

STEAM 教育に基づく高大連携

—Python による美しい幾何学図形の生成—

梅村 祥之*・直川 耕祐*

(令和5年11月24日受付)

High school-university collaboration based on STEAM education

—Generation of beautiful geometric shapes using Python—

Yoshiyuki UMEMURA and Kosuke NAOKAWA

(Received November 24, 2023)

概要

STEAM 教育に基づく高大連携において、Python プログラムを用いたコンピュータグラフィックスにより美しい幾何学図形を生成する授業教材を開発し、授業とグループワークを実施した。幾何学図形は、定規と歯車を使って美しい図形を描く子供用玩具のスピログラフをコンピュータ化し、複数の歯車を自由な回転スピードで連結して図形を生成できるようにしたものである。教材の中には、三角関数に基づく図形生成原理、Python プログラミング、パラメータの変更による様々な形の生成に関する内容が盛り込まれている。受講した生徒へのインタビュー、生徒が作成した発表会用ポスタの内容、発表会での発表者の発言内容から、美しい図形の創作を通じてコンピュータグラフィックスの技術を学ぶことに対して非常に成功した。幾何学図形と数学との関係を理解させることに対して、やや成功した。

キーワード：STEAM 教育幾何学図形コンピュータグラフィックススピログラフ Python

1 はじめに

近年、高等学校教育の中に STEAM 教育が導入されつつある。文部科学省の Web ページに「STEAM 教育等の各教科等横断的な学習の推進について」という以下の記事が掲載されている^[1]。

AI や IoT などの急速な技術の進展により社会が激しく変化し、多様な課題が生じている今日、文系・理系といった枠にとらわれず、各教科等の学びを基盤としつつ、様々な情報を活用しながらそれを統合し、課題の発見・解決や社会的な価値の創造に結び付けていく資質・能力の育成が求められています。文部科学省では、STEM (Science, Technology, Engineering, Mathematics) に加え、芸術、文化、生活、経済、法律、政治、倫理等を含めた広い範囲で A を定義し、各教科等での学習を実社会での問題発見・解決に生かしていくための教科等横断的な学習を推進しています。

広島工業大学高等学校では広島工業大学と連携した STEAM 教育を2022年度に開始した^{[2][3]}。教育プログラムの大まかな流れは次のとおりである。広島工業大学の12学科は各学科の特徴を持ったテーマに沿った授業コンテンツを作成する。それを用いて、大学教員が講義を行い、また、講義内容に関係した課題を説明し、その後、各生徒が課題解決のためのグループワーク（以下、創作活動と称する）を行い、その結果をポスタにまとめて、ポスタセッションを行う。本論文では、情報工学科で設定したテーマとそれに沿って開発した授業コンテンツについて述べる。

テーマ設定は STEAM 教育の趣旨に沿って設定した。まず、その考え方で、開発したコンテンツの概略を述べ、次に、それを使った創作活動とポスタセッションの実施状況及び、創作活動中に行った生徒へのインタビュー結果とポスタセッションでの本課題に対する生徒の印象に関する発言内容について報告する。

2 企図

情報工学科の特徴を持ったテーマとして、コンピュータグラフィックスによる幾何学図形の生成のテーマを設定する。子供用玩具として、定規を使って美しい図形を描くスピログラフ^[5]という玩具が1965年から現在に渡って市販されている^{*1}。図1にスピログラフの写真とそれを使って図形描画している様子を示す。

コンピュータで描く場合、物理的な定規の持つ制約を超えて、複数の歯車を連結させることができるし、歯車（以下、歯車を円と称する）の回転スピードを自由に設定でき、それによって、複雑で美しい図形の生成が可能である^[6]（図2）。

*1 SPIROGRAPH は米国企業ハズプロの登録商標である。

* 広島工業大学情報学部情報工学科

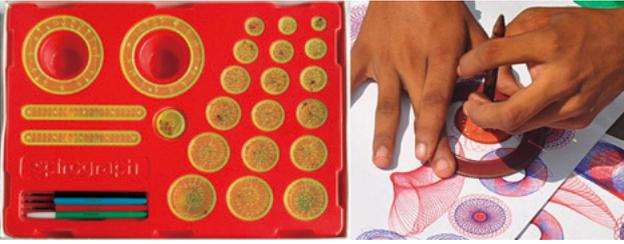


図1 スピログラフの商品と使用の様子（Wikipedia 英語版, Spirograph より転載）

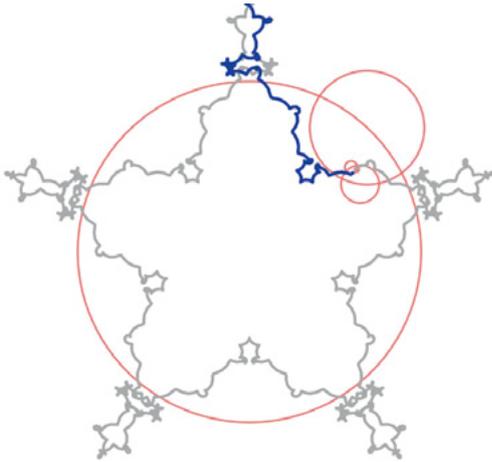


図2 スピログラフのコンピュータ版による描画例

現在、高校の授業科目情報Iでコンピュータプログラミングの学習が行われている。高大連携先の広島工業大学高等学校ではプログラミング言語としてPythonを採用している。そこで、授業コンテンツ開発にあたって、Pythonを使ったコンピュータグラフィックスにてスピログラフ図形の描画を行う^{*2}。授業コンテンツの企画にあたって、STEAM教育の各要素を次の考え方に従って盛り込む。

