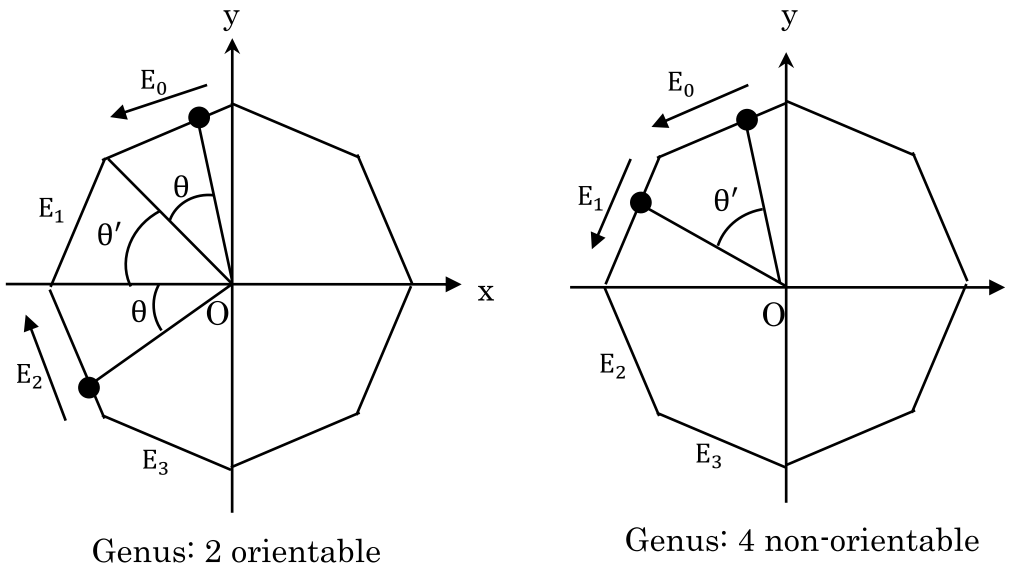

図.18 対応する頂点どうしの展開図の中心からの角度

図 18.に示した通り、向き付け可能な閉曲面の場合、「展開図の頂点から頂点の間までの原点を中心とした角度」(θ')と、「展開図の辺上のグラフの頂点と、展開図の辺の矢印の先側の頂点との間の、原点を中心とした角度×2」(2θ)を足した分だけ、原点を中心に回転移動させればよい。これは、展開図の辺上のグラフの頂点と、その展開図の辺の矢印の先側の展開図の頂点と原点が作り出す三角形が、合同であることから導き出せる。ただし、回転角度はグラフの頂点が乗っている展開図の辺の矢印の向きによって±が変わる。

向き付け不可能な閉曲面の場合には､同一視する展開図の辺が常に隣どうしであるのと、その境界線の矢印が同じ方向を向いているために､「展開図の頂点から頂点の間までの原点を中心とした角度」(θ')だけ、グラフの頂点を原点中心に回転させればよい。

回転角度の±の決定は、向き付け可能の場合、グラフの頂点が乗っている展開図の辺の矢印方向が反時計周りの場合と、時計回りの場合で分ければ問題ない。また、向き付け不可能の場合には、辺の番号(図.18 参照)が偶数なら+、奇数なら－となる。

◆ 5−3−2　実装方法

「展開図の頂点から頂点の間までの原点を中心とした角度」(θ')のラジアンの値は$\frac{\pi}{n}$としてすでに求まっているので、向き付け不可能な閉曲面の場合は問題ない。あとは向き付け可能な場合の、「展開図の辺上の頂点と、その辺の矢印の先側の展開図の頂点との間の原点を中心とした角度」(θ)を求めることができればよい。

Objective-C、ひいては C 言語には、逆余弦、逆正弦、逆正接を求める関数、acos、asin、atan という関数が標準ライブラリとして用意されているので、これら、特に逆余弦、acos を用いて角度を求めていくことになる。この関数は、余弦の具体的な数値から、対応する角度のラジアンを算出してくれる関数である。頂点座標を(x, y)とするとき、余弦は$\left(\frac{x}{\sqrt{x^2+y^2}}\right)$となるので、この値を acos という関数に渡せばそのラジアンの値を得ることができる。

