FRACTALS, FUNCTIONS, and FEEDBACK SYSTEMS

Summary: This article describes an investigation into fractals, the methods of creating them, as well as some mathematical considerations about their nature. This investigation led to a further study of the complex functions used in creating fractals. The results and observations employed *relaxation factors* in evaluating the numerical properties of the calculations needed to generate solutions of the equations. These equations are used in the generation of fractals. Some of the processes used in finding the roots of the investigated functions are also discussed. A comparison with feedback systems is also introduced.

1. The Mandelbrot Set

The notion "fractal" was coined by Mandelbrot and is derived from the Latin word *fractus*. The corresponding Latin verb *frangere* means "to break": to create irregular fragments. Many well known shapes or curves exhibit the properties of fractals. This is especially true since Mandelbrot reasons that these shapes or curves are of a different dimension than the dimensions of the Euclidian space and may assume a fractional value. The dimensions of fractals can be expressed as the division of the logarithms of two integer numbers and thus fractals can indeed be viewed as fractional dimensions. This can be demonstrated in lines, where the original line is broken up and lines of a different dimension are inserted in the gap that resulted from the braking up of the line. A similar construct can also be made for curves that are broken up. More about that at the end of this chapter on the Mandelbrot set.
Fractals can also be viewed as geometrical objects that display similar appearances and properties under a change of scale (for example magnification). The concept is helpful in disciplines where order can be perceived as disorder. In the case of a river and its tributaries, it appears that every tributary has its own tributaries and as a result a smaller portion of the river has the same structure and organization as the whole river. The same is true for the branching of trees and roots, the blood vessels, underground caves and grotto's's, the nerves and bronchioles in the human body. Many studies in fractals are devoted to the patterns of sea shores and ways to calculate its exact distances. The rise and fall of economic indices also exhibits self-similar structures, as well as the behaviour of weather patterns.
I will first discuss the method of creating the Mandelbrot set and how to create this set. The Mandelbrot set is the locus of points, C, for which the series:

$$Z_{n+1} = Z_n^2 + C, \quad Z_0 = (0,0) \tag{1}$$

is bounded by a circle of radius two, centered at the origin whilst Z and C are complex numbers. A complex number is a number consisting of a real part and an imaginary part. Z is represented by X + i*Y where i = $\sqrt{-1}$.
One of the fascinating things about the Mandelbrot set is the seeming contradiction in it. It

is said to be the most complex object in mathematics, perhaps the most complex object ever seen. At the same time, it is generated by an almost absurdly simple formula. In the literature the function that is used in creating fractals is referred to as *strange attractor*, although Mandelbrot himself states that there is nothing *strange* about it and the notion *attractor* describes its behaviour sufficiently well. What now follows is the way this attractor is used to generate fractals.

Multiply Z_n by itself. Add C. The answer is the new value for Z: Z_{n+1}. Repeat this until the absolute value of Z_n is greater than 2, or until a counter (the specified number of these repetitions) expires. If the absolute value of Z ever exceeds 2, then it will very quickly head off towards infinity; that means that the point is not in the Mandelbrot set. The Mandelbrot set is thus the definition of all points C for which the values of Z_{n+1} do not diverge. If the absolute value of Z_n doesn't exceed 2 after a specified number of iterations, then we give up and assume that the initial point is in the Mandelbrot set. The iterations also stop when the difference between the absolute values of Z_{n+1} and Z_n becomes smaller than a given accuracy. The choice of these values (number of iterations and the value of the accuracy) will be discussed later. There are two possibilities for comparing those values. One can either compare the absolute values or compare the absolute values of the real and imaginary parts of both complex numbers. The absolute value of a complex number is the distance of the complex number X + i*Y to the origin of the XY-coordinate system and is expressed as $\sqrt{\mathbf{X}^2 + \mathbf{Y}^2}$. See figure 1. If the difference between Z_{n+1} and Z_n becomes indeed smaller than the specified accuracy, we assume both values to be the same and the equation (1) can be written as the equation

