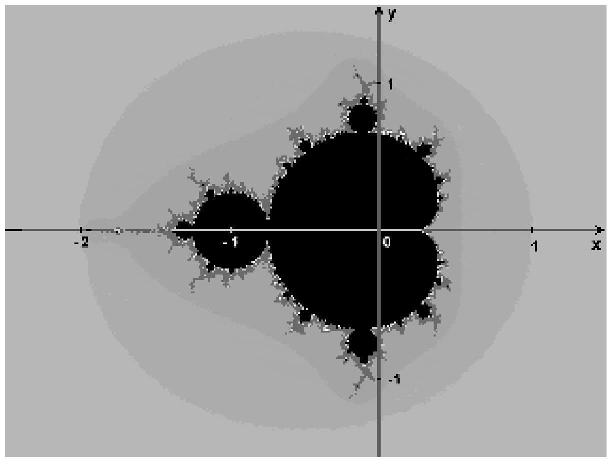
Using GW-BASIC for Drawing Mandelbrot Sets

Mr. LEUNG Chi Kit

The Hong Kong Taoist Association Ching Chung Secondary School

For a given complex number c_0, define the sequence of complex numbers { c_n }by $c_{n+1} = c_n^2 + c_0$ for $n = 0 , 1 , 2 , \ldots$. If the sequence is bounded, that is, we can find a real number M such that for every n, $| c_n | < M$, then c_0 is said to belong to the Mandelbrot set.



We can use the definition above to write a GW-BASIC program（Sorry！I always like to use the simplest computer language and I think it is not difficult to translate a GW-BASIC program to other computer language）as follow：

```
10    LEFT = 150 : TOP = 380 : W = 360 : M = .833
20    R = 2.64 : S = 2 * R / W
30    RECEN = 0 : IMCEN = 0
40    SCREEN 9 : CLS
50    FOR Y = 0 TO W
60        FOR X = 0 TO W
70            REC = S * (X – W / 2) + RECEN : IMC = S * (Y – W / 2) + IMCEN
80            RE = REC : IM = IMC
90            RE2 = RE * RE : IM2 = IM * IM : J = 0
100           WHILE RE2 + IM2 <= 256 AND J < 15
110               IM = 2 * RE * IM + IMC
120               RE = RE2 – IM2 + REC
130               RE2 = RE * RE : IM2 = IM * IM : J = J + 1
140           WEND
150           IF J < 3 THEN GOTO 220
160           IF J >= 3 AND J < 6 THEN COLOR 14 : REM YELLOW
```

```
170             IF J >= 6 AND J < 9 THEN COLOR 1 : REM BLUE
180             IF J >= 9 AND J < 12 THEN COLOR 2 : REM GREEN
190             IF J >= 12 AND J < 15 THEN COLOR 15 : REM WHITE
200             IF J >= 15 THEN COLOR 12 : REM RED
210             PSET (X + LEFT, (TOP – Y) * M)
220         NEXT X
230     NEXT Y
240     COLOR 15 : REM WHITE
250     LINE (LEFT, (TOP – W / 2) * M) – (W + LEFT, (TOP – W / 2) * M)
260     LINE (W / 2 + LEFT, (TOP – W) * M) – (W / 2 + LEFT, TOP * M)
270     END
```

The following explains the program：

W sets the size of the picture to be drawn on the computer screen. Initially W is set to 360（see line 10). This means we plan to draw the Mandelbrot set in a 360×360 square in the computer screen (see lines 50 and 60).

LEFT is the leftmost position of the picture on the screen, TOP is the lowest position of the picture (see lines 210, 250 and 260). Caution: in GW-BASIC, the coordinates of the computer screen go from top to bottom unlike our usual convention of going from bottom to top. So we have to use “TOP – Y” to convert the usual coordinate system to the computer screen coordinate system.

Since a pixel on the computer screen is not a square, so the horizontal and vertical sizes are not the same, hence we introduce M（= $\frac{5}{6}$）to adjust the length-to-width ratio（see lines 10, 210, 250 and 260).

Note W is only the size on the screen and not the actual coordinates of the complex numbers in the Mandelbrot set. So W needs to be transformed. R is the actual value（see line 20). That is, for the range of the picture, the real axis goes from –R to R and the imaginary axis also goes from –R to R. S computes the ratio of W and R and is used in later computations（see lines 20 and 70).

RECEN and IMCEN are used to locate the position of the center. The center is initialized to (0 , 0)（see line 30). By changing the value of R, RECEN or IMCEN, we can move or dilate the Mandelbrot set.

Line 40 chooses the format of the picture and erase the old screen.

Lines 50 and 60 of the program set the ranges of X and Y. Then line 70 computes the real and imaginary parts of the corresponding c_0.

Observe that if $c_0 = a_0 + b_0 i$，$c_n = a_n + b_n I$, then

$$\begin{aligned} c_{n+1} &= c_n^2 + c_0 \\ &= (a_n + b_n i)^2 + (a_0 + b_0 i) \\ &= a_n^2 - b_n^2 + 2a_nb_n i + a_0 + b_0 i \\ &= (a_n^2 - b_n^2 + a_0) + (2a_nb_n + b_0)i. \end{aligned}$$

So the real part of c_{n+1} is $a_n^2 - b_n^2 + a_0$ and the imaginary part is $2a_nb_n + b_0$.

Converting these computations to codes yield lines 110 and 120. REC and IMC are the real and imaginary parts of c_0 respectively. RE and IM are the real and imaginary parts of c_n respectively. RE2 and IM2 are the squares of the real and imaginary parts of c_n respectively.

J is a counter for running the loops in lines 100 and 140. Line 100 also computes the square of the modulus of c_n. If the square of the modulus is greater than 256 or the loop has been executed 15 times, then we terminate the loop. Consequently, the larger the value of J is, the closer the sequence will tend to "converge". That is, after many computations, the modulus of c_n still does not get big. Lines 150 and 200use colors to classify the rate of convergences. Red indicates the complex numbers with fastest converegence, then comes white, green, blue and yellow. The region with the fastest divergence is indicated in black. Line 210 chooses the color for the point.

After drawing the Mandelbrot set, we draw the horizontal and vertical axes in white（see lines 240 and 260) for reference. This concludes the program.

The running time for the program depends on the speed of the computer. For the present computers, the whole program can finish in about a minute.

<u>Reference</u>

Heinz-Otto Peitgen, Hartmut Jürgens and Dietmar Saupe (1992) *Fractals for the Classroom Part Two: Introduction to Fractals and Chaos.* NCTM, Springer-Verlag.