

# Fractals and Chaos Theory: A Management Primer

By SCOTT E. DAHNE

Advanced Technology Seminar

595.766.31

Fall 1990

## Background

Twenty five years ago, most practicing scientists shared a common set of beliefs about complexity: Simple systems behave in simple ways. Complex behavior implies complex causes. Different systems behave differently. Now, an alternative set of ideas has emerged. Simple systems give rise to complex behavior and complex systems give rise to simple behavior. Most importantly, these laws hold universally irrespective of the system's constituent atoms.<sup>1</sup> The basis for these statements is a new science known as chaos theory which uses, as one of its primary tools, the concept of fractal geometry to explain the interrelationship between simple and complex and chaos and order. There are many terminologies used in this new science which should be defined before starting to learn about the intricacies of the concepts.

Fractals are the graphical representations of mathematical equations used to study chaos. A fractal is a shape made of parts similar to the whole in some way.<sup>2</sup> Fractals were first named by an IBM research scientist named Benoit B. Mandelbrot in the early 1970s. In winter 1975, Mandelbrot was preparing his first paper for publication and needed a name for his shapes, dimensions, and geometry. He had discovered a unifying principle of nature and a mathematical means to express it and need a term to describe this class of "self-similar, scale -invariant" shapes.<sup>3</sup> He was thumbing through his son's Latin dictionary and came across the adjective *fractus*, from the verb *frangere*, to break. The resonance of the main English cognates---fracture and fraction---seemed appropriate. Mandelbrot created the word (noun and adjective, English and French) *fractal*.<sup>4</sup> What Mandelbrot discovered was that as you increase the magnification of any object, there exists within the larger object, some number of smaller objects which can be defined and iterated to make up the larger object. These objects, known as fractal curves, repeat on a smaller and smaller scale as the image is enlarged farther and farther.

The use of a concept known as fractal dimensions allows the researcher to measure the extent to which a one-dimensional thread (such as plotted on a graph) fills up a three-dimensional space. A technique known as the Hausdorff-Besicovitch dimension was developed for describing any curves whose dimension was greater than the standard Euclidian dimension. Mandelbrot used the term fractals to describe all curves whose Hausdorff-Besicovitch dimension is greater than its Euclidian dimension. For example, a

---

<sup>1</sup>J. Gleick, *Chaos: Making a New Science*, Viking Penguin Inc., New York, p. 303-304, 1987

<sup>2</sup>J. Feder, *Fractals*, Plenum Press, New York, p. 11, 1988

<sup>3</sup>J. McDermott, "Dancing to Fractal Time," *Technology Review*, Vol. 92 No. 1, p. 6, January 1989

<sup>4</sup>J. Gleick, *Chaos: Making a New Science*, Viking Penguin Inc., New York, p. 98, 1987

fractal dimension of 2.3 means that a plotted solution doesn't fill up a three-dimensional space but does overfill a two-dimensional space.<sup>5</sup>

One of the most famous fractal images discovered by Mandelbrot has been given the name "Mandelbrot Set". A picture of the set is seen in figure 1. The set consists of complex numbers  $C$  for which the size of  $Z^2+C$  remains finite no matter how many iterations the equation is put through. This set can be described mathematically as  $Z^2+C \rightarrow Z$  where  $Z$  is a complex number allowed to vary and  $C$  is a fixed complex number. All points  $C$  associated with connected Julia sets serve as a table of contents to those Julia sets. The graph of the set plots the number in various colors depending upon whether the number remains essentially the same during its iterations, approaches infinity, or oscillates somewhere in between. The color also indicates the speed at which infinity was approached in those cases.<sup>6</sup>