Science, Technology, Engineering

STEAM教育の分野でscienceは自然科学の意味で使われている。technologyは自然科学の法則に基づいて、何らかのものを生み出すための方法と解釈できる。engineeringは、生み出すものを人々の生活や産業に役立つものに限定するとともに、関連するtechnologyを学問体系に編成したものと解釈できる。先にSTEAM教育の趣旨を述べた文部科学省の考え方からすると、scienceよりも、technologyないしengineeringに重点を置くべきと考えられる。そして、technologyないしengineeringのうち、engineering教育は工学系大学の専門教育そのものであるため、学問体系としてのengineeringを取り扱うことはしない。

ただし、一部、技術開発における工学的的方法論を取り入れる。

すなわち、現象を観察して、法則性を見つけ出し、幾何学図形生成に応用することである。

一方、technologyとしてはコンピュータグラフィックスの技術を取り扱う。

^{*2} スピログラフを用いた描画図形であるエピサイクルに対してここではスピログラフ図形と表現する。

Art

STEAM教育のAは当初、art（芸術）を意味していたが、社会人文科学を意味するliberal artsも含むように拡張された。本論文が対象とする連携先の広島工業大学高等学校においてはさらに「あそび心」という解釈を含め、「好きなものにこだわりを持って取り組める」ことを目指している^[2]。

スピログラフ図形生成においては、円の個数、各円の半径、各円の回転スピードを様々に変更して、様々な美しい図形を創作することができる。この点から、芸術あるいは遊び心の要素を含むと考えられる。

Mathematics

スピログラフ図形をコンピュータで描くには、直交座標の知識、三角関数の知識、ベクトルの知識が必要である。この段階で数学の要素を含んでいる。

スピログラフの形を決めるパラメータによっては奇妙な形が発生することがある。そのような現象を対象として、なぜ、奇妙な現象が発生したのかの現象解明に取り組むための課題を設ける。現象解明にあたっては、論理的な思考力が必要となる。

3 教材開発—概略—

Pythonプログラムの統合開発環境として普及しているjupyter labのNotebook形式^[8]で教材を提供する。近年、人工知能に関する研究開発が活発になされ、数多くの研究用ソフトが公開されている。人工知能の研究分野ではプログラム言語としてPythonを使うことが多い。公開するソフトのプログラムコードとドキュメントを提供する枠組みとして、使い勝手の良いjupyter labのNotebook形式が最もよく使われている。以上を踏まえ、本教材もjupyter labのNotebook形式で提供する。

開発した教材の一部を図3に示す。章立ての構造をサポートしている。図に示す例では、「■綺麗な図形の生成」が章にあたり、その中に「●原理」という節がある。それぞれの単位で内容を折りたたんだり、開いたりできる。プログラムが網かけ部分に示される。この部分を実行でき、実行結果が表示される。

教材の章立てを図4に示す。教材の章立ては2部構成である。第1部は講義形式を用いて、コンピュータグラフィックスの基礎から始めてスピログラフ図形描画までを解説する。第2部は生徒の創作活動の題材として演習問題を6題提供するものである。美しい図形を創作するという

File Edit View Run Kernel Tabs Settings Help
+ - X K O > C > Markdown Notebook

■ 綺麗な図形の生成 ----- 見出し

● 原理 ----- 説明

my_plot を変更してx軸、y軸、格子、目盛を表示しないようにした関数 my_plot2 を定義する。綺麗な図形は複雑な形状のため、それらを表示しない方が見やすい。

my_plot2 には、もう1つ変更点がある。ax.plot(x,y,... を ax.plot(y,x,... とx、座標を逆にしている。こうすることによって、角度0がy軸の正の方向で、角度が増加するに従って時計回りに回るようになる。

