



Title	軸測投影に就いて
Author(s)	村守, 隆男
Citation	教授学の探究, 13, 87-100
Issue Date	1996-03-25
Doc URL	https://hdl.handle.net/2115/13595
Type	departmental bulletin paper
File Information	13_p87-100.pdf



軸測投影に就いて

村 守 隆 男

(北大理学部数学科)

1 はじめに

理系とはいっても、情報専門学科ではない数学科におけるコンピュータ実習教育の実験の一端を紹介するのが、ここでの目的です。

筆者は、1991年10月から現在まで、北大理学部数学科の計算機演習1,2(前期,後期,共に選択科目)を担当しています。この科目は、数学の教職課程の必修単位ということもあって、履修者は、数学科の学部3,4年生,大学院生,他学科の学生,社会人(科目等履修生)と幅が広く、学年,学科,学生,社会人の枠を越えたところが特徴です。現在の所,学内にある全学共用の情報処理教育センターを毎週一回2時間利用させて頂いています。計算機は,日立のパソコンH3020を,言語はターボパスカルv7.0を使用しています。

前期の計算機演習1は基礎編で,計算機に慣れることと,簡単な数値計算を覚えることが目的です。後期の計算機演習2は応用編で,二次曲線,二次曲面のグラフィック画面への描画を目的としています。ここでは,1993年10月~1994年3月に行なった計算機演習2のうちで,空間における座標系の設定についてお話ししたいと思います。

コンピュータグラフィクスでは,平面の座標系の選び方については,ディスプレイ画面の中央を原点として,簡単に直交座標系をつくることができます。しかし,空間においては,直交座標系のとり方は,いろいろあり,また作る図形にふさわしい図法を選択することが必要であり,これが作品の善し悪しにも影響します。

二次曲面の描画には,透視図法よりは,平行投影図法によるほうが,曲面の形を認識するうえでよいといわれています。さらに,平行投影法の中でも軸測投影は,多くの図形を一定の方向から観察するのに適しています。軸測投影について,D. F. RogersとJ. A. Adams[3]は,同次座標を用いて説明しています。(末尾参考文献の[3]参照。以下,同様。)

この,小論では,平行投影法の原理からどのような数学的手続きを経て,空間の直交座標系を投影面上に作ったら良いか,といったことについて,写像と図形を対応させながら説明します。これに関する適当な文献が,著者の調べた範囲では,みあたらなかったため,その方面へのいくばくかの寄与になればと筆を取った次第です。

2節では,指導,教材と授業計画,評価について説明します。3節では,軸測投影の設定方法について,解説します。4節では,具体的な使用例として,パスカルのプログラムへの設定例を掲載します。

謝辞 筆者は1994年4月から教育学部教育方法論講座の須田勝彦先生の担当する毎週火曜日午後のゼミに出席させて頂いており,活発な研究心に大きな刺激を受けています。本論文を投稿する機会を提供して下さいました須田先生及びゼミのメンバー,大学院生の前田輪音,高橋哲男,学部生の石川高行,留学生のTi Ti Yi(ティンティンイー,ミャンマー),金寧姫(キムヨン

ヒ、韓国)のみなさんに心からお礼申し上げます。

2 指導, 教材と授業計画, 評価

前期の計算機演習1で, 数値計算については, ある程度わかっていることを仮定して講義と実習が進められます。2時間の授業の内, 前30分がその日の課題についての講義と解説です。教育センターには, 一度に60名位が講義を受けることができ, video 装置, モニターを使つての説明も可能な視聴覚講義室が用意されています。

コンピュータ教育には, このような設備の整つた講義室での講義は大変に効果的です。後半90分がパソコンによる実習です。

コンピュータ実習教育は, 例えてみれば, 車の運転と同じようなもので, 技術的な側面が大きなウエイトを占めています。理論がわかっているだけでは車が動かないように, これと同じことが, コンピュータについてもいえます。習うより慣れることが, 学びの近道といえます。