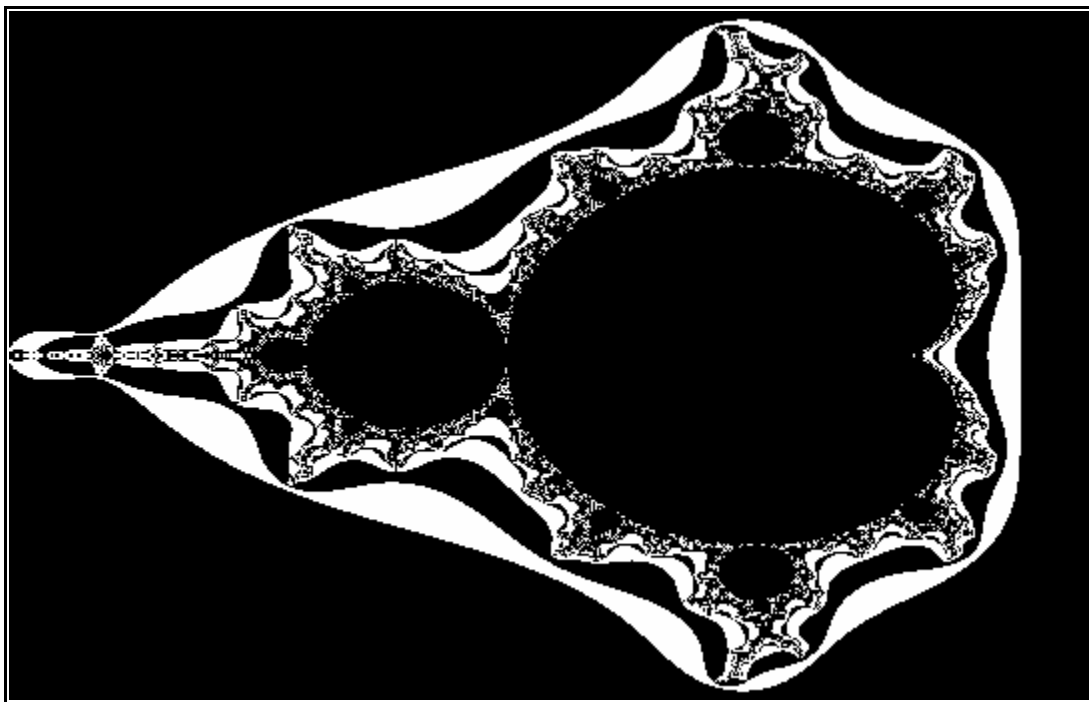


Figure 1

Picture of the Mandelbrot set

The Julia sets, of which the Mandelbrot set is one, can be defined as deterministic fractals which are a result of iterating analytical functions. Julia sets were originally discovered by the French mathematician Gaston Julia (one of Mandelbrot's teachers). Generation of a Julia set from within the Mandelbrot set involves iterating the Mandelbrot

---

<sup>5</sup>G. Goldstein, "Francis Moon: Coming to Terms with Chaos," *Mechanical Engineering*, Vol. 112 No. 1, pp. 44, January 1990

<sup>6</sup>J. Briggs and F.D. Peat, *Turbulent Mirror: An Illustrated Guide to Chaos Theory and the Science of Wholeness*, Harper and Row, New York, pp. 96-97, 1989

function  $Z^2+C \rightarrow Z$  while holding  $C$  constant and varying the initial value of  $Z$  successively. An example of the Julia set from a location within the Mandelbrot set is seen in figure 2.

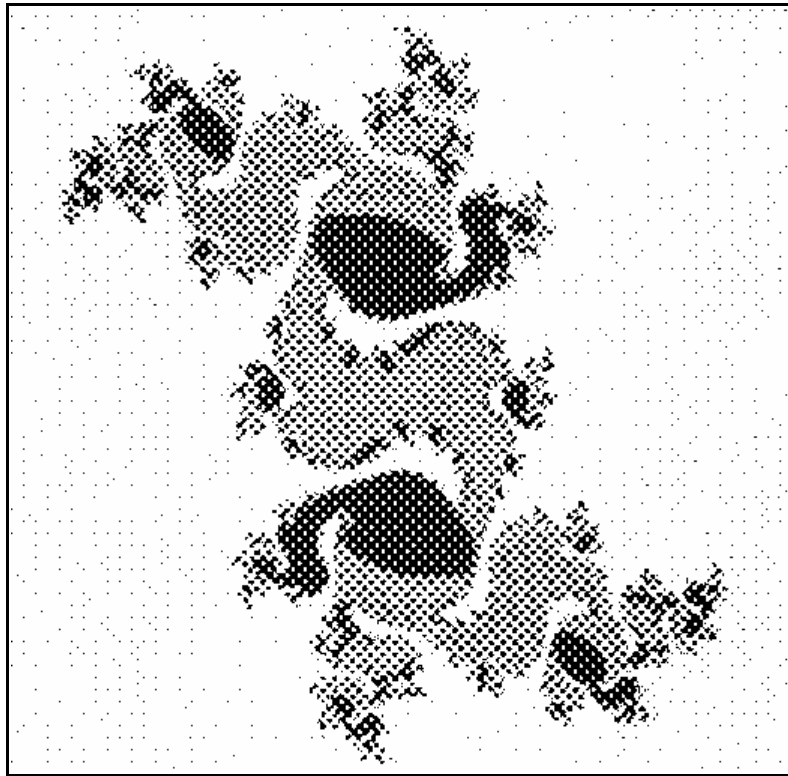


Figure 2

Picture of the Julia set from location  $P=0.238498$ ,  $Q=0.519198$

The term "chaos" was coined by James Yorke, a University of Maryland mathematician to describe "deterministic order". Such activity manifests itself as the seemingly random behavior of well-defined physical systems (systems described by well-known and often simple equations) under certain conditions.<sup>7</sup> Chaos is often referred to as the study of non-linear dynamics.