しかしながら、**図.19** の示すとおり、その値は x 座標軸との間の角度なので、θを求めるためには、$(|\theta_1 - \theta_2|)$という処理が必要になる。絶対値を取るのは、頂点が乗っている展開図の辺の矢印の先側の頂点が、グラフの頂点よりも反時計回りを正の方向としたときに負の方向にあるために、値が負の値になる場合があるからである。

この「展開図の辺の矢印の先側の頂点」自体はどうやって求まるかというと、グラフの頂点の情報を保持する Gpoint.m 内に保持されている辺の番号を参照している。その番号は、xy 座標(0, field_radius)を始点として反時計回りに 0 から順番にナンバリングしたものがデータとして保持されている。

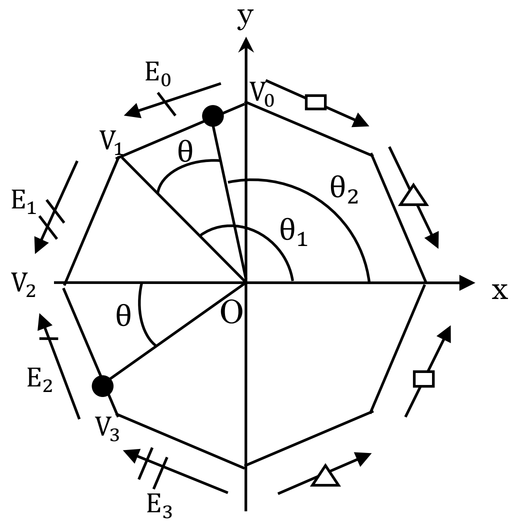

図.19 実際に移動する角度を求める(Genus：2 , orientable)

（承前）そして、グラフの頂点が乗っている展開図の辺上の番号を$\mathrm{E_i}$、展開図の頂点の番号を(0, field_radius)を$\mathrm{V_0}$として反時計回りに$\mathrm{V_1}$、$\mathrm{V_2}$としていくとき、向き付け可能な閉曲面の場合は、$\mathrm{E_i}$を 4 で割ったときの剰余が 0, 1 の場合には「展開図の辺の矢印の先側の頂点」は、$\mathrm{V_{E_i+1}}$で、2、3 の場合には$\mathrm{V_{E_i}}$となる。また、回転角度の正負も剰余 0、1 の際は正、2、3 の場合は負となる。向き付け不可能な閉曲面の場合には、2 で割ったときの剰余が 0 のときは「展開図の辺の矢印の先側の頂点」が$\mathrm{V_{E_i+1}}$で回転角度は正、1 のときは$\mathrm{V_{E_i}}$、回転角度は負となる。

グラフの辺の両端の頂点がどちらも同じ展開図の辺上に描かれている場合には、それも同様に対応するもう一方の展開図の辺上にグラフの辺を描く処理を施している。ただし、グラフの辺の情報を保持している Edge.m には、5-1 でも述べたように、具体的な xy 座標の情報は存在せず、辺の両端の頂点の識別番号と、その頂点の original もしくは、何番目の imaginary に繋がっているかのみ情報を持っている。

Gpoint.m は、グラフの頂点が展開図の辺上に乗っているとき、original が乗っている展開図の辺の番号と、imaginary が乗っている展開図の辺の番号、どちらの情報も保持しているので、この識別番号と original/imaginary という情報を頼りに、識別番号に対応する頂点のデータを保持している Gpoint.m から「グラフの頂点が乗っている展開図の辺の番号」を抜き出す。グラフの辺の両端の頂点が乗っている展開図の辺の番号が同じならば、両端のグラフの頂点の、そのグラフの辺が繋がっているのとは逆の頂点の xy 座標を取得し、それを結ぶことで対応するもう一方の展開図の辺上にもグラフの辺を描くことができる。