ここまでで準備が完了した。それでは、重ね合わせる円の数を5個に増やして、一番最初に描いた綺麗な図形を描画する。

```
[4]:
from kstean import * # 初期設定
theta = np.arange(0,2*math.pi,1/2*math.pi/360)
r1 = 1.0; r2 = r1/3.0; r3 = r2/3.0; r4 = r3/3.0; r5 = r4/3.0
# r1:地球の半径 r2:月の半径 r3:月の衛星の半径
# r4:月の衛星の衛星の半径 r5:月の衛星の衛星の衛星の半径
n1 = 1.0; n2 = -n1*4.0; n3 = -n2*4.0; n4 = -n3*4.0; n5 = -n4*4.0
# n1:地球の速度 n2:月の速度 n3:月の衛星の速度
# n4:月の衛星の衛星の速度 n5:月の衛星の衛星の衛星の速度
x = r1 * cos(n1 * theta) + r2 * cos(n2 * theta) + r3 * cos(n3 * theta) + \
    r4 * cos(n4 * theta) + r5 * cos(n5 * theta) # 5個の円を重ねる(x座標)
y = r1 * sin(n1 * theta) + r2 * sin(n2 * theta) + r3 * sin(n3 * theta) + \
    r4 * sin(n4 * theta) + r5 * sin(n5 * theta) # 5個の円を重ねる(y座標)
fig, ax = plt.subplots(); ax.plot(y,x,'k'); ax.set(aspect=1);
ax.axes.xaxis.set_visible(False); ax.axes.yaxis.set_visible(False); plt.show()
```

プログラム

実行結果

図3 Python 統合環境 jupyter lab を用いた教材の画面表示

| |
|---------------------------------|
| 第1部 講義 |
| 第1章 グラフ描画の基礎 |
| 第1節 四角形 |
| 第2節 円 |
| 第2章 綺麗な図形の生成 |
| 第1節 原理 |
| 第2節 様々な形の生成 |
| 第3節 原理に立ち戻る |
| 第4節 r と n を変更して様々な形を生成するアニメーション |
| 第2部 創作活動 |
| 課題(1) 2個の円からなる図形 |
| 課題(2) 5個の円からなる図形 |
| 課題(3) 5個の円からなる図形のアニメーション表示 |
| 課題(4) 回転対称でない図形 |
| 課題(5) 内側で回転させる場合と外側で回転させる場合 |
| 課題(6) スピードを整数でなくす |

図4 教材の章立て

“Art”の側面を備えた課題と、生成した形に対して、なぜそのような形になるのかを、工学分野で行われる現象解明のアプローチに基づいて解析する課題を用意している。以下、その詳細を述べる。

4 教材開発—第1部講義—

4.1 第1部第1章グラフ描画の基礎

第1部「第1章グラフ描画の基礎」は、基礎概念から始め、円を描画するまでを扱う。円を描画の前に、直交座標系の導入から四角形の描画までを扱う「第1節四角形」を設ける。以下、四角形を描画するまでのプログラミングを学習するための教材の構成を図5を参照しながら説明する。コンピュータで平面図形を描く際に、直交座標系を用いた座標指定によって、平面上の点が定まり、それを結んで線ができ、線を結んで幾何学図形ができるという基本概念を学習者に理解させる。次に、Pythonにおける標準的なグラフ描画モジュールである matplotlib^[7]を用いて与えられたx,y座標に対する点を描画するプログラム(図中に記載)が実行文5文で実現できることを理解させる。以下、1点から2点、2点からそれらを結んだ線、4本の線を結んだ四角形とステップバイステップでプログラミングを理解させる。

次は「第2節円」である。多角形の辺の数を増やして円を描画するまでの教材の構成を図6を参照しながら説明する。コンピュータで円を描くには、多角形の辺の数を多くして、人間が見て滑らかな曲線と見えるようにする方法を用いる。この概念を理解させたのち、多角形の頂点の座標を求めるために三角関数が必要であることを理解させる。Pythonの数値計算モジュールnumpyが提供する関数を用いたプログラムで計算できることを理解させたのち、多角形の辺の数を増やしてゆくと見た目には円と見える図形を描画できることを理解させる。

4.2 第1部第2章綺麗な図形の生成

第2章が教材の中心となるスピログラフ図形生成の原理とパラメータ変更による様々な形状の生成に関する説明部分である。円を多重に重ねて図形を生成することに対する基本原理はベクトルの合成である。図7を使ってこの原理を理解させる。この原理に基づいた描画をPythonで行うためのプログラムを提供している(図8)。

これ以降、図9に示すステップで複雑な図形の生成に向かって説明を進める。実際には、図に示すステップの間にもいくつかのステップを設けているが省略する。

5 教材開発—第2部創作活動—

生徒の創作活動の題材となる演習問題を6題提供する。美しい図形を創作するという“Art”の側面を備えた課題と、生成した形に対して、なぜそのような形になるのかを、工学分野で行われる現象解明のアプローチに基づいて解析する課題である。

はじめに美しい図形を創作する課題3題のうちの1つ課題(1)について述べる。講義で説明したスピログラフ図形生成プログラムの中の円の半径と回転スピードを決めるパラメータを様々に変更することによって、様々な形の図

