Report : Fractal – Alexandre Temperville

The **Mandelbrot set** is a fractal define as the set of points *c* of the complex plane such that the sequence (z_n) defined by recurrence by :

$$z_{n+1} = z_n^2 + c \quad \text{with} \quad z_0 = 0$$

does not go towards $+\infty$.

For $z_n = x_n - i*y_n$ and c = a + i*b, it is equivalent to study :

$$x_{n+1} = x_n^2 - y_n^2 + a \quad \text{and} \quad y_{n+1} = 2x_n y_n + b \quad \text{with} \quad x_0 = y_0 = 0,$$

so then the condition becomes : the absolute values of x_n and y_n should not go towards $+\infty$.

Parameters of mandel.c

They are in the procedure '*analyser*'. If we do not put any instructions after the name of the executable of file *mandel.c* (I call it *mandel*) and run it, then the parameters *mandel* uses are by default :

```
nb_iter = 100;
x_min = X_MIN;
x_max = X_MAX;
y_min = Y_MIN;
y_max = Y_MAX;
larg = 1024;
haut = 768 ;
```

1) **Command -b**

If the enter the command : **./mandel -b -2 2 -3 5** then we modify *x_min*, *x_max*, *y_min* and *y_max*. The final image is the Mandelbrot fractal in the space domain [-2;2]x[-3;5].

2) **Command -d**

If we enter the command : **./mandel -d 2048 500** then we modify the resolution of the image, and we see the image as an affine map of the image we could have without this command.

3) **Command -n**

If we enter the command : **./mandel -n 200** then we modify the number of iterations (200) to verify if (x,y) (represented by z_{200} here) belongs to the disk of center (0,0) and of radius 2.

Procedures

1) **'analyser'**

We explain it in the previous part of this report.

2) **'initialiser'**

Allocate memory for the structure im which there are an array containing caracters allowing to define the colour of pixels, the number of lines and columns these pixels represent.

3) **'sauvegarder'**

This procedure saves the caracters defining pixels in a picture. A colour is defined in a computer by 3 caracters (red, bleu, green), so here we put 3 times the same caractere at one pixel of to have shades of black.

4) **'calculer'**

In each point *c* of the picture, we compute if this c = a + i*b can define a convergent sequence (z_n) until the *nb_iter* loops.

A condition of divergence is given by $|z_n|>2$ represented by the condition x*x+y*y >4 (it is a theorem). When we have this condition then, we stop and we did a certain number of loops which define a number between 0 and 255 (colours are defined by caracters between 0 and 255).

More we are next to 255, more the pixel will be white. More we are next to 0, more the pixel will be black.

Parallel version of the Mandelbrot fractal

To implement a parallel version of the Mandelbrot fractal, I use MPI and a shell script.

1) **Program in MPI**

I propose to cut the picture we want in horizontal pictures with the height divided by the number of processors we want to use, then we gather them in a final picture called *final.png*. The computations are divided between the processors as each of them build its picture.

To do that, first of all we create the name of the picture of each processor, and order it in the decreasing order (this will be very useful for the script we explain later). I choose to name this files with 2 figures to be sure they are in the correct order (that is very important when running the shell script).

Next, we can analyse again if we put some parameters or not after the name of the executable **./mandel** with the procedure *'analyser'*.

Then, we need to modify the resolution parameter *haut* (*haut = haut/P* with *P* processors) and the values of *y_min* and *y_max* of each processors, I call them *y_min_proc* and *y_max_proc*.

At last, we create the horizontal picture of each processor called *mandel**.ppm* with * being a figure. We can see 100 horizontal pictures in the folder 'découpage' when I run the program for 100 processors.

2) **Script shell**

To run this script, we need first of all to make a 'mpdboot' in order to use MPI. Then we can run the script with parameters.