$$\mathbf{Z^2 - Z + C = 0} \qquad (2)$$

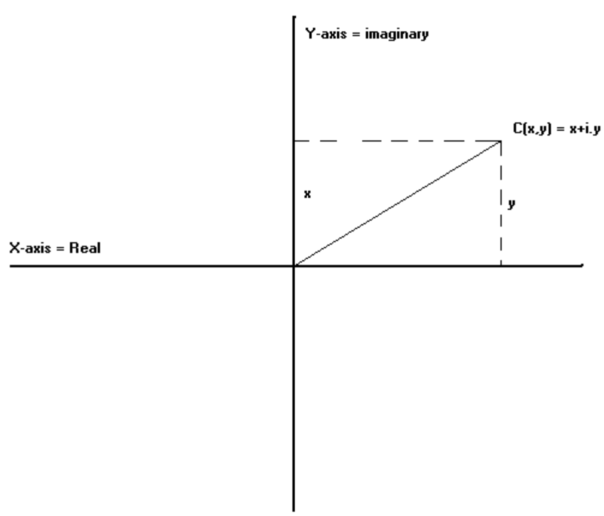

Figure 1: Coordinate system for complex numbers

In this method of successive substitution we use the equation $Z_{n+1} = Z_n^2 + C$ to find solutions for the equation $Z^2 - Z + C = 0$. The chance that we are able to solve this equation by means of the method of successive substitution is small and in most cases the method will quickly diverge. However, in certain areas of the complex plane (the coordinate system defined by the X- and Y-axis that represent the real and imaginary numbers) solutions will be found or will turn out to be undefined. The latter situation happens when the found values of Z start to oscillate and create a repetitive pattern or when the found solutions show a chaotic behaviour without diverging outside the specified boundary of the absolute value of Z. A repetitive pattern may consist of series of 2 to 12 values and are often referred to as the periods of the oscillating cycle. This does not mean though that solutions to the equation $Z^2 - Z + C = 0$ don't exist. For the moment we limit our observations to the method of successive substitution.

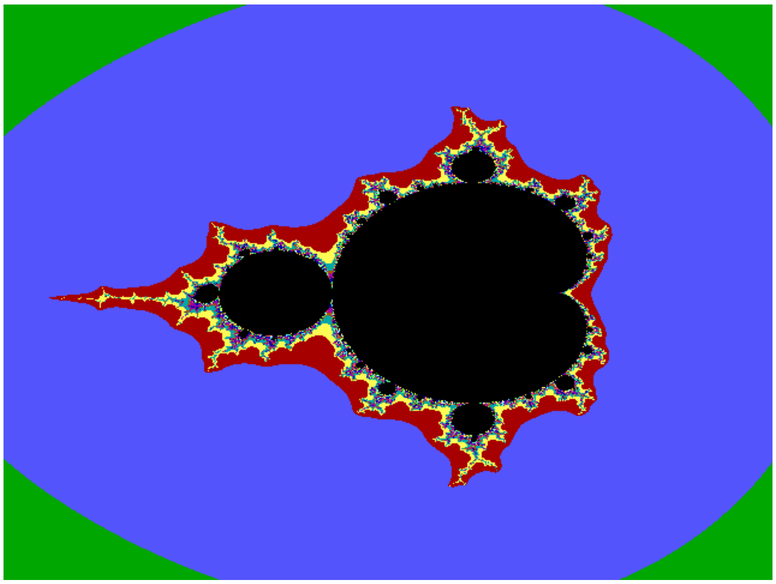

Figure 2: Mandelbrot set (black pixels)

The pretty pictures of a fractal can now be constructed in the following way. Let C be the set of all points that can be represented by the pixels of a colour display screen. Initialize Z to be a certain value (the seed value) of i.e. 0 + i*0. Assign this seed value to Z_n and determine Z_{n+1}. Assign the newly calculated value to Z_n and compare its new value with the previous one. If this process diverges, say after 200 iterations, assign a colour to the pixel represented by the x- and y-value (the real and imaginary value) of C. Assign a different colour too the pixel of C if the process diverges after 180 iterations, another colour after 160 etc. until a series of iteration bands have been assigned to the diverging C-values. In case the difference between the old and new value of Z_n is smaller than the specified accuracy, a solution is found and the pixel remains black. This is also the case if after the mentioned 200 iterations the process has not diverged yet. These pixels can also be coloured black or a specific colour can be assigned to these pixels.

