Ｅｘｃｅｌの活用例について

愛媛県立八幡浜高等学校　橋 本　　潔

１．はじめに

表計算ソフトの代表格となったＭｉｃｒｏｓｏｆｔ　Ｅｘｃｅｌ（以下Ｅｘｃｅｌ）は、現在、多くの先生方が成績処理や統計処理等、校務の中で活用しているアプリケーションである。現在使用されているＥｘｃｅｌは、非常に多くの機能を備えており、本来の表計算を主体とした処理以外にも使用できるアプリケーションに成長している。また、Ｏｆｆｉｃｅアプリケーションに実装されているＶｉｓｕａｌ　ＢＡＳＩＣ　ｆｏｒ　Ａｐｐｌｉｃａｔｉｏｎｓ（以下VBA）には、多数のコマンドや関数が用意されており、それを利用することでこれまでの作業の自動化や、多方面のアプリケーションを作成することが可能になる。そこで、Ｅｘｃｅｌの機能を拡張するＶＢＡを活用した例を考えてみたいと思い、この主題を設定した。

２．ＶＢＡについて

初心者向けのプログラム言語として開発されたＶｉｓｕａｌ　ＢＡＳＩＣをもとにＷｏｒｄやＥｘｃｅｌなどのＯｆｆｉｃｅアプリケーションの機能を拡張するためにカスタマイズされた言語である。

Ｅｘｃｅｌのセルをオブジェクトして扱えるようになっており、定型作業の自動化が簡単に行えるようになっている。

３．数学教材の作成

数学の授業の中でコンピュータの活用を考えた場合、最初に思い浮かぶのはＧｒａｐｅｓ等に代表される関数グラフソフトウェアである。そこで、今回Ｅｘｃｅｌを使用して関数グラフ教材を作成することとした。今年は、１年生を担当していることもあり、２次関数の分野で使用する教材を作成した。作例は、y=a(x-p)^2+qについて、各パラメータを変えることで、グラフが変化するようにしたものである。

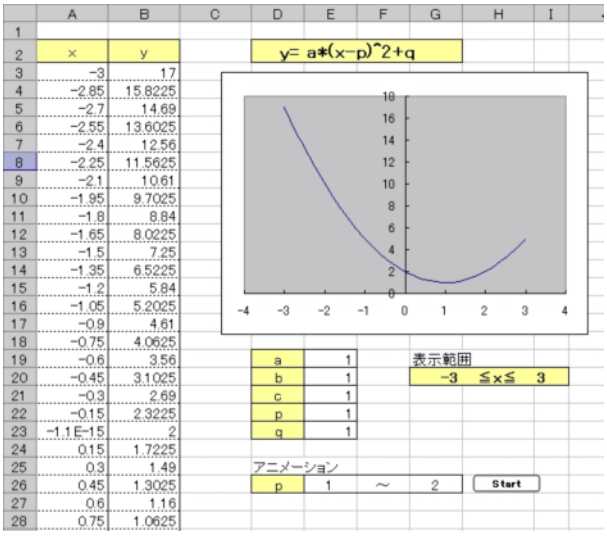

図１　作成した教材

⑴　グラフ描画について

Ｅｘｃｅｌには、直接、関数のグラフをかく機能はないため、今回は散布図を使用することとした。表示範囲を40分割し、41個の点を滑らかにつなぐことで、グラフを表現している。41個の点の座標はＡ３からＢ４３のセルで計算している。授業ではこのセルは非表示にして使用する方が画面はすっきり見える。

⑵　関数値の算出について

散布図で使用した41個の点のｙ座標を計算するためには、直接、Ｂ３～B43のセルに式を入力することもできるが、授業で活用することを考えれば、途中で式の形を変える場面も想定できる。そこで、あるセルに書かれた式を評価するために、つぎのような関数をＶＢＡで作成した。

```
Function eval(ByVal expr As String, dummy) As Double
    eval = Evaluate(expr)
End Function
```

リスト１　数式評価関数

この関数を用い、Ｂ３以下のセルには、「=eval(式,a&p&q&x)」と入力することで計算させている。２つ目の引数は、このeval関数内では用いていないが、これを指定しておくことでa,p,q,xの値が変更されたときに自動的にeval関数を使って再計算させることができる。

また、式中にあるa,b,c等のパラメータは、名前を定義しておくことで指定セルの値を使用して関数の値を計算することになる。今回定義した名前は次の表の通りである。授業利用の場面を想定し、一部作例で使用していない名前も定義している。Cという名前はＥｘｃｅｌで予約しているようで、_c(アンダースコアc)として定義している。