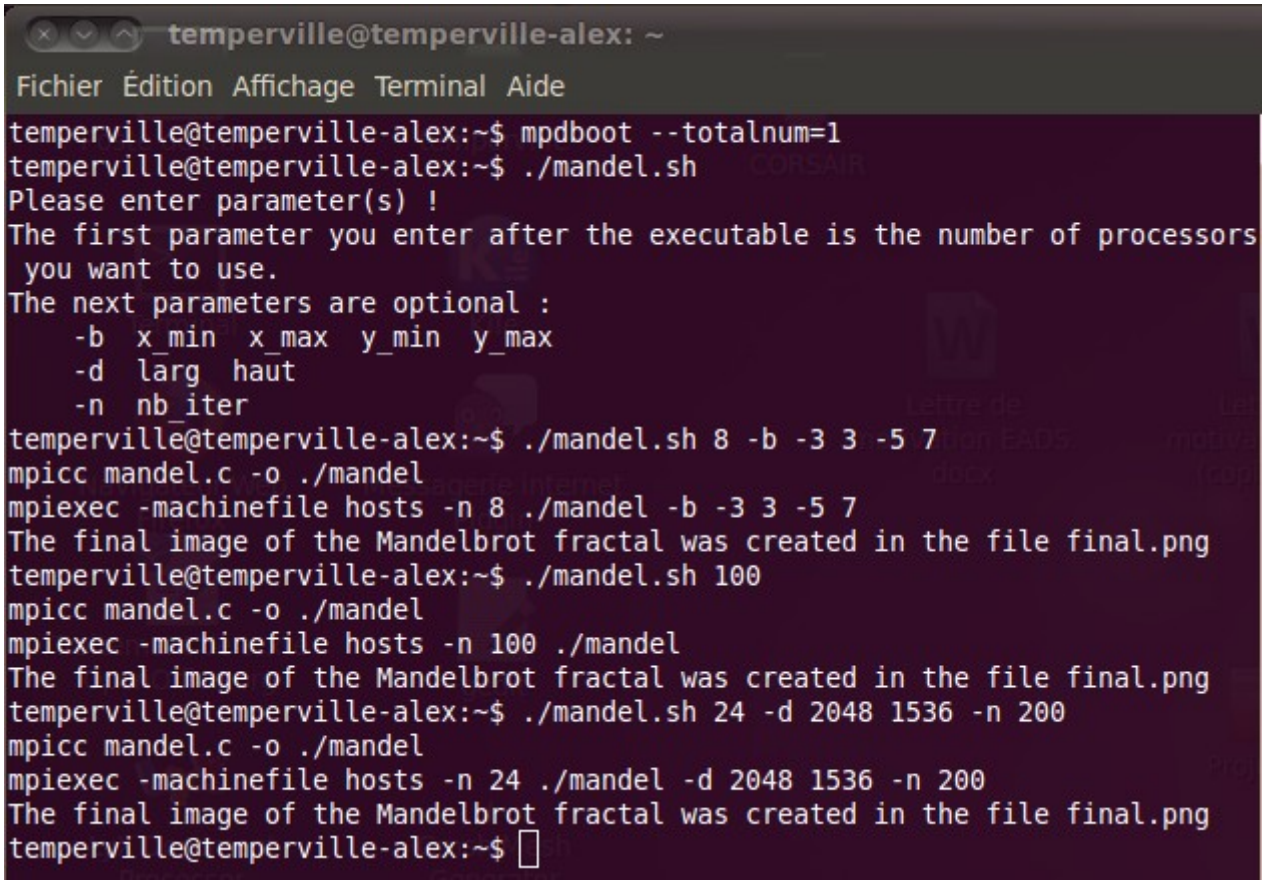
If we do not enter at least one parameter (the number of processors), we have an error message giving the parameters we can enter.

If we enter one parameter, then we do the instructions in the *else* :

- we compile the program in an executable called *mandel*.
- we execute mandel with optional parameters.
- we add the pictures in order of their name in a picture called *final.png*.
- we remove all the horizontal pictures created.

3) Running in the terminal

In my PC, I just use one computer, in the room 106, we could put a higher number of machines in *totalnum*. Here we can see what I obtain in the terminal when I run the script *mandel.sh*.



Concerning the folder

The folder Fractal contains this report, the program *mandel.c*, the script *mandel.sh*, a folder containing 100 horizontal pictures of the Mandelbrot fractal, the picture *final.png* representing the set of Mandelbrot (found thanks to the last command in the terminal) and the files *hosts* and *mpd.hosts* I use in my computer to run this program with MPI.

Bibliographie

http://www.imagemagick.org/script/convert.php