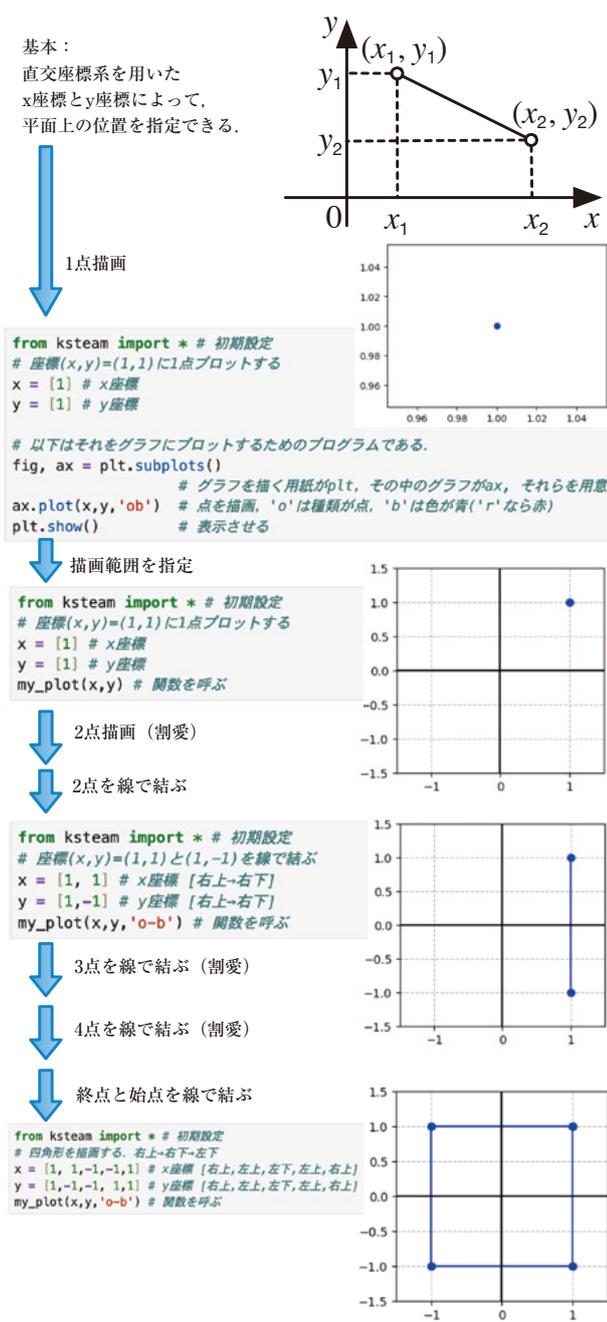
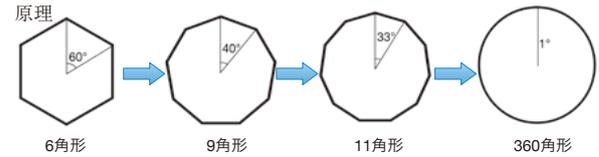


図5 座標系を用いた点の描画から四角形の描画に至るプログラミングの流れ

形が生成される。各自の感性に従って、美しいと感じられる図形を創作するのがこの課題である。

「課題(1) 2個の円からなる図形」では、図10のプログラムを用意し、プログラム上のどこに変更すべきパラメータが記載されているかをコメント文で示している。2箇所の数字を変更することによって、様々な図形を生成できる。プログラムと実行例(図10)を示す。

次に、生成された変わった形に対して、なぜそのような形になるのかを、工学分野で行われる現象解明のアプローチに基づいて解析する課題について述べる。高校生が短期間で行うことを前提としているため、本来の現象解明と呼ぶに相応しい内容、すなわち、仮説を打ち立て、実験に



三角関数を用いて多角形の頂点の座標を求める

```
from numpy import sin, cos; import numpy as np; import math
print('cos 30°=', cos(30*2*math.pi/360))
print('√3/2 =', np.sqrt(3)/2) # √3 / <math>2</math>
cos 30° = 0.8660254037844387
√3/2 = 0.8660254037844386
```

多角形の描画プログラム

```
# 円を描画する。座標データの個数, x座標, y座標の値をプリント
# 1周を20分割
from numpy import sin, cos; import numpy as np; import math
theta = np.arange(0,2*math.pi,2*math.pi/20)
# arange(開始, 終了, 増分)は系列をNumPy配列に入れる。増分は0.5°
x = cos(theta) # theta(角度, ラジアン)の要素ごとのcos値のNumPy配列
y = sin(theta) # theta(角度, ラジアン)の要素ごとのsin値のNumPy配列
print(len(x)); print(np.round(x,2)); print(np.round(y,2)) # 点数, x座標列,
my_plot(x,y,'o-b') # 関数を呼ぶ
```