名前	定義
式	=$E$2
A	=$E$19
B	=$E$19
C	=$E$20
P	=$E$21
Q	=$E$22
X	=$B3

表１　作例で定義している名前

⑶　アニメーション効果について

今回は、グラフが変化する様子をじっくり観察できるようにアニメーション機能も付けた。図１右下に配置したＳｔａｒｔボタンを押すことでリスト２のStart_Anim関数が実行され、対象となるパラメータが変化することで、グ

ラフが変形していく。見ている時間も考慮しており、アニメーション自体は４秒程度で終わるようにスピードを調整している。

```
Sub Start_Anim()
    st = (Range("G26").Value - Range("E26").Value) /
20
    For i = Range("E26").Value To Range("G26").Value
+ st Step st
        Range(Range("D26").Value).Value = i
        WaitSec (0.2)
    Next
End Sub
```

リスト２　アニメーション処理関数

Start_Anim関数中のWait_Sec関数は、一定時間待つ処理をするものである。この部分が無いと最近のコンピュータの演算性能であれば、一瞬にして処理が終わり、グラフの変化を観察することができない。

```
Sub WaitSec(sngWaitSec As Single)
  Dim dblCurDateTime As Double
  dblCurDateTime = Now + sngWaitSec / 86400
  Do
  Loop Until Now >= dblCurDateTime
End Sub
```

リスト３　処理待ち関数

４．バイナリエディタの作成

校務の中で、バイナリデータを扱う機会があったため、Ｅｘｃｅｌでバイナリエディタを作成してみた。Ｅｘｃｅｌには、バイナリファイルの入出力機能が用意されているため、比較的簡単に作成することができた。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	0	53	45	49	4B	4F	20	51	54	35	20	56	30	31	2E	30	30
2	10	51	54	35	58	C2	19	30	00	50	00	02	00	D8	13	4D	15
3	20	C2	16	37	18	AC	19	B2	19	00	00	01	00	00	00	00	00
4	30	00	19	00	03	00	00	00	00	00	00	00	00	00	00	00	00
5	40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
6	50	01	3E	08	0F	11	01	3E	08	19	11	01	3E	08	23	11	01
7	60	3E	08	2D	11	01	3E	08	37	11	01	3E	09	2D	11	01	3E
8	70	09	37	11	01	3E	0A	2D	11	01	3E	0A	37	11	01	3E	0B
9	80	2D	11	01	3E	0B	37	11	01	3E	0C	2D	11	01	3E	0D	14
10	90	11	01	3E	0D	19	11	01	3E	0D	23	11	01	3E	0D	28	11
11	A0	01	3E	0E	1E	11	01	3E	0E	28	11	01	3E	0F	1E	11	01
12	B0	3E	0F	28	11	01	3E	10	1E	11	01	3E	12	00	11	01	3E
13	C0	12	0F	11	01	3E	12	37	11	01	3E	13	00	11	01	3E	13

図２　バイナリエディタ

数学とは直接関連のない題材であるため、プログラム全体の説明は割愛するが、バイナリデータの出力部分は、リスト４のようなものである。

```
Sub データ出力()
    Dim oFile As String
    Dim buffer(412 * 16 + 2) As Byte
    Dim oFn As Long

    oFile = "Output.BIN"
    oFn = FreeFile

    Set sref = Worksheets("DATA")

    For r = 1 To 412
        For c = 2 To 17
            buffer((r - 1) * 16 + c - 2) = Val("&H"
& sref.Cells(r, c).Value)
        Next
    Next
    buffer(412 * 16) = 0
    buffer(412 * 16 + 1) = 0

    Open oFile For Binary As #oFn
        For i = 0 To 412 * 16 + 1
            Put #oFn, , buffer(i)
        Next
    Close #oFn
End Sub
```

リスト４　バイナリデータ出力関数

リスト後半のように、出力用のファイルをバイナリモードで開いておき、buffer配列に格納されたデータを１バイトずつPut関数で出力することで、簡単にバイナリファイルを作成することができる。

５．まとめ

今回はＥｘｃｅｌのＶＢＡを使用して、２本のプログラムを作ってみた。授業で使える程度の教材ならば、Ｅｘｃｅｌで対応できない部分にＶＢＡを使用することで簡単に作成できた。また、今回の教材に盛り込まなかったグラフの一部を太線にする処理も、使い慣れたＥｘｃｅｌならば少しの工夫で実現することができる。さらに、バイナリエディタといった表計算とは違った分野での使い方も容易に行えることが確認でき、ＶＢＡにより機能を追加することでＥｘｃｅｌの活用場面は飛躍的に広がるものと思われる。

今後も多くの機能を持つＶＢＡの活用分野を探し、ほとんどのパソコンにインストールされているＥｘｃｅｌの有効な利用方法を考えていきたいと思う。