A Poincare map is a method of viewing the dynamics of a system at discrete times, instead of viewing them as a continuous time trace. The map can also be looked at as a slice through phase space. When the points on a Poincare map are dense in certain areas and scarce in others, they tend towards what is known as attractors. These attractors seem to attract all nearby trajectories in the phase space yet have zero phase-space volume (dissipation). Since the dissipation leads to zero volume, even though the nearby initial conditions separate (due to chaos), there exists a "strange" situation involving the attractor, thus the term strange attractor is used.<sup>8</sup>

---

<sup>7</sup>A.M. Killian, "Playing Dice with the Solar System," *Sky and Telescope*, Vol. 78 No. 2, p.136, August 1989

<sup>8</sup>J.C. Sommerer, "Chaos and Fractals: A 'Heads Up' for Technical Managers", Lecture to JHU/APL Advanced Technology Seminar, November 20, 1990

Now that there is a basic understanding of the concepts and terminology used in working with fractals and chaos theory, we can begin to look at some of the applications for this "new" science and try to understand how it can benefit us in our everyday life.

## Potential Applications and Their Impact

Fractals have a value as art objects and also as a means of representing natural scenes. Fractals also occur naturally in the expressions for mathematical phenomena for the prediction of weather systems, the description of the turbulent flow of liquid, and the growth and decline of populations, among others. Fractals are useful in dimensional transformations used to express and compress graphical data and appear to provide the solutions to many unanswered questions at the frontiers of the physical sciences.<sup>9</sup>

Much of the reason such concepts as Poincare maps, strange attractors, and fractal dimensions are helping engineers is that they allow the engineer to visualize a problem in previously unavailable ways. Let us now take a look at the use of fractals and chaos theory in a number of different applications.

### *Animal Behavior*

Frank H. Heppner, a zoologist with the University of Rhode Island in Kingston has applied chaos theory to explain behaviors seen in flocks of birds and perhaps schools of fish. His approach may prove to be useful in designing spaces to accommodate mass movements of large numbers of people. Heppner filmed flocks of birds at three frames per second and then analyzed the films. Analysis indicated that there was no specific leader to the flock and yet the birds always maintained a state of dynamic equilibrium. Heppner and his associates designed a set of rules to model what they observed and was able to account for the flocks' movements in any number of varying situations. Heppner feels that this work may prove useful in explaining how people interact in large groups. This could be beneficial when designing areas such as aircraft interiors and large auditoriums. If a model can be constructed of how people will react in a situation of panic and confusion, the design of an area could account for this and many lives could potentially be saved.<sup>10</sup>

### *Chemistry*

Chemical reactions in the pores of membranes, excitation trapping in molecular aggregates, exciton fusion in composite materials, and charge recombination in colloids and clouds are all practical examples of "fractal-like" kinetics. The hallmarks of "fractal-like" reactions are anomalous reaction orders and time-dependent reaction rate "constants." Fractal kinetics can be used as a structural tool to investigate the surfaces of catalysts, electrodes, and electronic devices.<sup>11</sup>

At General Electric, fractals became an organizing principle in the study of polymers. The use of fractal-time processes to explain the molecular bonding interaction of

---

<sup>9</sup>R.T. Stevens, *Fractal Programming in C*, M&T Publishing, Redwood City, pp. 24-25, 1989

<sup>10</sup>R. Weiss, "A Flight of Fancy Mathematics: Chaos brings harmony to a birder's puzzle," *Science News*, Vol. 137 No. 11, p.172, March 17, 1990

<sup>11</sup>R. Kopelman, "Fractal Reaction Kinetics," *Science*, Vol. 241 No. 4873, pp.1620-1626, September 23, 1988

materials such as Lexan (a polycarbonate resin used for making bulletproof windows) could allow researchers to slow the aging process of polymers and reduce their tendency to become brittle over time. The use of fractal-time or defect-diffusion modeling also helps to explain relaxation behavior among glasses, polymers, and ceramics. The theory is mathematically simple and can be used to model behavior of most amorphous materials.<sup>12</sup> Application of fractals to polymer work could eventually lead to stronger, more reliable materials which could be produced cheaper due to a better understanding of their physical/chemical structure.

### *Crystal Growth*

By understanding how crystals grow, researchers can fashion materials with better physical, electrical, mechanical, and optical qualities. The use of fractal dimensions to characterize materials provides a new way of looking at crystallization. A material with a fractal dimension  $<2$  will become transparent as it grows; with a fractal dimension  $>2$ , it will become opaque. Application of this to cracks in silicon nitride has found that higher fractal dimensions mean tougher materials.<sup>13</sup> A better understanding of crystallization reactions will allow scientists to enhance materials used in semiconductor manufacturing to provide faster, cheaper, and more reliable materials for the next generations of computer chips.