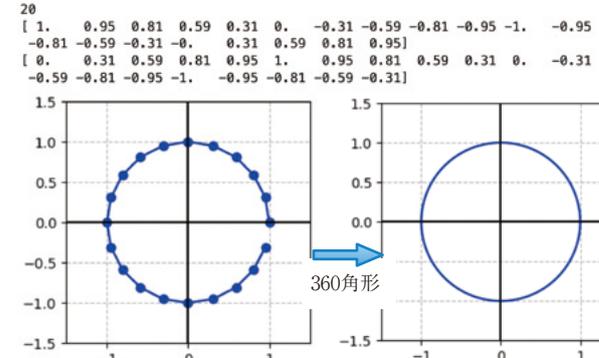


図6 多角形から円に至るプログラミングの流れ

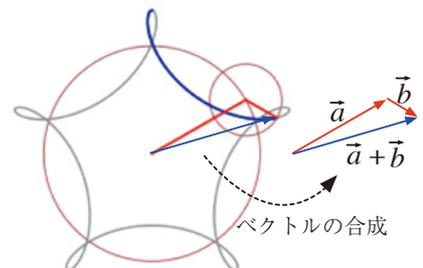


図7 ベクトルの合成に基づくスピログラフ描画の原理

```
from ksteam import * # 初期設定
theta = np.arange(0,2*math.pi,1/2*math.pi/360)
# 角度の配列。0.25°刻みで0°から360°の1440点からなる配列。
# コンソールで len(theta) を実行して点数が1440点であることを確認しよう
# print(theta) としてthetaの内容を確認しよう
r1 = 1.0 # 地球の半径
r2 = r1/3.0 # 月の半径
n1 = 1.0 # 地球のスピード
n2 = -n1*4.0 # 月のスピード, マイナスは反対回り
x = r1 * cos(n1 * theta) + r2 * cos(n2 * theta) # 2つの円を + して重ねている
y = r1 * sin(n1 * theta) + r2 * sin(n2 * theta)
# xとyはそれぞれ, x座標, y座標に関する1440点を持つ配列
# コンソールで len(x) を実行して点数を調べ, print(x)として内容を調べよう
my_plot2(x,y) # 図形描画
```

図8 ベクトルの合成に基づくスピログラフ描画のプログラム

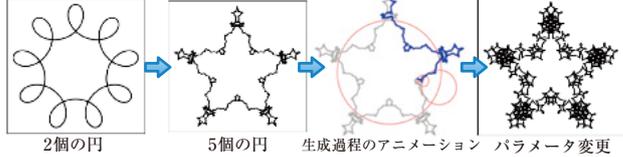


図9 第1部第2章後半綺麗な図形の生成の各ステップ

```
from ksteam import * # 初期設定
theta = np.arange(0,2*math.pi,1/2*math.pi/360)
# 角度の配列. 0.25°刻みで0°から360°の1440点からなる配列.
# コンソールで len(theta) を実行して点数が1440点であることを確認しよう
# print(theta) としてthetaの内容を確認しよう
r1 = 1.0 # 地球の半径
r2 = r1/4.0 # 月の半径 - これを変更する
n1 = 1.0 # 地球のスピード
n2 = -n1*5.0 # 月のスピード, マイナスは反対回り - これを変更する
x = r1 * cos(n1 * theta) + r2 * cos(n2 * theta) # 2つの円を + して重ねている
y = r1 * sin(n1 * theta) + r2 * sin(n2 * theta)
# xとyはそれぞれ, x座標, y座標に関する1440点を持つ配列
# コンソールで len(x) を実行して点数を調べ, print(x)として内容を調べよう
my_plot2(x,y) # 図形描画
```

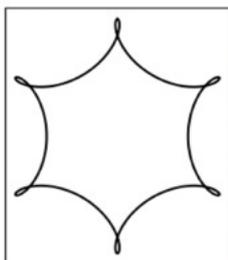


図10 「課題(1) 2個の円からなる図形」のプログラムと生成図形

よって、仮説が成立するか否かを検証し、その結果に基づいて仮説を修正し、再び、実験を行い、仮説検証するというサイクルまでは実施しない。その代わりに、2ステップからなる簡易的な現象解析のステップを用意する。「課題(4) 回転対称でない図形」がその枠組みで学習する課題である。教材内で課題を提示する部分の表示画面を図11に示す。

■ 課題(4) 回転対称でない図形

「原理」の最後に、回転対称でない図形を生成した。変数を変更して、回転対称でない、別の図形を生成しよう。

どういった条件が整えば回転対称でなくなるのか、様々な条件を試して調べ、考察を深めよう。簡単ではないため、ステップを踏んで1歩ずつ確かめてゆこう(科学における正攻法の進め方)。

```
from ksteam import * # 初期設定
theta = np.arange(0,2*math.pi,1/2*math.pi/360)
r1 = 1.0; r2 = r1/3.0+1.5; r3 = r2/3.0; r4 = r3/3.0+2; r5 = r4/3.0 #-- 変更
n1 = 1.0; n2 = -n1*4.0/2; n3 = -n2*4.0; n4 = -n3*4.0+2; n5 = -n4*4.0/2 #-- 変更
x = r1 * cos(n1 * theta) + r2 * cos(n2 * theta) + r3 * cos(n3 * theta) +\
  r4 * cos(n4 * theta) + r5 * cos(n5 * theta)
y = r1 * sin(n1 * theta) + r2 * sin(n2 * theta) + r3 * sin(n3 * theta) +\
  r4 * sin(n4 * theta) + r5 * sin(n5 * theta)
fig, ax = plt.subplots(); ax.plot(y,x,'k'); ax.set(aspect=1);
ax.axes.xaxis.set_visible(False); ax.axes.yaxis.set_visible(False); plt.show()
print('r1=',r1,'r2=',r2,'r3=',r3,'r4=',r4,'r5=',r5)
```



r1= 1.0 r2= 0.5 r3= 0.16666666666666666 r4= 0.11111111111111111 r5= 0.03703703703703703