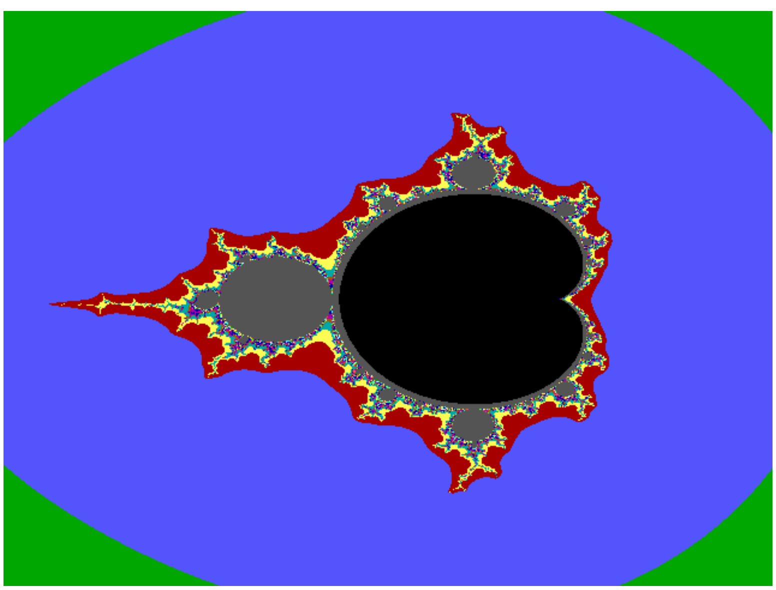

Figure 3: The black and gray pixels are part of Mandelbrot set

The points of the Mandelbrot set are thus typically colored black. The black, barnacle covered pear is the Mandelbrot set proper. All the bands of colour outside of it are simply curious artifacts that help to expose the detail of the Mandelbrot set itself. The result of this process is shown in figure 2. The coordinates of the left-bottom corner of the picture are (-2.2 ,-1.6), those of the right-top corner (1.1 ,1.6). The X-coordinates of C increase from left to right with a value equal to the difference between the coordinates of the right and left border divided by 640. In a similar way the Y-coordinates increase from bottom to top with a value that is equal to the top minus the bottom coordinate values divided by 480; 640 and 480 being the amount of screen pixels that are addressed. The limiting dimension of the large black cusp is set on the value 2. This defines the boundary of the Mandelbrot set. A solution to equation (1) is found when the difference between the real and imaginary parts of Z_{n+1} and Z_n is less than a set accuracy (here 0.0001). The iteration process stops after 200 iterations. Changing the values of the coordinates that define the boundary of the picture, the value that defines the boundary of the large cusp, the accuracy, or the maximum amount of iterations may change the resulting fractal. Figure 3 exhibits the same fractal as the one shown in figure 2. Only here the value of C that does not generate a solution after 200 iterations of formula (1), but are not diverging either, are coloured gray. The red colour represents the area where equation (1) diverges after less than 10 iterations, blue after less than 16; high blue after less than 22 etc.

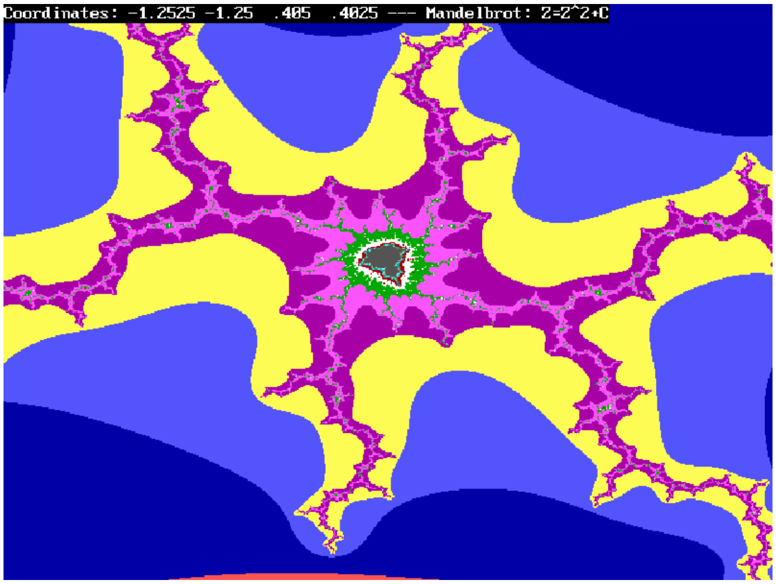

Figure 4: Magnification of part of the fractal in picture 2

The repetitive pattern of the fractal is shown in figure 4. This fractal is the result of applying equation (1) to a part of the fractal in figure 3 and magnifying that part along the X- and Y-axis with a factor of approximately 1000. Observe the repetitive pattern of the

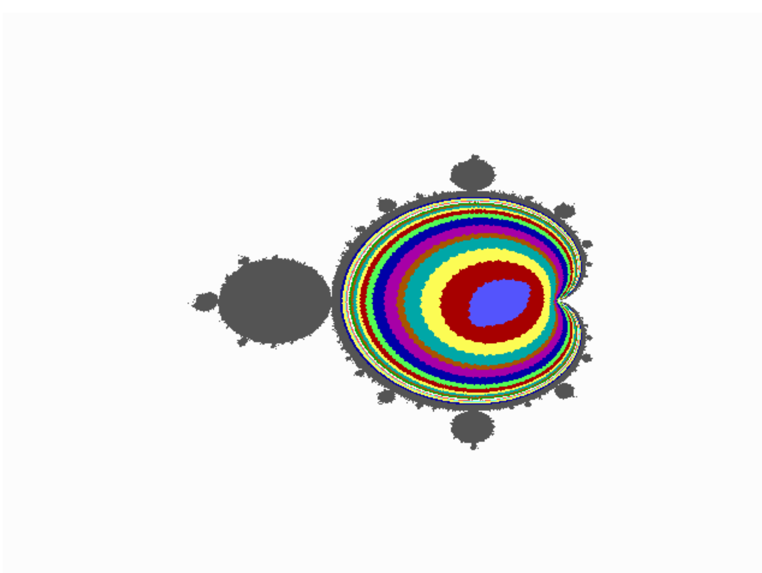

Figure 5: Values of C that don't diverge

Mandelbrot cusps in the fractal of picture 3.

.

There is a different way to represent the complete Mandelbrot set. Instead of colouring the points of C in equation (1) we could colour the found solutions of equation (2) and assign a specific colour (gray) to the values of C that do not diverge after having iterated equation (1) 200 times. The result is depicted in figure 5. This picture is generated with the same values for C and the same accuracy for the found solutions as those that were used in figures 2 and 3. This figure demonstrates clearly how the perimeter of the large cusp is broken (fractured) and how smaller cusps merge from those broken parts. Similarly the perimeters of the smaller cusps are broken and even smaller geometrical entities bulge out of these. The repetitive pattern which is shown so clearly in figure 4 would also appear in figure 5 in magnifying portions of the black coloured area that correspond with the gray areas in figure 4. The colours in the large cusp correspond with the speed with which solutions are found. The more we approach the center of the large cusp, the faster our iteration process goes.