マシンの操作, マシンとの応答に興味と関心が湧くように指導ができれば, 授業の進行はきわめてスムーズです。なにについても, いえることですが, 教える側に, 教える事への, (教育)愛と喜びがあれば, 細かいことであまり, 心配することはありません。教えられる側の学生のほうが技術的には, 進んでいるということが, この分野ではよくあることです。私も, 教えることを通して, 随分いろいろのことを学生から教わりました。教えるとは, 実は, 学ぶことの謂いであると思っています。

ひとつの, 指導上の留意点は, 学生によって技術面での差が大きいということがあげられます。技術が入った教育では, 十分技術力のある学生を, 積極的にとりあげて, ta (teaching assistant) とするとか, チーム を作つて chief にするとか, 規定課題, 自由課題について優秀作品を表彰するなどの, 工夫が効果的でした。

教材について話しを進める前に, グラフィックスの設定について触れておきます。グラフィックスの授業は図形が描画できるとカラフルで, いままでのプログラミングの苦勞も忘れるくらい楽しいのですが, 残念なことにコンピュータグラフィックスの世界には, 統一の規格といったものはありません。機種ごとに, グラフィックスの設定方法が異なっています。従つて, それぞれの機種に応じたライブラリ, ユニットの製作は, 教師の役目となります。

さて, 教材の選び方は, やさしいもの, 簡単なもの, 興味をひくものからだんだん高度なもの, 精巧なものへ, という一般的な, 教育的配慮が必要です。以下にあげるのは, ひとつの実践例です。

第一講目 (序論1)

オリエンテーション: H3020のシステム構成, Ms-Windows, Ms-Dos, ターボパスカル, コンピュータグラフィックスの概略説明

規定課題の説明: 二次曲線, 二次曲面の描画するプログラムについての説明

評価の説明: login 時間点, 毎回の Exercise 点, 自由課題点, 規定課題点 各100点による総合点 (240点以上合格, 可: 240~279点, 良: 280~319点, 優: 320~400点)

評価のポイント: login 時間点を重視する。作品は創造性を最も重視する。

教材: Dia. pas, Line. pas (ダイアの模様)

課題: 点, 線, 円, 色の描画の練習

*最も基本的な学習

第二講目（序論 2）

教材：Stereo. pas（立体視）

課題：簡単なステレオグラムの作成。または、正多角形の作成

*興味をもって取り組み、幾つかの良い作品が作成された。

第三講目（平面 1）

教材：Signal. pas, Pendulum. pas（信号、振り子）

課題：車、船、鳥、飛行機、ボール等の簡単な動画の作成

*ピタゴラスの定理のシミュレーションの作成は良い課題の一つです。

第四講目（平面 2）

教材：Plane. pas（平面の直交座標、サイクロイド、リサージュ曲線）

課題：複素一次関数の描画。または、地図、設計図の作成

* Plane. pas は平面の図形描画のライブラリです。

第五講目（空間 1）

教材：Space. pas（三軸測投影、円柱ラセン、双曲的放物面）

課題：隠線消去法の研究（中塗り法、max-mini 法）

*隠線消去法は、重要であるが、短い時間でのプログラムの作成は無理のようだ。

第六講目（空間 2）

教材：Pers. pas（透視図法）

課題：正多面体の回転、拡大、縮小等の動画の作成

*ここでは、z-軸方向からの透視図法を紹介する。

第七講目（トピックス 1）

教材：Tree. pas, Cantor. pas, Koch. pas（フラクタル図形）

課題：ヒルベルト曲線、マンデルブロ集合、ジュリア集合等の作成

*トピックス1,2は、いろいろの話題を適宜利用するとよい。例えば、一次変換による図形の移動の描画、スプライン曲線の表示等。

*フラクタルについての video を鑑賞する。高い関心呼び、大変効果があった。

第八講目（トピックス 2）

教材：Contour. pas（等高線図）

課題：カッシーニの卵形等の陰関数のグラフを描画するプログラムの作成

*複素関数の絶対値関数の等高線図の描画

第九講目（シミュレーション 1）

教材：Sinewave. pas (正弦曲線のシミュレーション)

課題：グラフィクスを用いた簡単なゲーム，アニメのプログラムの作成

*動きのある図形はグラフィクスの授業を，一層楽しくする。

第十講目 (シミュレーション 2)

