Clever fast Mandelbrot - please explain!

4th September, 2020, 12:12 (This post was last modified: 4th September, 2020 12:14 by EdS2.) **Post: #1**

EdS2
Senior Member

Posts: 525
Joined: Apr 2014

Clever fast Mandelbrot - please explain!

I refer to Valentin Albillo's remarkable exposition VA040b, calculating the area of the Mandelbrot set, as linked here.

The specific optimisation I'd like to understand is the shortcut which detects the two otherwise very expensive central areas: the main cardioid and the largest circular bud, aka the main disc. Normally these are rendered in black, and take the maximum number of iterations for each point evaluated.

Valentin notes:

> **Quote:**
> if we can quickly check whether a given z belongs or not to the main cardioid or the main disk we'll save lots of running time and as it happens, indeed we actually can, using just a few steps for the RPN version or just 2 lines of code for the BASIC version.

All this is clear, but it's the two lines of code, or the few steps, which have me baffled, as to how exactly they determine membership in these two areas.

Any explanations welcome!

Ref:
HP Article VA040b - Boldly Going - Mandelbrot Set Area (HP-71B)

> **Quote:**
> Line 3: checking whether the point belongs to the main cardioid (thus, to M).
> Line 4: checking whether the point belongs to the main disk (thus, to M)

See also
HP Article VA040a - Boldly Going - Mandelbrot Set Area (HP42S)

> **Quote:**
> Steps 45-53: checking whether the point belongs to the main cardioid (thus, to M). { 9 steps }
> Steps 54-59: checking whether the point belongs to the main disk (thus, to M). { 6 steps }

4th September, 2020, 22:10 **Post: #2**

Valentin Albillo
Senior Member

Posts: 970
Joined: Feb 2015
Warning Level: 0%

RE: Clever fast Mandelbrot - please explain!

Hi, **EdS2**:

> **EdS2 Wrote:** (4th September, 2020 12:12)
> I refer to Valentin Albillo's remarkable exposition VA040b, calculating the area of the Mandelbrot set [...].

Thank you very much for your continued interest in my productions, much appreciated (and keep doing it, it will encourage me to create more ... 🙂)

> **Quote:**
>
> The specific optimisation I'd like to understand is the shortcut which detects the two otherwise **very expensive** central areas: the **main cardioid** and the largest circular bud, aka the **main disc**. Normally these are rendered in black, and **take the maximum number of iterations for each point evaluated**.

Correct, they're incredibly expensive to evaluate, most specially if the max. number of iterations is huge.

> **Quote:**
>
> All this is clear, but it's the two lines of code, or the few steps, which have me baffled, as to how exactly they determine membership in these two areas.
> **Any explanations welcome!**

But f course !. I'll explain it for the **HP42S** version, which is based on the **HP-71** version (which I wrote *first*, then adapted for the *HP42S*):

1) Steps 45-53: checking whether the point belongs to the main cardioid (thus, to M). { 9 steps }

First of all, mathematically we have, using the **main cardioid**'s equation:

All points ***C*** *within the main cardiod* $\boldsymbol{C < 1/4*(2e^{ti} - e^{2ti})}$ *are in* ***M***,

where **C** is a complex number representing a point, **i** is the imaginary unit, **t** is the argument *arg(**C**)*, and *0 <=* ***t*** *<= 2Pi*.

Now, we have (in **HP-71B** *BASIC* code which will be momentarily converted to **HP42S** *RPN*):

$\mathbf{2e^{ti}}$ = 2*EXP(0,ARG(C)) = **2*SGN(C**)
$\mathbf{e^{2ti}}$ = EXP(0,2*ARG(C)) = EXP(0,ARG(C))^2 = **SGN(C)^2**

so the expression becomes:

C < 1/4*(2*SGN(C)-SGN(C)^2)

To speed the computation we use an auxiliary variable **Z**=SGN(C) and the expression now becomes:

C < 1/4*(2*Z-Z^2) -> C < 1/4*(Z*(2-Z)) -> **ABS(C) < 1/4*ABS(Z*(2-Z))**

which we convert to RPN like this:

```
    ...           begin generation of a random point C within the enclosing box
 37 RAN
 38 RCLx 06
 39 RCL- 07
 40 RAN
 41 RCLx 08
 42 COMPLEX       C
 43 ENTER         C                 C
 44 ENTER         C                 C             C

                  now check if C belongs to the main cardioid

 45 SIGN          Z                 C             C
 46 RCL- 07       Z-2               C             C
 47 RCLx ST L     Z*(Z-2)           C             C
 48 ABS           ABS(Z*(Z-2))      C             C
 49 RCLx 09       1/4*ABS(Z*(Z-2))  C             C
 50 X<>Y          C                 1/4*ABS...    C
 51 ABS           ABS(C)            1/4*ABS...    C
 52 X<Y?          ABS(C)    <?      1/4*ABS...    C
 53 GTO 04        yes, it does belong to the main cardioid and thus to M, go increment the
tally
```

2) Steps 54-59: checking whether the point belongs to the main disk (thus, to M). { 6 steps }

M also contains a **main disk** with radius **1/4** and centered at (-1, 0):

point **C** = (x,y)
the equation of the circle is: $\mathbf{(x+1)^2+y^2 = (1/4)^2}$, so if $(x+1)^2+y^2 < (1/4)^2$, then the point belongs to the circle, i.e. is in **M**

Taking the square root, we have: $\text{sqrt}((x+1)^2+y^2) < 1/4$, which in **HP-71B** code is : if **ABS(C+1)<1/4** then the point belongs to the circle, i.e. in **M**

```
          ...            ABS(C)            any          C      {stack contents at this point, as seen
above}
      54  SIGN           1                 any          C
      55  RCL+ ST Z      C+1
      56  ABS            ABS(C+1)
      57  RCL 09         1/4               ABS(C+1)
      58  X>Y?           1/4       >?      ABS(C+1)
      59  GTO 04         yes, it belongs to the main disk and thus to M, go increment the tally.
```

Well, I hope this leaves the matter crystal-clear for you but if there's still something you want explained, just ask and I'd be glad to obligue.

Again, **thanks for your interest** and best regards. Have a nice weekend and take care.
V.

PM WWW FIND EDIT QUOTE REPORT

7th September, 2020, 10:19 **Post: #3**

EdS2
Senior Member

Posts: 525
Joined: Apr 2014

RE: Clever fast Mandelbrot - please explain!

Many thanks Valentin! (I may need a little time to digest and experiment. My brain is not as agile as it was.)

EMAIL PM FIND QUOTE REPORT

7th September, 2020, 21:55 **Post: #4**

David Hayden
Senior Member

Posts: 424
Joined: Dec 2013

RE: Clever fast Mandelbrot - please explain!

I wrote a very fast Mandlebrot plotter in the late 80's for the PC. One very effective optimization was to consider a rectangular area. Divide it into 4 sub-rectangles. Now compute the value at the 9 points of intersection (top left, top center, top right, middle left, middle center, middle right, bottom left, bottom center, bottom right).

If all 9 points generate the same value then assume that the entire rectangle is that value. Otherwise recurse on the sub-rectangles. While this optimization isn't always correct, in practice, it provides outstanding results.

Some day, I'd like to port that code to the Prime.

As I recall, I also realized that the Mandlebrot set and the Julia set are the same equation, but using two different planes of a 4-dimensional space. My program lets you plot the equation on different planes, resulting in some really cool effects.

EMAIL PM FIND QUOTE REPORT