At the onset of this chapter I mentioned that fractals have dimensions that don't have to be integer numbers. Mandelbrot defines a *fractal dimension* to be the quotient of two logarithmic values: the natural log of the number of segments of a line (or curve) divided by the reciprocal value of the ratio of the size of the segment with respect to the segmented line (or curve). The original line AG contains a gap: BE. The gap is replaced by BC, CD and DE. The geometrical entity consisting of AB-BC-CD-DE-EF-FG is called a *generator* and can be used to construct a fractal curve that consist of elements with the same structure as the one displayed in figure 6. This can be accomplished by replacing every line segment of AG by a fraction of the shape of the generator and proceed to do that *ad continuum*. The dimension is here log6/log4, assuming that the length of all 6 segments are 1/4th of the length of the line without a gap: ABEFG.
A similar approach to determine the dimension of Mandelbrot's fractal can be used to the curve of the large cusp in figures 1 and 2. However, determining the length of the gaps in the cusp and the exact structure of the smaller cusps that bulge out of the big one is extremely difficult and can only be approximated. The resulting dimension will however be the same for the calculated dimensions of the smaller cusps that in itself contain gaps and cups that bulge out of them.
Note: Many programs that have been devised to create fractals are not stopping of the iterative process of the attractor (attracting function) when the successive approximations of the Z-value yields a solution. The iterative process continues until the maximum number of iterations exceeds a specified value or the process diverges. Programs that stop after having found a solution are by definition many times faster than those who ignore this and keep processing until the specified number of iterations has been exhausted..

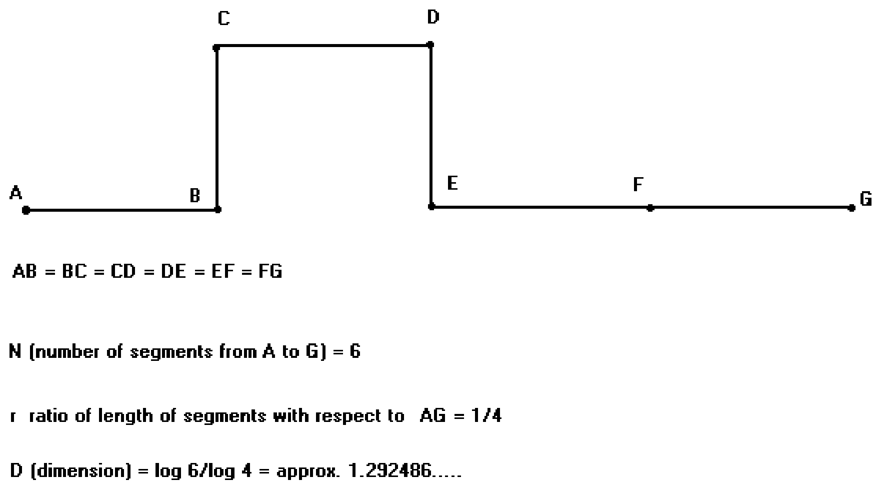

Figure 6: A 'fractured" dimension

2. Compromised fractals.

Equation (1) is just an example of an attractor with which fractals can be generated. After having explored the various details of the fractals of this attractor and the solutions that are a result of solving equation (2) I tried a number of other attractors like $Z_n^3 + C$ and $Z_n^4 + C$ The results, though different from those of equation (1), did not satisfy my curiosity. Their behavioural pattern were in fact similar to those of the attractor of equation (1). To my surprise I discovered a problem in my old research notes that was not too difficult to crack at that time but the mathematical formulation of the problem had a striking similarity with the earlier mentioned attractors. The problem to solve was to find the roots of the equation:

$$Z_n^2 - \frac{\kappa}{Z_n^2} - Z_n + C = 0 \tag{3}$$

Z and C are complex numbers, while κ is a positive real number < 1.
Solutions can be found with the method of successive substitution, as described in the previous chapter. Bringing Z_n to the right hand side of the equation gives us

$$Z_{n+1} = Z_n^2 - \frac{\kappa}{Z_n^2} + C \tag{4}$$

It turns out though that only a limited range of values for κ can be used to determine the solutions for a restricted value of C. Finding those solutions will be dealt with in a different chapter. Although my main objective was to view the fractals that could be generated with the attractor in equation (4), it soon became clear that investigating the values of κ for which equation (4) would still converge (and find solutions) became another part of my investigations. It is clear that for κ = 0 the attractor of equation (1) would be recreated. It appeared that the values to be selected were lying in the range $0.02472 > \kappa > 0$. For example, for $\kappa = 0.01$. I generated the fractal shown in figure 7 and for $\kappa = 0.02471705$ the fractal shown in figure 8. The boundary coordinates of figure 7 are (-2, .8) for the top left corner and (1.2, -2) for the bottom left corner. These values are respectively (-.5, 0) and (1, -1) for figure 8. The maximum amount of iterations are in both cases 200 and the required accuracy for found solutions .0001. The maximum value of the absolute value of Z for which equation (4) is supposed to not diverge is set to the value 2. The black area in figure 7 contains all values of C for which solutions to the equation (3) were found. In figure 8 no black areas are visible. For this value of κ solutions were found for C = .3647905 - i*1647145 and C = .3647945 - i*1647125. It may be expected that between these two values more solutions can be found that satisfy equation (3). However the number of iterations of equation (4) to arrive at these solutions were 719308 and 673787 respectively while the accuracy of found solutions was limited to 10^{-3}. At this point my interest in "pretty" pictures was replaced by an interest in finding the maximum value of κ for which solutions for the equation (3) could still be found and in my quest to finding this value of κ with the method of successive substitution I ran out of computer resources. I do expect though that the maximum value of κ will not exceed 0.027172.

