

## 研究論文

# Excel VBA へのグラフィックス描画命令の実装

総合情報基盤センター 教授 木原 寛

**概要:** プログラムの作成や実行が容易な環境として、表計算ソフトウェア Excel の VBA に注目し、基本グラフィックスとタートルグラフィックス描画機能等をモジュールとして追加し、教育用プログラミング学習環境として利用可能とした。

**キーワード:** プログラミング、VBA、Excel、グラフィックス

## 1. はじめに

大学の一般教育においてプログラミング教育を実施する上での困難や問題点としては、次のような点が挙げられる。

- ・プログラミングに興味を抱かせたり、それを持続させることが難しい。
- ・プログラムを作る際に、プログラムの実際の実行過程をイメージさせにくい。
- ・文法や約束事の説明に時間を取られ、実用的なプログラムを作成できる段階まで至らないため、折角獲得した知識や技術が定着しにくい。

上記の問題点を克服するためには、初心者がプログラミングに興味を持ちやすく、プログラムの作成や実行が容易なシステムが望まれる。

この条件を満たすシステムの一つとして、既に広く普及しており、Microsoft Windows でも Mac OS でも利用可能な Excel VBA に着目し、グラフィックス描画命令を追加してプログラミング学習用の環境を構築した。

## 2. Excel VBA の特徴

Excel に付属の VBA (Visual Basic for Applications) は、表計算機能等の自動実行や複雑な処理機能を付加するための仕組みであるが、純粋プログラミング言語としても充分利用可能なことは、以前から知られている。<sup>1) 2)</sup>

Excel VBA の長所としては、次のような点が挙げられる。

- ・Excel に付属しているため新たに購入する必要がなく、ほとんどの PC 実習室で利用可能である。

- ・主要な部分の規格がバージョンごとに大きく異なる。
- ・インタープリタで利用しやすく、結果の確認が容易である。
- ・IDE (統合開発環境) が高機能で、入力補助機能や入力時の文法チェック機能がある。
- ・高機能なデバッガを備えている。
- ・手続き型であり、構造化が可能である。
- ・豊富な組み込み関数を有し、比較的容易に実用的なプログラムを記述できる。
- ・フォームの利用やイベント・ドリブン型プログラムの記述が容易である。

逆に、短所としては次のような点が挙げられる。

- ・マクロのセキュリティレベルの設定が必要である。
- ・オブジェクト指向の階層構造を持っているため、オブジェクトやメソッドをある程度憶える必要がある。
- ・言語仕様が洗練されていない点がある。
- ・グラフィックス描画機能がない。

とくに、グラフィックス描画命令などが標準的な機能として提供されていない点は、初心者用のプログラミング学習環境としては大きなハンディとなる。

森口の著書では、グラフィックスとして直線、矩形、円などを扱っているが、標準的なグラフィックス関数として提供するのではなく、記録マクロを利用してその都度必要な部分を抜き出すというアプローチなので、誰でもが手軽に利用できるという形にはなっていない。<sup>2)</sup>

そこで、Excel VBA のモジュールとしてグラフィックス描画サブプロシージャを新たに定義し

ライブラリ化することにより、初学者でも手軽に利用でき、実用的なプログラムの作成が可能なプログラミング環境を提供することを試みた。

### 3. グラフィックス・ライブラリの概要

#### 3.1 グラフィックス・ライブラリの作成

Excel のシート上に直線や矩形及び楕円などの図形を様々な条件で描き、その際に得られた記録マクロの内容を詳細に解析した。その結果を元に、グラフィックス描画命令をプロシージャ

として定義した。ラインスタイルや色の指定などに関する Visual Basic の定数については、VBA のヘルプの記述を参考にした。

座標や主要なパラメータ変数は宣言部で定義し、大域変数としてプロシージャ間での値の引渡しを行うこととした。(表 1)