教材：Kranz. pas, Logistic. pas (X マスカード，ロジスチック方程式の分岐図)

課題：地球儀等のシミュレーションのプログラムの作成

*カオスの分岐図の描画は，コンピュータグラフィクスによる最近の成果です。

* Kranz. pas は，簡単なシミュレーションと sound コマンドを用いた，音の出るプログラムであり，コンピュータ支援の総合的視聴覚教育の一つの方向の試みである。これは，学生の興味を惹起することに大いにあずかった。

成績表は每学期末公開掲示しています。以下が，今回掲示したものの写しです。

*** Computer 2 passer ***

Jan. 24 1994

no.	time pt.	ex. pt.	reg. pt.	free. pt.	sum	rank
S050101	90	80	80	85	335	A
S040105	95	65	70	90	320	A
S050107	100	65	65	80	310	B
S040118	100	90	65	90	345	A
S030133	65	65	65	90	285	B
S050129	100	80	65	100	345	A
S050133	100	100	100	100	400	A
S050131	80	85	65	80	310	A
S050157	90	65	65	80	300	B
S050158	90	90	80	100	360	A
S040147	65	65	65	90	285	B
S050159	80	80	65	80	305	B
S043716	65	65	95	95	320	A
S050148	90	65	65	80	300	B
S050147	100	100	65	100	365	A
S020138	95	65	65	95	320	A
S050121	70	65	65	70	270	C
S040128	100	65	65	90	320	A
S050146	85	85	65	85	320	A

優秀作品：

torus. pas (トーラス曲面)：兩宮貴之君 (s050101)，石川晶君 (s050107)，水野克也君 (s050158)

tree. pas (フラクタル図形)：蔭平和憲君 (s040118)，遠山悦郎君 (s050146)

stereo. pas (立体視)：北見 健君 (s050129)

quad. pas (二次曲面)：小西弘将君 (s050133)，西田顕郎君 (s043716)

ball. pas (ころがる ボール) : 小崎 誠君 (s050131)
voronoi. pas (ボロノイ図形) : 西田顕郎君 (s043716)
pers. pas (透視図法) : 西川達雄君 (s050147)
minimal. pas (極小曲面) : 乗木 裕君 (s020138)

反省及び所感 :

1. 講義の後の実習が, 諸事情が重なり, 一, 二階で, 三室に分離して行なうことになった。巡回して, 質問を受けたり, 個人指導に努めたが, 実習室の数, 収容人数, 機器の設置方法, 視聴覚教材の拡充等, 施設面での改善を痛切に感じた。
2. グラフィクスには, 最低限度の数値計算の知識は, 必要である。また, パスカル v7.0は, 統合環境なので, self-contained であるとはいっても, Ms-Dos, Windows の知識は多少必要である。これらが不足のため, 今期の単位をあきらめた学生が, 何人か出た。この方面の follow up は 今後の課題である。
3. テーマを一つにしぼって系統だてて実習を試みることを, 模索しているところである。グラフィクスについていえば, 例えば, unix 上で, gnuplot をもちいて, カオス (chaos) を描画する等, 考えられる。
4. コンピュータの利用環境は, 現在発展途上にあり, 今後ますます, 拡大すると思われる。教育現場において, 授業, 質疑, 指導, 連絡, 評価, 研究等にコンピュータが果たす役割は, 増えることはあっても, 減ることは, ないと思われる。しかし, このような時代趨勢においてもなお, 教師一人一人の人間性が, 教育の最大の要 (かなめ) であることに, 全く変わりはないと考える。

3 軸測投影の設定方法

この節では, 平行投影 (parallel projection) のうちの正射影 (orthogonal projection) を用いて, 空間図形をどのようにして投影面に実現するかを10の小節に分けて順次説明する。

1. 記号の約束 1

考えている空間と投影面の枠を次のように置く :