図11 「課題(4) 回転対称でない図形」の説明文, プログラム, 生成図形

図に示される回転対称ではない図形を現象解明の対象とする。この現象を前にして、次の2ステップによって解析を進める。

- Step 1 : 条件を単純化する
- Step 2 : 単純化した要素について、様々な値で、形を調べる

Step 1 は条件を単純化して、現象を生じさせる要因を明確化する過程である。元の図形は5個の円を使っている。円の個数を減らしてゆく。1個の円は回転対称であり、5個の円で、回転非対称であるため、2個、3個、4個を調べ、円の数が最も少なく、かつ、回転非対称の場合が、最も単純な場合である。それを調べることを求めている。提供するプログラムの中に、それぞれの円の半径を0にするための、変更箇所をコメント文で示している。それに基づいて、変更して実行すれば、対応する図形が作成されて表示される。3個の円にした場合の画面表示を図12に示す。この場合でも回転非対称が続いている。2個の円で実行してみると回転対称となり、最小個数は3個であることがわかる。

Step1 条件を単純化する。

上の例では5個の円を使っている。円の数を減らす。最も少ない円の数を探る。そのためには、先のプログラム中の r2 から r5 のどれかを0にすれば良い(半径0は円がないのと同じ)。

```
from ksteam import * # 初期設定
theta = np.arange(0,2*math.pi,1/2*math.pi/360)
# r1=1.0; r2=0.5; r3=0.16666666666666666; r4= 0.11111111111111111; r5= 0.03703703703703703
r1=1.0; r2=0.5; r3=0.16666666666666666; r4= 0; r5= 0 #-- r2~r5のいずれかを0にする
n1 = 1.0; n2 = -n1*4.0/2; n3 = -n2*4.0; n4 = -n3*4.0+2; n5 = -n4*4.0/2
x = r1 * cos(n1 * theta) + r2 * cos(n2 * theta) + r3 * cos(n3 * theta) +\
  r4 * cos(n4 * theta) + r5 * cos(n5 * theta)
y = r1 * sin(n1 * theta) + r2 * sin(n2 * theta) + r3 * sin(n3 * theta) +\
  r4 * sin(n4 * theta) + r5 * sin(n5 * theta)
fig, ax = plt.subplots(); ax.plot(y,x,'k'); ax.set(aspect=1);
ax.axes.xaxis.set_visible(False); ax.axes.yaxis.set_visible(False); plt.show()
```

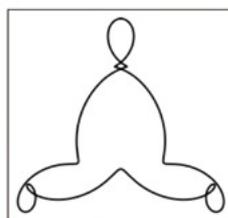


図12 「課題(4) 回転対称でない図形」における解析過程 Step 1 の実行画面

Step 2 は円の数を最低限にして単純化したあと、残ったパラメータに関してパラメータスタディを行うステップである。残ったパラメータは3個の円の半径と回転スピードの6個である。6個のパラメータ全てを一挙に変更すると混乱するため、円の半径を初期値のまま固定して、3個の回転スピードを様々に変化させて、回転対称性がどうなるかを調べるように指示している。図13に回転スピードを変更した結果、回転対称図形となった場合の表示画面を示す。

この他にも現象解析に重きを置いた課題として2課題を用意する。ここでは説明を省略する。

```
Step2 単純化した要素について、様々な値で、形を調べる
Step1の結果、半径r1からr3までの3つの円で非対称が生じると分かった場合には、r1,r2,r3は変化させず、n2とn3を変化させて、どの値の組み合わせなら非対称になるかを調べる。

from ksteam import * # 初期設定
theta = np.arange(0,2*math.pi,1/2*math.pi/360)
r1=1.0; r2=0.5; r3=0.16666666666666666;r4=0;r5=0
n1 = 1.0;
n2 = -2 # - 変更 (n2 = -n1*4.0/2)
n3 = 10 # - 変更 (n3 = -n2*4.0)
n4 = 0; n5 = 0
x = r1 * cos(n1 * theta) + r2 * cos(n2 * theta) + r3 * cos(n3 * theta) + \
    r4 * cos(n4 * theta) + r5 * cos(n5 * theta)
y = r1 * sin(n1 * theta) + r2 * sin(n2 * theta) + r3 * sin(n3 * theta) + \
    r4 * sin(n4 * theta) + r5 * sin(n5 * theta)
fig, ax = plt.subplots(); ax.plot(y,x,'k'); ax.set(aspect=1);
ax.axes.xaxis.set_visible(False); ax.axes.yaxis.set_visible(False); plt.show()
```