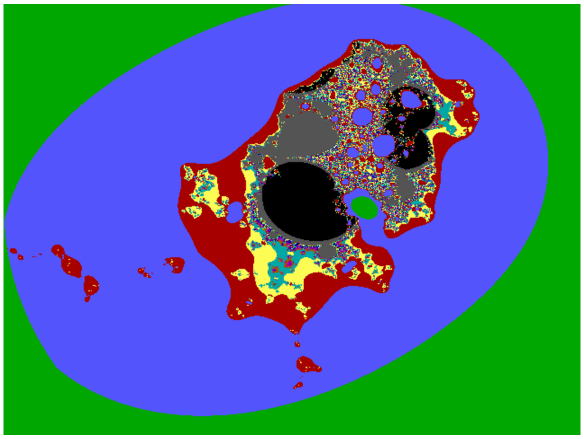

Figure 7: Fractal of equation (4);$\kappa = 0.01$

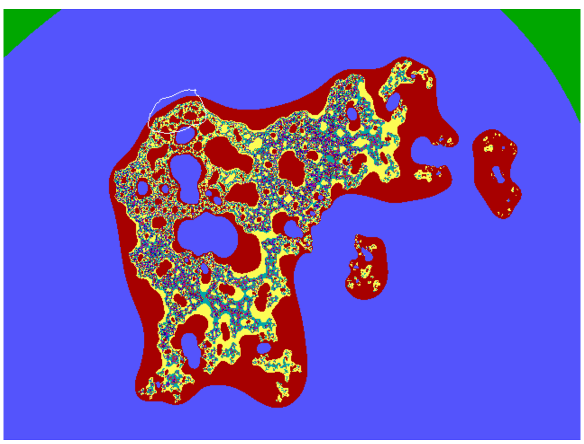

Figure 8: Fractal with $\kappa = 0.0271705$

For an analytical approach in determining this maximum value of κ we have to solve the following equations:

$$a(2b - y + q) - b(x - p) = 0 \quad (5)$$

and

$$a(a - x + p) + b(y - q - b) = \kappa \quad (6)$$

where $C = p + i*q$; $Z = x + i*y$; $b = 2xy$; and $a = x^2 - y^2$. The area were for $\kappa = 0.0271705$ the abovementioned solutions were found is encircled with a pencil in figure 8.

For specific values of C and κ it is possible to evaluate the existence of a solution for Z. For instance: Z = 1 - i for $C = 1 + i*(\kappa/2+1)$, no matter what value we choose for κ. For Z = 3 + i, the value of C becomes $(-125+2\kappa)/25 + i*(-250-3\kappa)/50$. This means that we can always find solutions for equation (3) for all values of κ. However, the method of successive substitution does not help us here. At this point the orientation of my work shifted towards finding solutions for equation (3). This and the use of *relaxation factors* will be discussed in the next chapter.

3. Finding solutions.

So far I have used an unmodified version of the method of successive substitution.. A variant to this method will generate many more solutions than the strict way in which the newly found value Z_{n+1} is inserted in the equation containing a function of Z_n, as in:

$$Z_{n+1} = F(Z_n) + C \qquad (7)$$

The method of the successive substitution does not provide us with all of the possible solutions of the equation:

$$F(Z) - Z + C = 0 \qquad (8)$$

The more successful method consists of inserting a value of Z_{n+1} into equation (7), where Z_{n+1} consists of a combination of the previous value of Z and the newly calculated value of Z.

That combination is achieved with the aid of a *relaxation factor*. The new value used to calculate the right hand side of equation (7) becomes $\rho.Z_n + (1\text{-}\rho).Z_{n+1}$. In this case equation (7) is transformed into:

$$Z_{n+1} = \rho.Z_n + (1-\rho).(F(Z_n) + C) \qquad (9)$$

where ρ is the relaxation factor. This method was first proposed by Wegstein and I will refer to this modified version of the method of successive substitution as Wegstein's method.

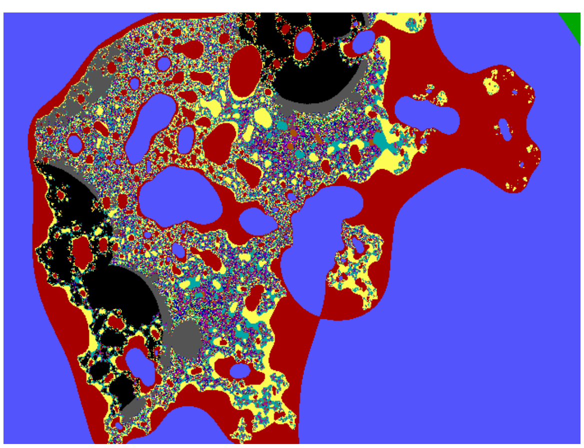

Figure 9: As figure 8 with ρ = 0.1

The amount of found solutions for ρ = 0.1 and ρ = 0.35 are shown in figures 9 and 10. These figures need to be compared with figure 8. The exact same values for generating figure 8 were used to create the, what I call, "compromised" fractals in figures 9 and 10.