R^3 : 3次元ユークリッド空間

$\{O : e_1, e_2, e_3\}$: R^3 の正規直交枠 (Orthonormal Frame), O は原点

$R^2 (\subset R^3)$: 2次元ユークリッド空間

$\{\bar{O} : \bar{e}_1, \bar{e}_2\}$: R^2 の正規直交枠 (Orthonormal Frame), \bar{O} は R^2 の原点

2. 軸測投影の定義

R^3 の直角座標軸上に点 X, Y, Z をとり, X, Y, Z の張る平面を $\pi(X, Y, Z)$ ($=\pi$ と略記する) で表す。点 $P \in R^3$ の, 平面 π の上への正射影を点 P の軸測投影 (axonometric projection) と呼ぶ。そして, この平面 π を投影面, OX, OY, OZ を主軸と呼ぶ。cf. Fig. 1
軸測投影は R^3 から R^2 への中心アフィン写像 (central affine mapping) の特別の場合と考えられる。

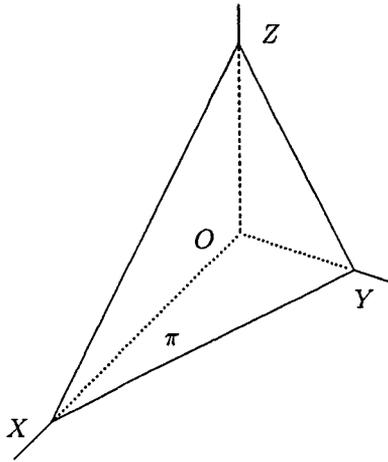


Fig. 1

3. アフィン写像の定義 (cf. [1], p298)

写像 $f: R^3 \rightarrow R^2$ がアフィン写像であるとは、線形写像 $f_*: TR^3 \rightarrow TR^2$ が存在して、 $\forall P \in R^3, \forall v \in TR^3$ に対して

$$f(P+v) = f(P) + f_*(v)$$

i. e.,

$$f_*(v) = \overrightarrow{f(P)f(P+v)}$$

が成り立つことをいう。

4. 記号の約束 2

アフィン写像 f , 及び f に付随する線形写像 f_* によって、平面 π に写される諸量を次のように定義する：

$$f(O) := O^*$$

i. e.,

$$f_*(e_j) = \sum_{i=1}^2 a_{ij} \bar{e}_i, \quad j=1, 2, 3,$$

$$f_*(e_1, e_2, e_3) = (\bar{e}_1, \bar{e}_2)A,$$

ここに行列 A , ベクトル $\overrightarrow{OO^*}$ を次のように置く；

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix}$$

$$\overrightarrow{OO^*} = : b = \sum_{i=1}^2 b_i \bar{e}_i.$$

角度 θ , ϕ を次のように定義する：

$$\begin{aligned}\theta &= \angle(f_*(e_1), O^*, f_*(e_2)) \\ \phi &= \angle(f_*(e_1), O^*, f_*(e_3))\end{aligned}$$

e_1, e_2 と e_3 の, f_* による縮小率を次のように置く：

$$\begin{aligned}\lambda &= \|f_*(e_1)\| \\ \mu &= \|f_*(e_2)\| \\ \nu &= \|f_*(e_3)\|\end{aligned}$$

写像 f が正射影のときには, λ, μ と ν はそれぞれ, 0 以上 1 以下の値をとる。

5. 3 軸測投影の定義

組 $\{O^*: f_*(e_1), f_*(e_2), f_*(e_3)\}$ を R^2 における回転と平行移動によって移して

$$\begin{aligned}(C1) \quad & f_*(e_3) // \bar{e}_2, \text{ (same direction)} \\ (C2) \quad & b=0, \text{ (central affinity)}\end{aligned}$$

としたものを改めて $\{O^*: f_*(e_1), f_*(e_2), f_*(e_3)\}$ と書き表すことにする。

さらに, 縮小率 λ, μ, ν と角度 θ, ϕ の間に, 次の軸測条件を仮定する； (cf. [2], p362)

$$(C3) \quad \begin{aligned}\lambda^2 &= \frac{-\cos(\theta+\phi)}{\sin(\theta)\sin(\phi)} \\ \mu^2 &= \frac{\cos(\phi)}{\sin(\theta)\sin(\phi+\theta)} \\ \nu^2 &= \frac{\sin(\theta)}{\sin(\phi)\sin(\theta+\phi)}\end{aligned}$$