また、グラフの頂点が展開多角形の頂点上に来た場合には、話は少し複雑になる。そうした場合、Gpoint.m の「original が乗っている展開図の辺の番号」という情報を保持する場所に「グラフの頂点が展開図の頂点上に来た」ということを表すある負の値を収納している。もし辺で繋がっているもう一方のグラフの頂点がいずれかの展開図の辺上にある場合、その辺も同時にいずれかの展開図の辺上に乗っていることは自明であり、その頂点が持っている展開図の辺の番号から、仮想のグラフの辺を描く展開図の辺の番号はわかる。しかし、その展開図の辺の両端にあるどちらの頂点と結びつければ良いのかを判定する処理が必要になるということである（**図 20.**参照）。

グラフの辺が、展開図のどの頂点と同じ位置のグラフの頂点と繋がっているのかは、繋がっているグラフの頂点の座標を、展開図の頂点の座標とひとつずつ照らし合わせることで判定することが出来る。その点とその辺が乗っている展開図の辺の番号との関係から、仮想辺がどちらのグラフの頂点を結んで描くかを決めることができる。

具体的に図.20 を用いて説明すると、グラフの辺が乗っている展開図の辺の番号をE_i、展開図の頂点の番号を(0, field_radius)をV_0、反時計回りにV_1、V_2…と表現するとき、辺が繋がっているグラフの頂点が乗っている展開図の頂点の番号をV_iとして、描く仮想辺が繋がるグラフの頂点をVであらわす。このとき、向き付け可能な閉曲面で、E_iを 4 で割ったときの剰余が 0, 1 のとき、$V_i = E_i \rightarrow V = V_{i+3}$、$V_i = E_i + 1 \rightarrow V = V_{i+1}$となり、剰余が 2, 3 のときは$V_i = E_i \rightarrow V = V_{i-1}$、$V_i = E_i + 1 \rightarrow V = V_{i-3}$となる。

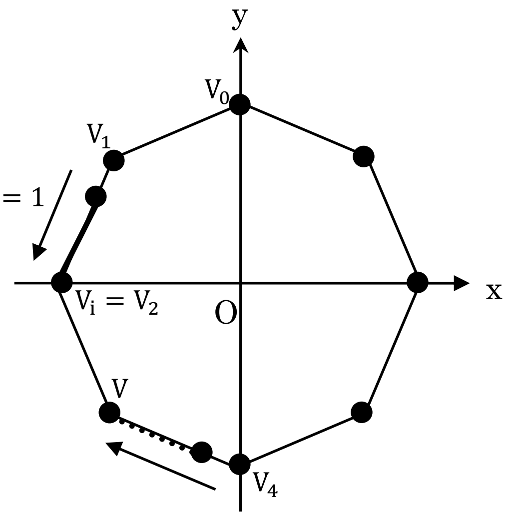

図.20. 展開図の頂点上のグラフの頂点と境界線上の頂点辺に対応する
仮想辺の描画

◇5－4　辺の選択方法

マウスがグラフの頂点を選択していない状態のときにタッチパネルをドラッグすることで、そのドラッグした線と交わっているグラフの辺が選択可能であることはすでに述べた(図.12 参照)。グラフの辺が選択されたと判断された場合、それに対応する Edge.m 内に保持してある、選択されているかどうかという情報を書き換えている。

ではここで、辺が選択されたという状況をどのように判定しているかをここで簡単に述べておく。単純な一次関数の交点の判定である。**図 21.**を参照する。ドラッグするとき最初に触れた部分を(x_1, y_1)、最後に触れた部分を(x_2, y_2)とすると、この線分の一次関数の傾き$a_1 = y_2 - y_1 / x_2 - x_1$、切片$b_1 = y_1 - a_1 x_1$となり、同様に頂点間の辺の一次関数も傾き$a_2 = y_4 - y_3 / x_4 - x_3$、切片$b_2 = y_3 - a_2 x_3$である。このふたつの一次関数の交点の$x = b_2 - b_1 / a_1 - a_2$、$y = a_2 x + b_2$で求めることができる。