プロシージャの引数の並びは NEC PC-9801 の N88-BASIC に準拠した形式とし、ワールド座標でのグラフィックス描画命令を標準モジュール内のプロシージャとして定義しライブラリ化した。(表 2)<sup>3)</sup>

表 1 ワールド 座標グラフィックスのための大域変数

Public gWindowXOrigin As Double, gWindowYOrigin As Double	ワールド座標上の原点
Public gFactorX As Double, gFactorY As Double	スクリーン座標との換算比率
Public gCurrentX As Double, gCurrentY As Double	ワールド座標上の現在位置
Public gScreenXLeft As Single, gScreenYTop As Single	Viewportの左上座標
Public gScreenXRight As Single, gScreenYBottom As Single	Viewportの右下座標
Public gScreenX As Single, gScreenY As Single	スクリーン上の現在位置
Public gLineWidth	線の幅
Public gLineStyle	線の幅
Public gDashStyle	線の種類
Public gLineColor	線の色
Public gAreaColor	塗りつぶしの色

表 2 ワールド 座標グラフィックスプロシージャ

```

Sub InitializeGraphics()
Sub SetViewPort(ViewLeft, ViewTop, ViewRight, ViewBottom)
Sub SetGraphicsWindow(WindowLeft, WindowTop, WindowRight, WindowBottom)
Sub DrawLine(x1, y1, x2, y2, Optional cLineRGB)
Sub DrawRectangle(x1, y1, x2, y2, Optional cLineRGB)
Sub DrawRectangleFill(x1, y1, x2, y2, Optional cLineRGB, Optional cAreaRGB)
Sub DrawOval(x, y, rx, Optional ry, Optional cLineRGB)
Sub DrawOvalFill(x, y, rx, Optional ry, Optional cLineRGB, Optional cAreaRGB)
Sub PointSet(x, y, Optional cLineRGB)
Sub Move(x, y)
Sub DrawPolyLine(x, y, n)
Sub SetDashStyle(ds)
Sub SetLineStyle(ls)
Sub SetLineWidth(lw)
Sub gClear()

```

### 3.2 グラフィックス・ライブラリの動作環境

Windows 95, 98, Me, 2000, XP

Excel 97, 2000, XP, 2003

Mac OS, Mac OS X

Excel 98, X, 2004

簡単なグラフィックス描画プログラムをいくつか作成し、OSの違いやExcelのバージョンによらず、いずれも適切に動作することを確認した。<sup>4)</sup>

### 3.3 グラフィックス・ライブラリの利用法

Excel VBAの標準モジュールのModule2にグラフィックス描画命令プロシーダを組み込んだファイルを配布している。<sup>5)</sup> 利用者がVisual Basic Editorで、標準モジュールのModule1などにプログラムを記述して使用する。図1に、辺の長さが1の正多角形を描くプログラムの例と実行結果を示す。

```
Sub PolygonTest()
    Dim x(20), y(20), i
    InitializeGraphics
    SetViewPort 10, 10, 479, 479
    SetGraphicsWindow -2, 3, 3, -2
    For i = 3 To 8
        RegularPolygon x, y, i
        DrawPolygon x, y, i
    Next i
End Sub
```

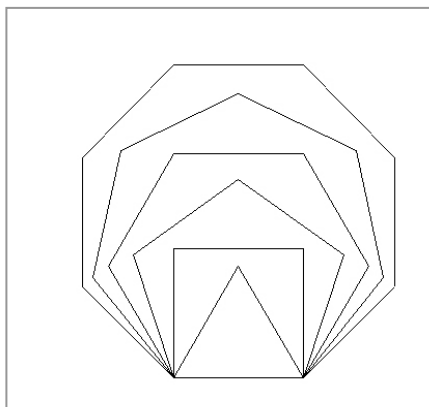


図1 図形描画プログラムの例

あらかじめ定義されたプロシーダを利用することにより、簡単なプログラムで容易に結果を得ることができる。

別なExcelファイルからグラフィックス描画命令を呼び出す場合には、ファイルごとに参照設定 (Visual Basic Editorで「ツール」-「参照設定」) を行う必要がある。その際、参照されるファイルは固定したフォルダ (ディレクトリ) に置く必要がある。