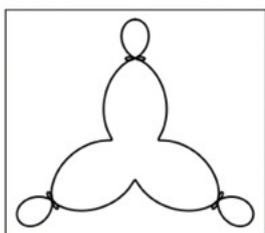


図13 「課題（4）回転対称でない図形」における解析過程 Step 2 の実行画面であり、回転対称な図形が生成されている

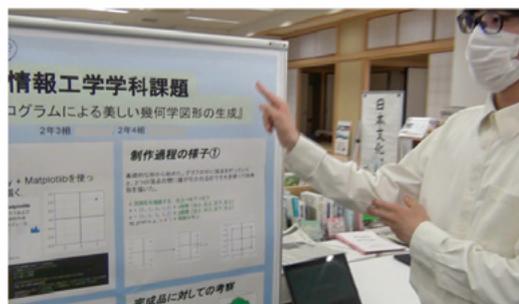


図15 講義中の授業風景およびポスタセッションにおける発表風景

6 授業および教育効果

6.1 授業

開発した教材を用いて、広島工業大学高等学校のKSTEAM 類型 CL コース^[4]の2年生を対象に授業を実施した。実施時期は初回講義が2023年6月6日であり、最後の発表会が9月19日であった。6回の授業の構成を図14に示す。講義の聴講者は74名で、本テーマの創作活動に参加して実施し、発表会で発表した生徒はその中の10名であった。講義およびポスタセッションの写真を図15に示す。

- 講義50分
- 創作活動100分×3回
 - 教材の詳細説明、図形生成の演習1
 - 図形生成の演習2、現象解明の演習、グループごとの実習テーマの選定
 - グループごとの実習テーマ実施
- ポスタ発表用資料作成100分
- 発表会100分

図14 授業の構成

提示した6課題のうち、美しい図形を生成する3課題については全員が達成できた。現象解明に関係する3課題のうち、難易度の低い1課題は全員が達成でき、難易度の高い2課題は取り組みを行うグループがなかった。難易度の高い2課題のうちの1つは前述の課題（4）である。

6.2 インタビュー調査

創作活動の最後である第3回の時間中、2名からなるグループごとに、各グループ約10分の時間を使い、本テーマ

に参加しての感想をインタビュー調査した。質問事項をあらかじめ用意せず、臨機応変にやりとりした。発言のメモをとり、インタビュー後に文章化した。そのため、著者の解釈が混入している。

- 本テーマのどういうところに魅力を感じたか。
 - 元々、情報系の内容に興味があった（半数以上の人の回答）
 - プログラミングが好きだから
 - 美術的な題材に興味があった
 - スピログラフに触れたこともあり、親近感を持っていた
 - 数学に関係したプログラムに興味があった
- モノづくりの楽しさを感じたか。
 - 規則性や法則性を見つけるのが楽しかった
 - モノづくりを大々的に行なったと言えるほどの実感はないが、できた図形に美しさを感じ、それが楽しさにつながった
 - パラメータを変えて自分でいろいろな試行錯誤を行うと、様々な形が生成されて楽しさを感じた
 - 少しパラメータを変えるだけで全く異なる図形が生成されることがあり、刺激的であった
 - 教材の中で提供されていないコンピュータグラフィックス処理を自発的に盛り込んだ。プログラムの修正によって様々な機能を実現できるところに楽しさを感じた。具体的には、色をつける処理などである
- 数学的な内容を理解できたか。
 - 図形と三角関数のつながりを理解し、実感できた
 - 図形と三角関数のつながりを理解するのが難しかった

- 難しかったところはどこか。
 - 通常授業が与えられたことを覚えるタイプの授業に対し、能動的に取り組まないといけないところが難しかった
 - 1年生で学習した三角比を忘れていたため、その先を学ぶのが大変であった
- 変わった図形が生成される現象に対する現象解明の内容（課題4など）は面白いと感じたかに対して、多くの生徒が興味を示さなかったが、興味を示した生徒も少数いた。

6.3 発表内容および発表者の発言内容

ポスタセッションにおいて、5グループの発表が行われた。ポスタに掲載された美しい幾何学図形の例を図16に示す。

ポスタ記載の内容および発表者の発言内容の中から、特筆すべき点を述べる。

- 作品創作に対する感想
 - 創作活動の第一印象に関するポスタの記載：「ずっとやられているような中毒性がある」
 - 創作活動を終えての感想に関するポスタの記載：「めっちゃめっちゃおもしろい」「わくわくが止まらなかったので挑戦し続けることが可能」
 - 様々なパラメータを試していて、偶然、人間の顔のようなパターンが発生した作品のポスタセッションでの発表者の発言：「個人的に出来作です！」（図17）
- 教材で与えていない新機能の追加
 - 図形の内部を指定した色で塗りつぶす処理をPythonプログラムの中に（教員の力を借りず）生徒独自で追加（図18）
 - 様々な形の複数のスピログラフ図形を（手作業で）重ね合わせて美しい作品を創作（図19）