このとき、図 21.の Case2 のように交点が辺の延長線上にくる場合がある。そのため、

(x_1,y_1)と (x_2,y_2)、(x_3,y_3)と (x_4,y_4)の組み合わせにおいて、値が大きい方の x 座標を x_l、小さいほうをx_sとする。同様に y 座標も大きいほうをy_l、小さい方をy_sとして、交点(x, y) が$(x_s < x < x_l) \cap (y_s < y < y_l)$であるとき、辺が選択されたととらえる。

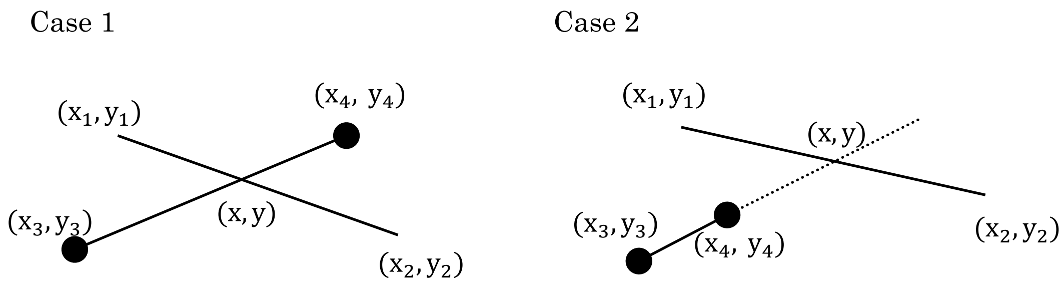

図 21. 辺の交点

◇5－5　展開図の辺の対応図示について

展開図の辺は３章で述べた法則に従って繋がっているものと捉えていて、システム的にそれを実現するようになっていることはここまで述べたとおりである。しかし表示されるウィンドウに明確にどことどこが繋がっているのかを図示しなければ、その対応関係自体がわかりにくいし、結果としてグラフの構造自体も把握しにくい。その傾向は種数が増えれば増えるほど顕著になる。

そこで、この境界線の対応関係を図示するものも、展開図の自動生成と同時に生成されなければならない。その処理がどう行われているのかについて、ここで解説する。

まず展開図よりも一回り大きめの図形を成す頂点群を、展開図の生成とともに作成する。それは「展開図の中心から展開図の頂点までの距離」を少し大きくした状態で、展開図と同じように頂点を決めていけばよいだけであるが、どれぐらい大きくするかは閉曲面の種数との割合で決めている。種数が大きくなるに従って、各辺は徐々に長さが少なくなっていくため、境界線図示のための図形の大きさを一定にしてしまうと、見た目のバランスが悪くなる(図 22.)。

続いて、そうして描いた展開図より一回り大きい図形の各辺の両端を少し削る。どれくらい削るのか、これもまた割合によっているが、割合のもとになる長さは、各辺の両端の x 座標の距離である。これのある一定の割合分だけを各辺の両端の x 座標から内側に詰める。その x 座標でのその辺の一次関数より y 座標を求めることがでる。これで矢印の棒の部分が作られる。

さらに、同様に各辺の x 座標どうしの距離の割合から、矢印のヒレの部分となる線分の長さを決定する。矢印のヒレの部分がどちらの向きになるのかは、各棒の対応する展開図の境界線の番号と、閉曲面が向き付け可能／不可能という情報から割り出せる。