また、グラフィックス描画命令をアドインとして追加し、利用する方法もある。ライブラリを組み込んだExcelファイルをAddIn形式で保存し、AddInディレクトリにコピーした後、Excelのメニュー「ツール」-「アドイン」で追加する。<sup>6)</sup>

### 3.4 グラフィックスの設定と利用上の注意点

初期設定プロシーダ InitializeGraphicsの内容を図2に示す。

```
Sub InitializeGraphics()
    Call SetViewPort(0, 0, 639, 479)
    Call SetGraphicsWindow(0, 0, 639, 479)
    gLineWidth = 1 ' 線の太さ
    gLineStyle = msLineSingle ' 単線
    gDashStyle = msLineSolid ' 実線
    gLineColor = vbBlack ' 黒
    gAreaColor = gLineColor ' 線の色と同じ
End Sub
```

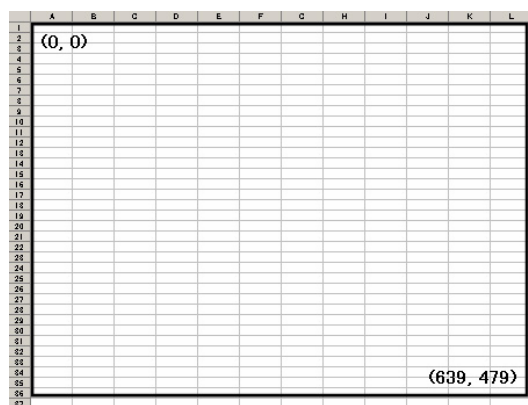


図2 初期設定プロシーダの内容

初期状態では、シートの左上が原点(0, 0)で、Viewport は仮に(0, 0)-(639, 479)としている。

右方向と下方向については、Viewport に関係なく描画が可能である。しかし、左方向と上方向については、Excel の機能によるもの以外のクリッピング処理は行っていないため、注意が必要である。

Viewport と Window を各自で定義することにより、描画領域と座標の設定を自由に変更することが可能である。縦横の比率を変え、ワールド座標系を定義した例を図3に示す。

```
Sub WindowTest()
  InitializeGraphics
  SetViewport 100, 10, 500, 410
  SetGraphicsWindow -30, 20, 30, -20

  DrawRectangle -30, -20, 30, 20
  DrawAxis 30, 20
  DrawLine -30, -20, 30, 20
End Sub
```