条件 (C1) ~ (C3) を満足する軸測投影 $f: R^3 \rightarrow R^2$ を **3 軸測投影** (*trimetric projection*) と呼び, R^2 をその**投影面**と呼ぶ。

3 軸測投影は平行投影の定義から明らかなように θ, ϕ を決めれば一意に決定する。特に, $\theta = \phi$ のときには, **2 軸測投影** (*dimetric projection*), $\theta = \phi = 120^\circ$ のときには, **等測投影** (*isometric projection*) と呼ぶ。

6. 3 軸測投影の座標変換公式

$\forall x \in R^3$ について

$$\begin{aligned}\vec{Ox} &= \sum_{j=1}^3 x_j e_j \\ \vec{Of(x)} &= \sum_{i=1}^2 \bar{x}_i \bar{e}_i\end{aligned}$$

と置く。このとき, (C2) により

$$\begin{aligned}\overrightarrow{Of(x)} &= \overrightarrow{O^*f(x)} \\ &= f_*\left(\sum_{j=1}^3 x_j e_j\right) \\ &= \sum_{i=1}^3 \sum_{j=1}^2 a_{ij} x_j \bar{e}_i,\end{aligned}$$

すなわち $\bar{x}_i = \sum_{j=1}^3 a_{ij} x_j$, $i=1, 2$, i.e.,

$$\begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \end{pmatrix} = A \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

7. 変換の表現行列の決定

Fig. 2 より直ちに

$$\begin{aligned}f_*(e_1) &= \lambda(\cos(90+\phi)\bar{e}_1 + \sin(90+\phi)\bar{e}_2) \\ f_*(e_2) &= \mu(\cos(270-\theta-\phi)\bar{e}_1 + \sin(270-\theta-\phi)\bar{e}_2) \\ f_*(e_3) &= \nu\bar{e}_2,\end{aligned}$$

これより

$$A = \begin{pmatrix} -\lambda \sin(\phi) - \mu \sin(\theta + \phi) & 0 \\ \lambda \cos(\phi) - \mu \cos(\theta + \phi) & \nu \end{pmatrix}$$

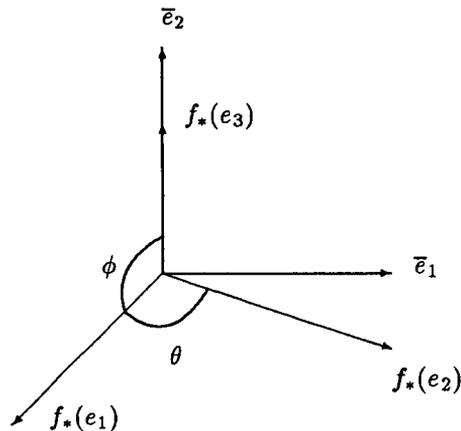


Fig. 2

8. 行列 A を角度 α , β を使って書換える

角度 α , β を Fig. 3 のように取るとき,

$$f_*(e_1) = \lambda(\cos(180+\alpha)\bar{e}_1 + \sin(180+\alpha)\bar{e}_2)$$

$$f_*(e_2) = \mu(\cos(-\beta)\bar{e}_1 + \sin(-\beta)\bar{e}_2)$$

$$f_*(e_3) = \nu\bar{e}_2$$

であるから、これより直ちに、前述の表現行列 A

$$A = \begin{pmatrix} -\lambda \cos(\alpha) & \mu \cos(\beta) & 0 \\ -\lambda \sin(\alpha) & -\mu \sin(\beta) & \nu \end{pmatrix}$$

と書き表すことができる。ここで、スクリーン座標への変換 (x 軸についての折り返し)

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

を A に左から施すと、結局、行列 A は

$$A = \begin{pmatrix} -\lambda \cos(\alpha) & \mu \cos(\beta) & 0 \\ \lambda \sin(\alpha) & \mu \sin(\beta) & -\nu \end{pmatrix}$$