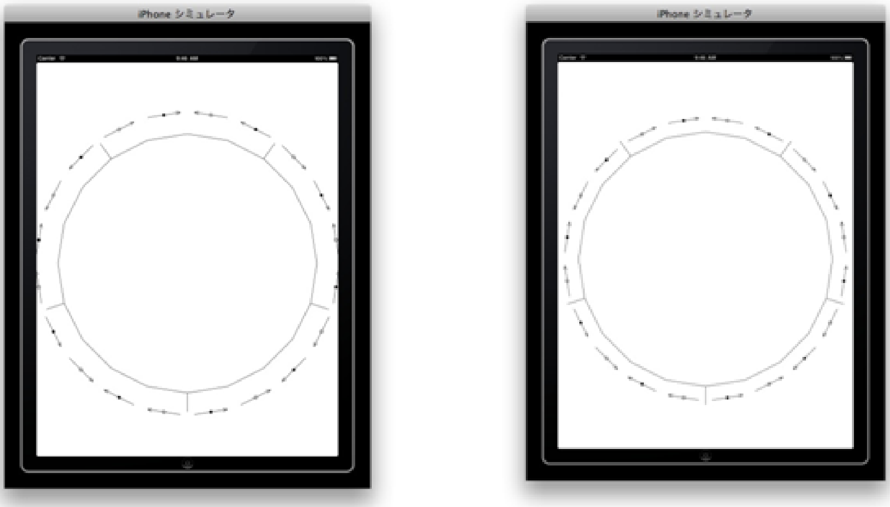

図 22. 境界線の対応図示(左：割合補正なし、右：補正あり)

ヒレとなる線分の長さもまた各辺の両端の x 座標の距離との割合なので、決められた割合分だけヒレをつける方の端の x 座標から内側に詰める。詰めたところの y 座標も同様に一次関数から導き出せる。こうして決まった点を、辺の端の xy 座標を中心として±の両方向に適当な角度分回転させれば、矢印のヒレの部分を描くことが出来る。

こうして矢印ができた後は、その矢印の対応関係を表すために、対応する辺ごとに同じマークを矢印の棒の中心に描けばよい。図.22 に見えるように、●と○で対応関係を表している。さらにひとつのトーラス／射影平面を表せるように、展開図の頂点から区切り線を入れることでさらに閉曲面の構造の可視性を高めている。

◇5−6　吸着処理

頂点を移動している際に、展開図の境界線を超えた場合や、頂点の座標がある程度展開図に近い場合には、操作性の観点から展開図の辺上にグラフの頂点を自動的に乗せてしまう処理を施している。この処理を実行する条件は、点と辺の距離を求める公式を用いて、展開図の辺と頂点との距離を算出し、それがある一定の距離以上近づいた場合と、展開図の各頂点を中心にした、任意に定めた半径距離内に入った場合である。

任意に定めた円の半径内かどうかは、三平方の定理から二点間の距離が算出できるので、これと任意の円の半径を比較すればよい。点と辺の距離(d)を求める場合の公式は、点の座標を(x_0, y_0)、一次関数 $f(x) = ax + b$ とするとき、

$$d = \frac{|y_0 - ax_0 - b|}{\sqrt{1 + a^2}}$$

となる。これが一定の距離以下になった場合に吸着処理を施せばよい。吸着する場所は、そのグラフの頂点を通る展開図の辺と直交する一次関数と、展開図の辺の一次関数の交点の座標とする。

第6章　考察

◇6−1　開発の達成度

2−5 で掲げたとおり、

- ユーザビリティに関する問題点の修正
- あらゆる閉曲面の展開図上にグラフを描くための理論の考案と実装

あらゆる閉曲面の展開図上にグラフを描くための理論の考案と実装するという目的は、五章で詳述した通りのことが実際にコーディングすることができたので、概ね達成されたと見てよい。ただ、7 章でも述べるように、ひとつだけ解決すべき問題を抱えている。それは、グラフの辺の境界線通過である。無論、辺が境界線を通過できないことで、実現することができないグラフの構造があるということはない。しかしながら、そうすることでグラフの頂点を動かすたびにそれに付随する辺も同時に動き、動いている辺が通り抜ける展開図の辺に人間の視覚は注目しやすくなり、結果として展開図の辺の対応関係がより明示されやすくなることが期待できる。これはユーザビリティの向上に一役買うであろう。

とりあえず現段階でのユーザビリティに関しては、実際に使われる状況を想定して開発したので、ある程度満足の行くものになっているとは思うが、より広く利用されていく中で、さらなるユーザビリティの向上を目指せる余地が残っていると思われる。