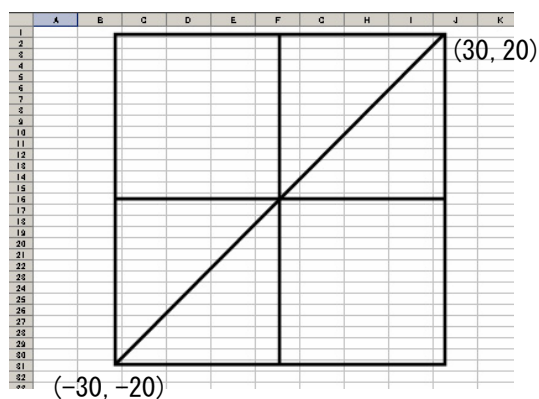


図3 ワールド座標系の設定例

パラメータを変えながら連続してリサージュ図形を描くプログラムの実行結果の一部を図4に示す。

線および塗りつぶしの色は、RGB 形式の16進定数、VB 定数または QBColor 関数により指定する。

色を指定しない場合は、直前に使用した色またはデフォルトの色が用いられる。

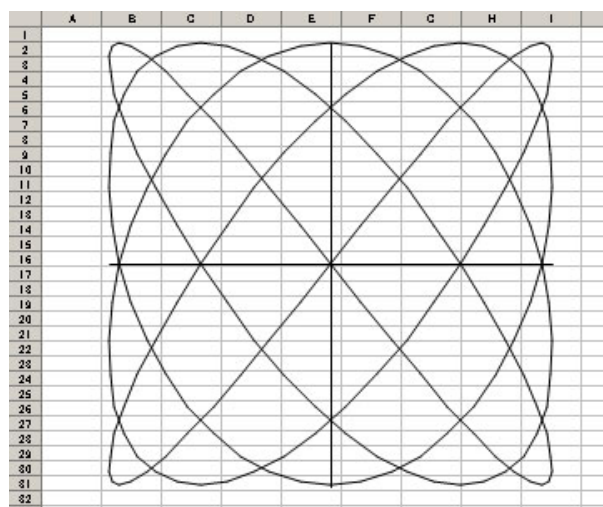


図4 リサージュ図形の描画例

線の太さやラインスタイルの変更は、プロシージャ呼び出しまたは大域変数への直接代入により行うことができる。

一方、Excel VBA でのグラフィックスの利用に関しては、以下に述べるような若干の制約があることが分かった。

- a) 描画オブジェクトの数が非常に多くなると、ある段階から処理速度が急激に低下する現象が発生する。

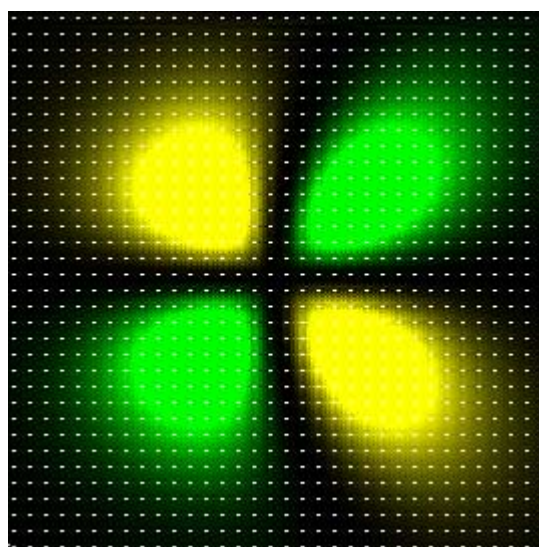


図5 3dxy 原子軌道関数の確率密度分布



例えば、図5に示すのは、水素原子の 3dxy 原子軌道関数の電子の確率密度分布を  $100 \times 100$  の格子点で計算し、フルカラーでプロットした結果である。

格子点の数をさらに増やし  $200 \times 200$  とした場合、実行に必要な時間は単純に4倍とはならず極めて長くなる。これは、Excel がアンドゥ機能を備えていることによるものと思われる。

- b) 描画命令はまとめて実行されるため、描画命令のみが連続するプログラムを実行した場合には、その間画面の再描画が行われず、画面には新たに何も表示されない。そこで、結果を確認した後、グラフィックスを消去し次の処理を行うことができるよう、マシンの時計を利用して指定時間だけ処理を停止するプロシージャ Delay を作成した。
- c) プログラム中で描画命令の占める割合が極めて大きい場合、実行途中での割り込みや中止などの制御が効かなくなることがある。

### 3.5 Turtle グラフィックス

次に、初心者向けのプログラミング教材での利用を目的として、LOGO 類似の形式で利用できる Turtle グラフィックス命令をプロシージャとして定義し、ライブラリ化した。(表3) <sup>7)</sup>

表3 Turtle グラフィックスサブルーチン

```
Sub InitializeTurtleGraphics()
Sub TGTurn(a)
Sub TGMove(length)
Sub TGMoveL(length, Optional cRGBLine)
Sub TGSetPoint(x0, y0)
Sub TGSetAngle(a0)
Sub TGRightTurn(a)
Sub TGLeftTurn(a)
Sub TGBackward(length, Optional cRGBLine)
```