のように書換えられる。

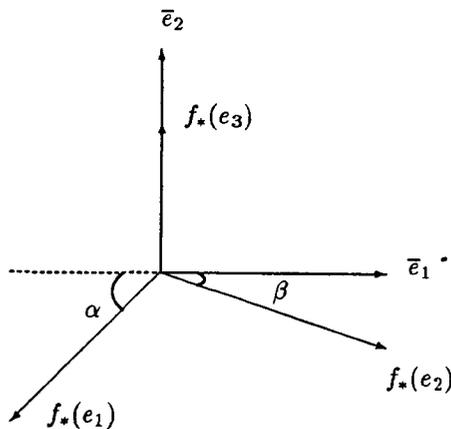


Fig. 3

9. 別解 (初等幾何的解法)

今、スクリーン座標、 R^3 のデカルト座標を次のように置く：

$P(X, Y)$: スクリーン座標 (直交系)

$O(x_0, y_0)$: スクリーン座標の原点

$P(x, y, z)$: R^3 のデカルト座標 (直交系)

Fig. 4 より座標 X, Y は

$$X = x_0 + OE,$$

$$Y = y_0 - PE \quad (\text{スクリーン座標への変換公式})$$

また、容易な計算により

$$\begin{aligned}
 OE &= CB - CA \\
 &= CB - FO \\
 &= \mu CD \cos(\beta) - \lambda OC \cos(\alpha) \\
 &= -\lambda x \cos(\alpha) + \mu y \cos(\beta)
 \end{aligned}$$

$$\begin{aligned}
 PE &= DP - DE \\
 &= \nu z - (BD + BE) \\
 &= \nu z - (\mu CD \sin(\beta) + \lambda OC \sin(\alpha)) \\
 &= -\lambda x \sin(\alpha) - \mu y \sin(\beta) + \nu z
 \end{aligned}$$

以上より、次の変換公式を得る；

$$\begin{aligned}
 X &= x_0 - \lambda x \cos(\alpha) + \mu y \cos(\beta) \\
 Y &= y_0 + \lambda x \sin(\alpha) + \mu y \sin(\beta) - \nu z,
 \end{aligned}$$

これより

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + A \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

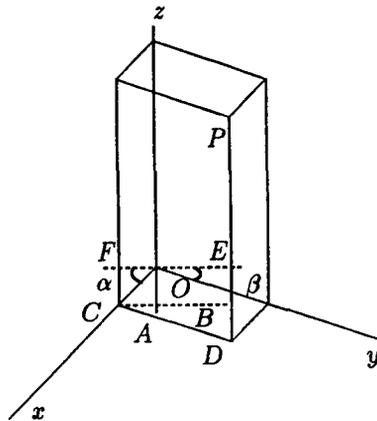


Fig. 4

10. プログラム Space. pas への適用

第四節で掲載するプログラム **Space. pas** は $\alpha=35^\circ$, $\beta=25^\circ$ の 3 軸測投影の場合である。ゆえに、この時 $\theta=120^\circ$, $\phi=125^\circ$ 。これらを (C3) に代入すると

$$\lambda=0.772\dots,$$

$$\mu=0.855\dots,$$

$$\nu=0.821\dots,$$

を得る。角度の組 (α, β) として、このほかに、 $(30^\circ, 20^\circ)$, $(45^\circ, 15^\circ)$, $(30^\circ, 15^\circ)$, $(20^\circ, 10^\circ)$ がよく用いられる。**Fig. 5** は、一辺が10の立方体の $\alpha=35^\circ$, $\beta=25^\circ$ の 3 軸測投影図である。