◇6−2　本ソフトウェアの有用性

展開図の辺がシステム的に繋がっているので、実物の閉曲面にグラフを描くのとほぼ同じ感覚で描いていくことができる。つまり、どの展開図の辺どうしが繋がっているのかという対応関係をほとんど考えることなくグラフを編集することができるので、紙面上でグラフを考えるよりもスムーズにものを考えていくことができるであろう。「Anchor Ring」では今述べたことがトーラスの閉曲面のみで可能であったが、今回はさらに「あらゆる任意の閉曲面」においてそれを達成することができた。これは研究支援ソフトとしては既存のものにはない有用性を兼ね備えているといえるであろう。

さらに、動作するプラットフォームが iPad であるので、既存のアプリケーションよりも、いつでもどこでも利用出来る可能性が広がっており、研究者間のコミュニケーションの一層の活性化につながることが期待できる点も本アプリケーションの有用性であるといえる。

◇6−3　問題点

今回のアプリケーションは、「あらゆる任意の閉曲面の展開図上でのグラフ編集」ということであったが、あまりにも種数が多くなりすぎると、展開図の一辺が短くなりすぎて、閉曲面の展開図としてはほとんど機能をなさない(**図.23**)。ただ、そんなにも大きい種数でのグラフについて考えることはほとんどない、というのが実際の研究者からの助言であるので、そのことで本アプリケーションの有用性が著しく低下するとは考えないが、「大きい種数でのグラフ編集」という事態には対応できないため、この点についてはまた新たに考慮する必要があるであろう。

その他、ユーザビリティや、アプリケーションとしての完成度という観点から不十分だと考えられる点が散見されるので、以後それらについて述べる。

まず、ユーザーインターフェースであるが、現段階では操作方法に関する記述が何もないので、初めてアプリケーションに触れる人にとって非常に不親切になってしまっている。各状況−頂点が選択された状態、辺が選択された状態など−に応じて、実行できる操作を記述することが望ましい。また、今の段階では種数と向き付け可能／不可能の選択がインターフェースからではできなくなってしまっているので、その点についても早急に改善しなければならない。

「gm standard」には搭載されている機能であるが、頂点／辺のラベリングや、色の塗り分け、グラフの拡大縮小、そしてグラフの構造から作られる「面」を自動的に判別する機能など、支援ツールとして搭載しておくべき機能もまだまだ存在する。また、編集したグラフを保存することができないのも問題である。

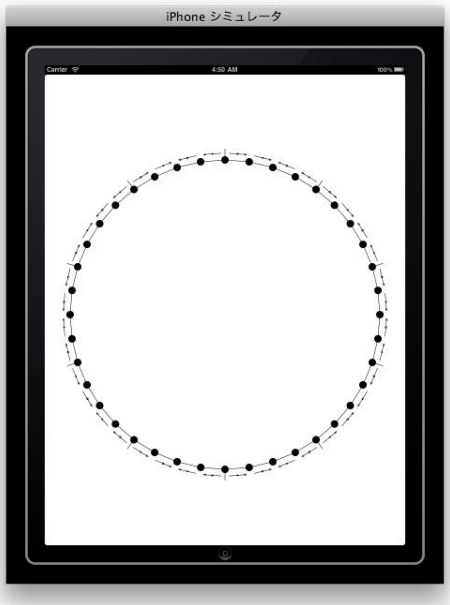

図.23　genus 10, orientable

第7章　将来的課題

◇7−1　グラフの辺が展開図の辺を通過する処理

本アプリケーションでは、グラフの辺が展開図の辺を通過しないようになっている。グラフの辺が展開図の辺上の頂点と結びついていて、その頂点のもう一方を動かした際に、辺は境界線を通過した形で描かれるわけではなく、内部で繋ぎ直す仕様になっている。これは、グラフの辺の展開図の辺の通過の処理をどう実現したらよいかが最後まで問題として残ってしまったためである。そこでここでは、実現に向けて私が考案した基本的な理論的方法について述べる。