As in previous figures, the black area contain the values of C for which equation (3) yields results with the Wegstein's method of using a relaxation factor of 0.1. The gray areas bordering the black ones represent values of C for which the number of iterations necessary to find a solution is greater than 200, the limiting factor that we introduced in order to create the fractals described in this paper. These values of C (the gray areas) will definitely contain solutions to equation (3) if we allow the process to proceed beyond the 200 iterations. This area may also contain values of C for which the process might eventually diverge. But choosing an even larger relaxation factor will also force the process to produce more solutions as is shown in figure 10 which represents the results of finding solutions for equation (3) with a relaxation factor of 0.35. It is also clear that the structure and properties of this compromised fractal have changed dramatically in this situation. Areas containing values of C that already diverged after 4 to 8 steps with the method of successive substitution take much longer with Wegstein's method and an appropriate choice of the relaxation factor. Also patterns that were completely absent in figure 9 emerge here, and extend across the boundaries that were set for the process with which figure 9 was created.

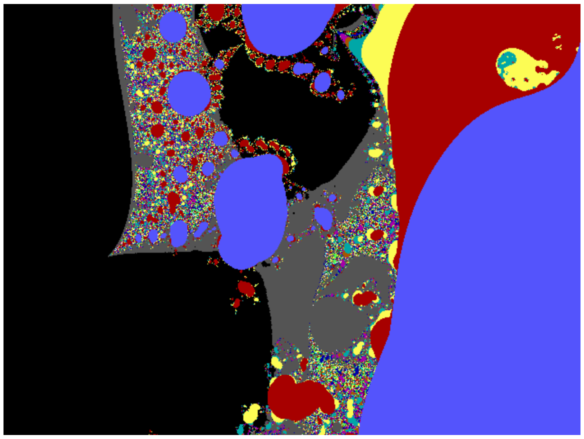

Figure 10: As figure 9 with $\rho = 0.35$

I applied Wegstein's method in trying to find the roots of equation (3) with κ=162.5. This results in a value for C = 8 - i*14.75. (See the end of the previous chapter). Wegstein's method refused to generate Z=3 + i as a root of equation (3). Substituting this value of C into equations (5) and (6) produced the required results 0 and 162.5 respectively. The relaxation factor ρ that had to be used to obtain these results was 0.96. The seed value (initial value) Z with which the iterations started was Z = 1 + i. The amount of iterations required to obtain this solution was 257. The way in finding other solutions will be discussed later.

The beauty of using this method is that it has a convergence speed that is equal or even faster than that of the Newton-Raphson method without having to use the first derivative of the function we are trying to solve. The difficulty in applying this method consists of having to isolating Z from its equation and bringing it to the right-hand side (or zero side of the equation. This difficulty is best shown in solving the equation:

$$X^3 - 2 = 0 \qquad (10)$$

This equation can be written as:

$$X = X^3 + X - 2 \qquad (11) \text{ or}$$

$$X = (2 + 5X - X^3)/5 \qquad (12)$$

Applying the substitution method, equation (11) quickly diverges, no matter what initial value (seed value) of X is being used. Equation (12) however generates a solution with an accuracy of 5 fractional digits in 4 steps. A similar problem may be encountered with the use of relaxation factors. Applying Wegstein's method to equation (1), the equation with which the Mandelbrot set was generated. I discovered that most solutions to equation (2) could be produced by applying a relaxation factor of approx. 0.95. It was remarkable how these compromised fractals changed by generating pictures with equation (1) and using relaxation factors with increasing values. The initial Mandelbrot fractal kept its shape for a long time, but started to display completely different characteristics once the relaxation factor was set to 0.5 or higher.

The technique of applying the method of successive substitution to the abovementioned equation (10) and resulting in equations (11) and (12) was also used to find more roots of equation (3). This equation was substituted by the following 2 equations:

$$Z^2 - \kappa/Z^2 + Z - 2Z + C = 0 \quad (13)$$

and

$$Z^2 - \kappa/Z^2 - 2Z + Z + C = 0 \quad (14)$$

These 2 equations can be written as equations to which the method of successive substitution can be applied:

$$Z_{n+1} = (Z_n^2 - \kappa / Z_n^2 + Z_n + C)/2 \quad (15), \text{and}$$

$$Z_{n+1} = -(Z_n^2 - \kappa Z_n^2 - 2Z_n + C) \quad (16)$$

respectively.