LOGO のテキストに掲載されているプログラムのいくつかを移植し、同等な描画結果が得られることを確認した。図6, 7に Turtle グラフィ

ックスによる描画の例を示す。

この場合も、定義済みのプロシージャを利用することにより、簡単なプログラムを記述するだけで、目的に沿った結果を容易に得ることができる。

```
Sub TurtleTest2()
  Dim length, angle, d, c
  d = 0.005
  length = 1
  angle = 89
  InitializeGraphics
  SetViewPort 10, 10, 489, 489
  SetGraphicsWindow 0, 1.2, 1.2, 0
  c = QBColor(9)
  TGSetPoint 0.1, 0.1
  Do While length > d
    TGMoveL length, c
    TGTurn angle
    length = length - d
  Loop
End Sub
```

図6 Turtle グラフィックスプログラムの例

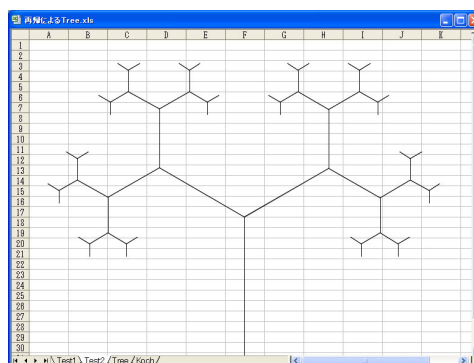
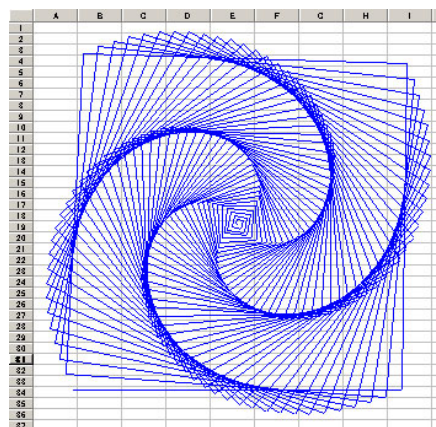


図7 Turtle グラフィックスによる描画の例

#### 4. おわりに

Excel VBA のプログラムは、機種、OS および Excel の版を問わず大多数のパソコンで動作可能なため、情報教育環境が比較的整備されている大学だけでなく、中学や高校での教育および一般社会人向けのプログラミング教育でも活用することができると思われる。

今後、Excel VBA のプログラミング文法とアルゴリズムの学習を兼ねた中級の教材やシミュレーションを中心とした問題解決型の課題を作成し、学習者の論理的思考力を育成するためのプログラミング学習環境として利用できるようになることを目指している。

#### 参考文献及び注

- 1a) 木原 寛、「Excel VBA を利用した論理的思考力を育成するためのプログラミング学習環境」、教育情報システム学会研究報告、Vol. 20, No. 6 (2006)
- 1b) 薮哲郎、<http://www.uopmu.ees.osakafu-u.ac.jp/~yabu/private/zuissou/program2.html>
- 2) 森口繁一、「Excel/Basic 基礎指南」 日本規格協会刊 (2000)
- 3) PSET は予約語となっているものの利用することができなかった。そのため、点を打つ命令 POINTSET は、小さな円を描くことで代替している。
- 4) Excel 2007 では、ハングアップする場合があります、実際上は利用することはできない。
- 5) <http://www3.u-toyama.ac.jp/kihara/vba/> に、Excel VBA 用グラフィックス・ライブラリとサンプルプログラムを掲載している。
- 6) Windows XP上でOffice 2003を使用した場合のAddInディレクトリは、” マイコンピュータ - ローカルディスク (C) - Documents and Settings - “User” - Application Data - Microsoft - AddIns” となる。
- 7) 木原 寛、「Excel VBA 環境によるグラフィックス機能を活用したプログラミング入門教材の作成」、平成 18 年度情報教育研究集会講演論文集、p. 189-190 (2006)