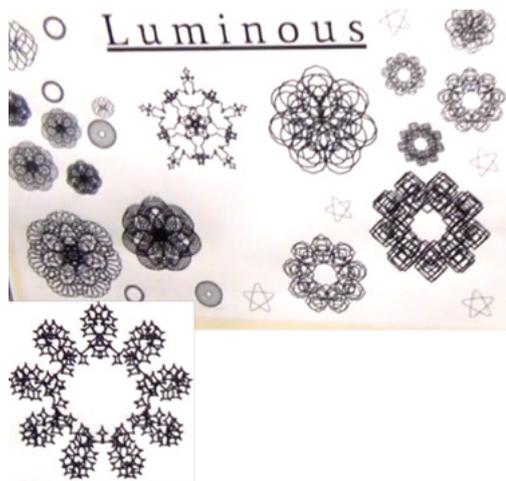


図16 ポスタに掲載された美しい幾何学図形



図17 顔のパターンが生じた作品



図18 プログラムを作成して図形の内部を緑色で塗りつぶした作品

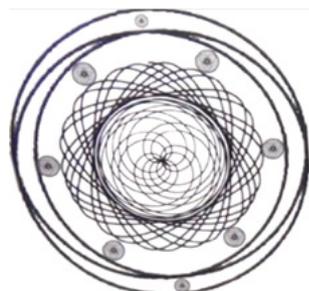


図19 複数の図形を重ね合わせた作品

7 考察およびまとめ

STEAM 教育に基づく高大連携授業として、Python プログラムを用いたコンピュータグラフィックスにより美しい幾何学図形を生成する授業コンテンツを開発し、授業を実施した。

受講した生徒へのインタビュー、生徒が作成した発表会用ポスタの内容、発表会での発表者の発言内容を集約した本取り組みの成否に対する筆者の印象をまとめる。

- 美しい図形の創作を通じてコンピュータグラフィックスの技術を学ぶことに対して非常に成功した。
- 幾何学図形と数学との関係を理解させることに対して、やや成功した。
- 不思議な現象に対して、工学的なアプローチで現象解明する試みは成功しなかった。

STEAM 教育の構成要素の各々に対する達成度に対する筆者の印象を述べる。ただし、science については他の要素との重複が大きいため、単独の要素としては扱わない。

- technology に関し、コンピュータグラフィックスとプログラミング技術を学習できたということから達成できたと考ええる。
- engineering に関し、工学的アプローチによる現象解明の取り組みを盛り込み、試行した結果、未達成に終わった。
- art に関し、美しい図形生成に対して、ポスタで「ずっとやられているような中毒性がある」と述べられるなど、ほとんどの生徒の満足度が高かったことから達成できたと考ええる。
- mathematics に関し、幾何学図形と数学との関係を理解させることに対して、つながりを実感させることができたことから、ある程度達成できたと考ええる。

謝辞

高大連携での大学側取りまとめ広島工業大学電子情報工

学科升井義博教授、高校側取りまとめ広島工業大学高等学校平原豪人教諭、情報工学科テーマの授業運営を担当された森田将征教諭に感謝いたします。授業サポートをしていただいた広島工業大学情報工学科梅村研究室升田康平君、岩崎楓也君に感謝いたします。本テーマを選択して創作活動を積極的に行っていただいた広島工業大学高等学校 CL コースの生徒の皆さんに感謝いたします。

参考文献

- [1] 文部科学省:STEAM 教育等の各教科等横断的な学習の推進, 入手先 <https://www.mext.go.jp/a_menu/shotou/new-cs/mext_01592.html> (参照 2023-10-14)
- [2] 広島工業大学高等学校:学校案内パンフレット, 入手先 <<https://www.kodaikoko.ed.jp/zennichi/admission/img/2023gakkouannai.pdf>> (参照 2023-10-14)
- [3] 広島工業大学高等学校: K-STEAM 類型—STEAM 1 の授業, 入手先 <https://www.kodaikoko.ed.jp/zennichi/school_life/k01-2022-learning.html> (参照 2023-10-14)
- [4] 広島工業大学高等学校: K-STEAM 類型—CL コース, 入手先 <https://www.kodaikoko.ed.jp/zennichi/school_life/k-steam.html> (参照 2023-10-14)
- [5] Wikipedia:Spirograph, 入手先 <<https://en.wikipedia.org/wiki/Spirograph>> (参照 2023-10-14)
- [6] C.J.Chen:Fun math art (pictures) -benice equation, 入手先 <<http://benice-equation.blogspot.com/2012/01/fractal-spirograph.html>> (参照 2023-10-14)
- [7] The Matplotlib development team: matplotlib, 入手先 <<https://matplotlib.org>> (参照 2023-10-14)
- [8] Project Jupyter: jupyter, 入手先 <<https://jupyter.org>> (参照 2023-10-14)