Applying the substitution method without using relaxations factors failed in both cases. I was more successful with Wegstein's method, for which the successive iteration equations can be expressed as:

$$Z_{n+1} = \rho.Z_n + (1-\rho).(Z_n^2 - \kappa / Z_n^2 + Z_n + C)/2 \quad (17)$$

and

$$Z_{n+1} = \rho.Z_n - (1-\rho).(Z_n^2 - \kappa / Z_n^2 - 2Z_n + C) \quad (18).$$

Expression (17) took 980 iterations with a relaxation factor ρ of 0.96 to find the solution Z = 1.57933 + i*3.93098 and expression (18) found the solution with which we started the calculations of κ (162.5) and C (8 - i*14.75), namely Z = 3 + i in 47 iterations with a relaxation factor ρ being 0.9. All these calculations were performed with specially written programs. No fractals were generated and verification of the results was done on examining the calculated numerical results.

4. Feedback systems compared with the methods of successive substitution and Wegstein.

A feedback system consists of a (more or less complicated) process by which input data or input stimuli are transferred in output data or stimuli and where these outputs are fed back as inputs into the process. See figure 11. These kind of systems are used in many industrial systems and disciplines. Quite often a feedback loop is required to put the process in a stable state after a start-up. The fed back data or stimuli are used to enhance the initial data or stimuli until a steady state has been achieved. Applications of feed back systems are to be found in electronic engineering (creating steady states for parts of a circuit or memory), mechanical systems (to make sure that changes in the input do not destabilize the process), and chemical or production systems (making sure that the process is constantly fed with data

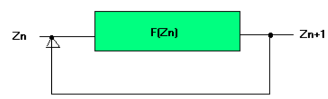

Figure 11: Graphical representation of the method of successive substitution.

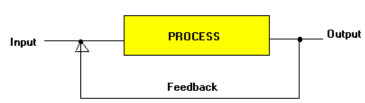

Figure 12: Process with a feedback loop.

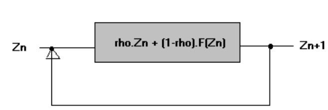

Figure 13: Graphical representation of Wegstein's method

or stimuli in order to provide for a continuous flow of the process).

The method of successive substitution can be compared with a feedback system in that the output Z_{n+1} is fed back into the function $F(Z_n)$. Normally we end this process until Z_{n+1} has reached a value that is acceptable. Mostly this means that feeding Z_{n+1} back into the function $F(Z_n)$ does not change the resulting value of this function. The input value Z_n is assigned the calculated value Z_{n+1}. The comparison with the earlier mentioned feedback system would be complete if we would continue to calculate the function, despite the fact that no further changes occur to the series of values that we calculate for Z_{n+1}. (Figure 12).

Wegstein's method works in exactly the same way, be it that we mix the input value with the calculated output value and feed the combination of these two values back into the function. Also here, the process continues until we achieve a stable situation, that is the subsequent values of Z_{n+1} become stable and change no more. Figure 13 is a graphical representation of Wegstein's method.

When determining the new input value we calculate the product of the relaxation factor ρ and Z_n and add the product of (1-ρ) and $F(Z_{n)}$ to that. In the programs that were used for these calculations I took the old value of Z_n, determined Z_{n+1} with $F(Z_n)$ and combined Z_n and Z_{n+1} with the equation $\rho.Z_n+(1-\rho).Z_{n+1}$. This value is then assigned to Z_n and becomes the new input value. We can view the representations in figures 12 and 13 as a feedback system. These systems will take a certain time to stabilize which is the case once $F(Z_n)$ and $\rho.Z_n+(1-\rho).Z_{n+1}$ yields results that are not changing anymore. In that case initialization of the system has been completed. **Any small changes of the input value Z_n will correct the system and will continue to result in the re-**

quired output. This is of great value in situations where the input value is supposed to remain constant. Slight variations will be corrected immediately by this kind of feedback system. Of course, these corrections are obtained faster with the Wegstein method than the method of successive substitution. The choice of the right value of ρ is important here. Equation (4) shows such a situation. The problem was to find a minimum value of κ and obtain a stable situation for the feedback loop defined with this equation. (The equation represented part of a mathematical model for a heat transfer problem in an experimental Sterling engine).

5. Fractals revisited.

As described in chapter 3 we used the equation $Z_{n+1} = Z_n^2 + C$ (equation 1) to create a Mandelbrot set and its accompanying fractal. The real and imaginary values of C were varied in this process and we started every set of iterations for each value of C with the same initial value of Z, being 0 + i*0. The fractal that was generated that way consisted of all points for which the iterations would diverge. A large number of points were thus excluded from the definition of the fractal since many solutions were found and many other points stayed within the boundary limit of Z (being the absolute value of Z and specified to have the value 2). Another way of generating fractals in which we are almost always assured of diverging iterations is to fix the value of C and vary the initial or starting values of Z. Figure 14 is a fractal of which the value is fixed at C = -.01 + i*0.88. A completely arbitrary choice. We then varied the initial value of Z from -1.6 + i*1.6 and 1.2 - i*1.2. In other words the values of the real part of Z or X-coordinates vary from -1.6 to 1.6 with intervals equal to 3.2/640. The values that correspond with these intervals are mapped to the values of the pixels along the horizontal axis of the screen. Similarly, the values of the imaginary part of Z vary from 1.2 to -1.2 with intervals equal to 2.4/480. These correspond with the values of the pixels along the vertical axis of the screen. With a specified accuracy of any possible solution equation $Z^2 - Z + C = 0$ of 0.0001, a maximum number of iterations