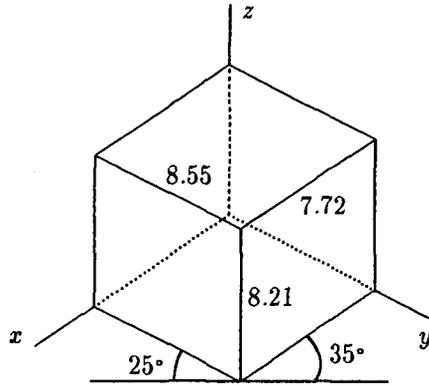


Fig. 5

4 パスカルのプログラムへの設定例

プログラムの作成には、[4]が参考になる。ここでは、第五講目の教材のサンプルプログラム Space. pas の全文を掲載する。手続き名等からその内容が明らかであり、また、プログラムそのものも平明であるので、コメントは最小限にとどめ、詳しい注釈は、省略した。

```

{-----}
      S P A C E
      Nov.15 1993
{-----}
PROGRAM Space(output);
      {Space.pas}
USES
  Graph;
CONST
  radian=pi/180.0;
  xo=319;yo=239; {origin}
  xu=10;yu=10;zu=10;{unit}
VAR
  Gd,Gm:Integer;
  alpha,beta,r1,r2,r3,ca,cb,sa,sb:Real;
  xi,yi,zi:Real;
  iang1,iang2:Real;
  j,k:Integer;

  { ----- Procedures ----- }

PROCEDURE Initialize;
  BEGIN
    Gd:=Detect;
    InitGraph(Gd,Gm,'d:\#tp#\#bgi');
    IF GraphResult<>grOK THEN halt(1)
  END;{Initialize}

PROCEDURE Init_Set;{Trimetric Projection}
  BEGIN
    alpha:=35*radian;beta:=25*radian;
    r1:=0.772;r2:=0.855;r3:=0.821; (* 三軸測投影の設定 *)
    ca:=Cos(alpha);cb:=Cos(beta);
    sa:=Sin(alpha);sb:=Sin(beta)
  END;{Init_Set}

```

```

PROCEDURE Line_(x1,y1,z1,x2,y2,z2:real);
VAR
  xs1,ys1,xs2,ys2:Integer;
BEGIN
  xs1:=round(xo-r1*x1*ca*xu+r2*y1*cb*yu);
  ys1:=round(yo+r1*x1*sa*xu+r2*y1*sb*yu-r3*z1*z2);
  xs2:=round(xo-r1*x2*ca*xu+r2*y2*cb*yu);
  ys2:=round(yo+r1*x2*sa*xu+r2*y2*sb*yu-r3*z2*z2);
  Line(xs1,ys1,xs2,ys2)
END;{Line_}

PROCEDURE Line_Rel(dx,dy,dz:real);
BEGIN
  LineRel(round(xo-r1*dx*xu+r2*dy*cb*yu),
          round(yo+r1*dx*sa*xu+r2*dy*sb*yu-r3*dz*z2))
END;{Line_Rel}

PROCEDURE Line_To(x1,y1,z1:real);
VAR
  xs,ys:Integer;
BEGIN
  xs:=round(xo-r1*x1*xu+r2*y1*cb*yu);
  ys:=round(yo+r1*x1*sa*xu+r2*y1*sb*yu-r3*z1*z2);
  LineTo(xs,ys)
END;{Line_To}

PROCEDURE Move_(leng:real);
BEGIN
  Line_(xi,yi,zi,xi+leng*Cos(iang2*radian)*Sin(iang1*radian),
        yi+leng*Cos(iang2*radian)*Cos(iang1*radian),
        zi+leng*Sin(iang2*radian));
  xi:=xi+leng*Cos(iang2*radian)*Sin(iang1*radian);
  yi:=yi+leng*Cos(iang2*radian)*Cos(iang1*radian);
  zi:=zi+leng*Sin(iang2*radian)
END;{Move_}

PROCEDURE Move_Rel(dx,dy,dz:real);
BEGIN
  MoveRel(round(xo-r1*dx*xu+r2*dy*cb*yu),
          round(yo+r1*dx*sa*xu+r2*dy*sb*yu-r3*dz*z2))
END;{Move_Rel}

PROCEDURE Move_To(x1,y1,z1:real);
VAR
  xs,ys:Integer;
BEGIN
  xi:=x1;yi:=y1;zi:=z1;
  xs:=round(xo-r1*x1*ca*xu+r2*y1*cb*yu);
  ys:=round(yo+r1*x1*sa*xu+r2*y1*sb*yu-r3*z1*z2);
  MoveTo(xs,ys)
END;{Move_To}

PROCEDURE Put_Pixel(x,y,z:real;pixel:word);{Cartesian Coord.}
VAR
  xs,ys:Integer;{Screen Coord.}
BEGIN
  xs:=round(xo-r1*x*ca*xu+r2*y*cb*yu);
  ys:=round(yo+r1*x*sa*xu+r2*y*sb*yu-r3*z*z2);
  PutPixel(xs,ys,pixel)
END;{Put_Pixel}

PROCEDURE Turn_(ang1,ang2:real);
BEGIN
  iang1:=iang1+ang1;
  iang2:=iang2+ang2
END;{Turn_}