◆7−1−1　向き付け可能の場合

図.24 を例にとる。まずグラフの辺が通過した展開図の辺を対象軸として、展開図の中心点を線対称に移動させ、そこを中心として同じ形の仮想上の展開図を描く。さらに、同一視する展開図の辺が一致するように仮想上の展開図をθだけ回転させる。こうすることで、展開図どうしをつなぎあわせることができる。この時、展開図の移動→回転に合わせて、図中のVの頂点も同時に移動回転させる。そのことによって、Vの仮想点V′の xy 座標を決めることができる。この時、V_0とVの二点を通る一次関数と展開図の辺の一次関数の交点が、展開図の辺を通過した際に描画されるべきグラフの辺の通過点となる。V側のグラフの辺の展開図の通過点は、5-3 で述べた方法から導き出せる。

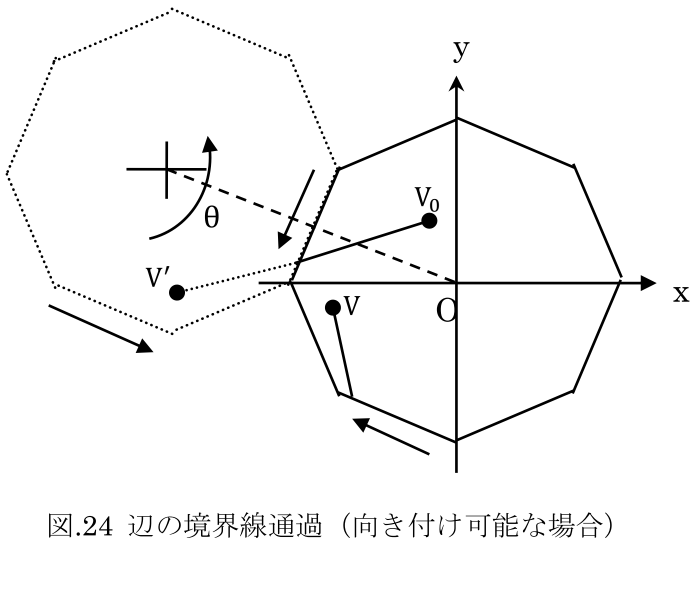

図.24 辺の境界線通過（向き付け可能な場合）

◆7−1−2　向き付け不可能の場合

仮想展開図を作るために、移動→回転というのは同じだが、向き付け不可能の場合には、回転移動させたところで、矢印の向きが合わない。なので、最後に仮想上の展開図の中心点と、本来の展開図の中心点を結んだ直線を対称軸として線対称に仮想頂点V'を移動させる必要がある。この処理を加えることで、境界線の向きをあわせ、仮想頂点の xy 座標が確定する。後は、向き付け可能の場合と同じく、展開図の境界線との交点を求めることで、グラフの辺を通過する処理をさせることができると考える。

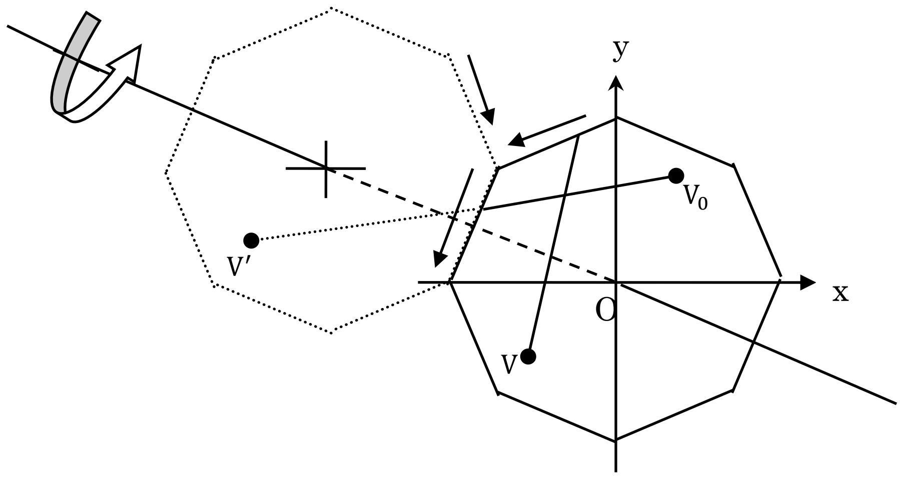

図. 25 辺の境界線通過処理（向き付け不可能の場合）

では、なぜこれらを実装することに問題が生じたかというと、第一に、仮想頂点との間の直線V_0V'と、通過するべき展開図の辺が交わらないという場合がありえる。その場合にどのような処理を施すべきなのかが課題として残る。