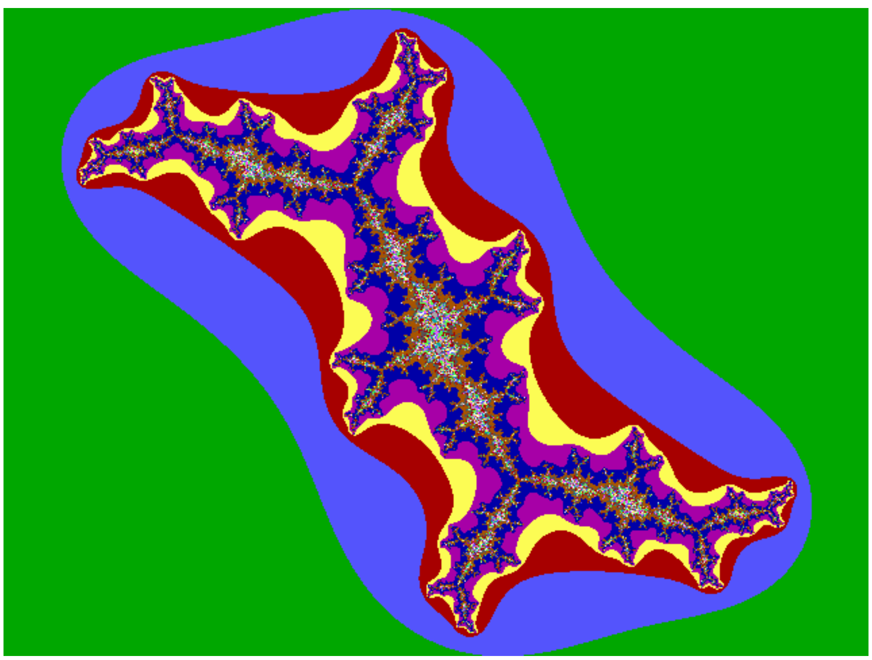

Figure 14: Julia fractal with fixed C value

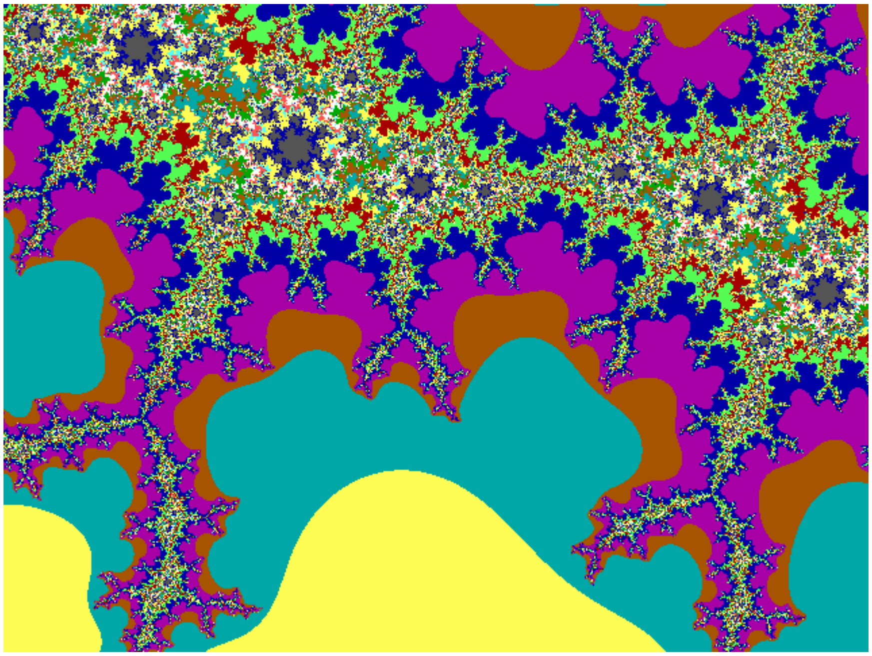

Figure 15: Detailed view of the fractal of figure 14.

set to 200 and an absolute value of Z being equal to 2 we obtained a fractal as shown in figure 14. Every initial value of Z corresponds with a specific way in which equation (1) behaves with respect to the speed with which it diverges. For every specified value of C we will obtain a different fractal that is associated with the use of this equation. This fractal does not resemble the ones describes in chapter 3 at all. However, the repetitive character is still there and is best shown in the detailed view of figure 15 of this fractal. Here the top-left corner of the fractal has an initial Z-value of -0.32- i*0.256 and the value of the bottom-left corner is here -0.96 - i*0.144. During the generation of the fractal it was noticed that one solution was found. The Z-value of this solution was -.3058 + i*0.5467. The dark gray areas in figure 15 represent points of the fractal where the initial value of Z exceeded the maximum amount of 200 iterations. If I would have been focused entirely on generating fractals I should have should these method of generating them, but I would never have been sidetracked into trying to find solutions for the various equations that I used. The fractals generated with this method are called Julia fractals. The name has its origin in that of the French mathematician Gaston Julia (early 20th century).