```

```

{ ----- Cartesian Coordinates ----- }

PROCEDURE Axis;
CONST
  tx=25;ty=25;tz=23;{length of axis,ディスプレイ画面の補正}
VAR
  x1,x2,x3,x4,y1,y2,y3,y4,x5,y5,x6,y6:Real;
BEGIN
  x1:=xo-tx*xu; y1:=yo-ty*sb*yu;
  x2:=xo+tx*xu; y2:=yo+ty*sb*yu;

  y3:=yo-ty*yu; y4:=yo+ty*yu;

  x5:=xo-tx*ca*xu; y5:=yo+tx*sa*yu;
  x6:=xo+tx*ca*xu; y6:=yo-tx*sa*yu;

  Line_( 0, 0, 0,25, 0, 0);{x axis}
  Line_( 0, 0, 0, 0,25, 0);{y axis}
  Line_( 0, 0, 0, 0, 0,23){z axis}
END;{Axis}

PROCEDURE Scale;
VAR
  i:Integer;
BEGIN
  FOR i:=-22 TO 22 DO
    BEGIN
      Put_Pixel(i,0,0,Red);Put_Pixel(0,i,0,Red);
      Put_Pixel(0,0,i,Red)
    END
  END;{Scale}

{ ----- Sample Procedures ----- }

PROCEDURE Helix(a,b:real);
VAR
  i:Integer;
BEGIN
  FOR i:=0 TO 7200 DO
    BEGIN
      Put_Pixel(a*cos(i/10*radian),a*sin(i/10*radian),
        b*i/10*radian,Red)
    END
  END;{Helix}

PROCEDURE Surface;
CONST
  dx=0.2; dy=0.2;
VAR
  x,y,z:Real;
  h,i:Integer;

  FUNCTION S_(x1,y1:real):real;
  BEGIN
    S_ :=(x1*x1-y1*y1)/10 {hyperbolic paraboloid}
  END;{S_}

```

```

BEGIN
  FOR h:=-10 TO 10 DO
    BEGIN
      xi:=h;yi:=-10;zi:=S_(xi,yi);
      Move_To(xi,yi,zi);
      FOR i:=-10 TO 10 DO
        BEGIN
          Line_(h,i,S_(h,i),h,i+1,S_(h,i+1));
          Line_(h,i+1,S_(h,i+1),h+1,i+1,S_(h+1,i+1));
          Line_(h+1,i+1,S_(h+1,i+1),h+1,i,S_(h+1,i));
          Line_(h+1,i,S_(h+1,i),h,i,S_(h,i))
        END
      END
    END;{Surface}

{ ----- Main Program ----- }

BEGIN
  Initialize;
  Init_Set;
  Axis; Scale;

  {Circular Helix:
  Helix(5,1);
  readln;
  CloseGraph
  END.}

  {Surface:}
  Surface;

  readln;
  CloseGraph
  END.

```

参 考 文 献

- [1] 伊原信一郎, 河田敬義: 線形空間・アファイン幾何, 岩波書店 (1990)
- [2] 岩田至康: 幾何学大辞典第二卷, 槇書店 (1974)
- [3] D. F. Rogers and J. A. Adams: *Mathematical Elements for Computer Graphics*(second edition), McGraw-Hill, Inc (1989)
- [4] Turbo Pascal Version 7.0 Programmer's Reference, Borland International, Inc. (1992)