また、辺が展開図の辺を通過している、という情報をどのようにデータ構造に組み込んで情報保存させるか、という課題を解ききるまでに今回は時間が足りなかった。というのは、一度だけ展開図の辺を超えるだけならば話はそれほど難しくないのだが、ひとつのグラフの辺が何度も展開図の辺を超え始めたときに、どのようにその通過情報を管理すべきか、今ひとつ納得のいく答えを出すことができなかったということである。

私自身この論文を提出した以降も考えたい問題であるが、研究室の後進にもひとつ考えてもらいたい問題である。

第8章　終わりに

今回作成したソフトウェア「gm standard for iPad」は、より広く研究者に利用してもらえるように配慮したつもりである。

6,7 章でも述べたとおり、このソフトウェアを真に実用的にしようと思ったとき、やらねばならないことは山とある。今回の研究成果を土台に、今後の私自身の研究の成果として、または本研究に興味を持っていただいた方々の尽力でもって、本ソフトウェアが改善されていくことを期待したい。

本ソフトウェアを作成したことによって､グラフへの興味がより一層深いものとなった。今後ともトポロジー、グラフ理論のより一層の発展を願ってやまない。そこに「gm standard for iPad」が微力でも何かしら寄与するものがあったとすれば、この上なく幸せなことである。また、このソフトウェアに触れることで位相幾何学的グラフ理論への興味を持っていただくきっかけになったのならば、それもまた望外の喜びである。

■ 参考文献類

・Web

- 東京大学 河野俊丈教授の授業資料
  http://www.ms.u-tokyo.ac.jp/~kohno/lectures/07.pdf

・書籍

- 根上生也：位相幾何学的グラフ理論入門、横浜図書（2001）
- Christopher Allen & Shannon Appelcline 著, 正　健太朗 訳：
  iPhone アプリ×Web サイト開発入門、日経 BP 社(2009)
- Stephen G. Kochan 著:Programming in Objective-C 2.0, Addison Wesley(2009)

■ 付録

本アプリケーションの理解に必要と思われる部分を抜粋して巻末に載せる。

Vertex.h/Vertex.m (座標(x,y)の保持)
Field.h/Field.m (表示する閉曲面の展開図のデータ保持)
Edge.h/Edge.m (辺のデータ保持)
Gpoint.h/Gpoint.m (頂点のデータ保持)
reportView.h/reportView.m (画面への出力、データ編集のための関数群)

■ 謝辞

本アプリケーションの開発と論文の執筆にあたって、様々な方からの助言、協力をいただきました。

私の指導教官でもある根上生也教授には、学部生の頃から含めると 4 年間、様々な面で大変お世話になりました。プログラミングを学んでいたことと、グラフに関心を寄せていたことをうまくリンクさせて、グラフ理論研究支援ツールの開発という研究のテーマの発端を作っていただいたのは根上教授の助言でした。そこから本稿を仕上げるに至るまで、的確な助言、ご指導いただきました。厚く御礼申し上げます。

また、同じ研究室に所属する先輩諸氏には、親身になってご協力いただきました。特に、環境情報学府後期課程の星野好晃先輩、佐野照和先輩には、グラフ理論の専門的な知識ならびに、私が悩み躓いていることに親身になって相談に乗っていただき、具体的な助言をしていただきました。不束な私を最後まで面倒見ていただいたことに感謝いたします。

それから、研究室の同期は、切磋琢磨できるよい仲間でした。お互いに苦楽を共にし、互いに励まし合うという関係があったからこそここまでやってこられたと思います。ありがとうございました。

最後になりましたが、2 年間の大学院生生活を通してお世話になった先生方、友人はじめ、出会った全ての方々が、研究の集大成としてこの論文を上梓するための私の力の糧になっていると考えます。この場を借りて厚く感謝いたします。