### *Cosmology*

Edward L. White of UCLA has used fractals to model dust which could explain unexpectedly strong radar echoes observed from comets IRAS-Araki-Alcock in 1983 and Haley in 1985. Radar echoes occurred at a wavelength of 12.6 centimeters while the dust particles are tens of thousands of times smaller than that reflection. Spherical masses of dust would account for the echoes but would also be an improbably large mass for cometary material. Wright's fractal models show fractal aggregates which are fluffy and porous and can account for the radar echoes. The models can also account for the polarization variations from different comets.<sup>14</sup>

Many studies are currently underway to look at the behavior of stellar bodies such as asteroids, moons, and planets. With the use of special high speed computers, it is possible to predict the orbits and positions of stellar bodies to a degree of accuracy which has been previously unattainable. By predicting out to many millions of years in the future, it is possible to see chaotic behavior which would not be noticed in shorter range studies. One such model has been taken out to 845 million years in the future to look at our own solar system. The results of this study show relative stability with the exception of the planet Pluto which shows several different resonance periods ranging from 3.8 million to 600 million years. These multiple resonances with long periods suggests chaotic behavior.<sup>15</sup>

---

<sup>12</sup>I. Peterson, "Time to Relax," *Science News*, Vol. 135 No. 10, pp. 157-159, March 11, 1989

<sup>13</sup>J. McDermott, "Dancing to Fractal Time," *Technology Review*, Vol. 92 No. 1, p. 6, January 1989

<sup>14</sup>Anonymous, "Fractal-Shaped Dust," *Sky and Telescope*, Vol. 78 No. 3, p.240, September 1989

<sup>15</sup>A.M. Killian, "Playing Dice with the Solar System," *Sky and Telescope*, Vol. 78 No. 2, p.140, August 1989

The universe and the rules which bind it together are an area of much deliberation as we turn away from the boundaries of our own planet and begin to look elsewhere for explanations of why things happen in nature. The application of fractals and chaos theory could help to solve many of the riddles which have eluded us in our study of the cosmos.

### *Geophysics*

One of Mandelbrot's original papers on fractal geometry looked at scale invariance applied to the length of the British coastline. It was shown that the length of the coastline varied dependent upon the size of the ruler being used to make the measurements. Changing the scale of the coastline in relation to the altitude from which it was viewed also showed that the coast was composed of identical features which were built upon each other to form larger features which looked the same as the smaller features (scale-invariance). Geological objects exist at many different scales at which their appearance is very similar. Fractal geometry allows a new paradigm for geology or geophysics because one can now view objects as parts of fractal sets which obey certain scaling rules over a range of scales.<sup>16</sup> Geometric objects can now be characterized by functions instead of one number.<sup>17</sup>

### *Hollywood*

Fractals are used to create phenomenally realistic earthly and extraterrestrial landscapes for special effects in movies.<sup>18</sup> Steven Spielberg, George Lucas, and many others are now using fractals to generate alien landscapes which are used as background effects for new movies. Fractals were used to create the "Genesis Effect" in Star Trek III and the Jovian atmosphere in 2010. The initial cost of the computer systems used to perform the calculations is quite large, however, it is a one time expense. The system can be used repeatedly to generate new images. Contrast the fractal techniques with the older method of actually building a set on a sound stage or painting each frame by hand if the scene encompasses motion and the economic benefits of using fractal imagery becomes apparent immediately. The cost of production is not only lowered, but the program can be stored for use on sequels instead of the need to rebuild the entire set or store the stage flats in a warehouse.

### *Image compression*

Commercial image compression algorithms are capable of compressing images to 1/50th of their original binary size without loss of detail. By representing an image as a collection of fractals, the compression can approach a factor of 1,000 or even 10,000. An Atlanta company called Iterated Systems founded by Georgia Institute of Technology mathematician Michael Barnsley has determined that since fractals have the same amount of detail at any scale, and many larger objects are made up of many smaller objects which

---

<sup>16</sup>C.H. Scholz and B.B. Mandelbrot, ed., *Fractals in Geophysics*, Birkhauser Verlag, Basel, p. 1, 1989

<sup>17</sup>C.H. Scholz and B.B. Mandelbrot, ed., *Fractals in Geophysics*, Birkhauser Verlag, Basel, p. 5, 1989

<sup>18</sup>J. Gleick, *Chaos: Making a New Science*, Viking Penguin Inc., New York, p. 114, 1987

are identical, it is only natural to try to represent complicated pictures as a set of fractals repeated many times. Barnsley defines a fractal as having high geometric complexity but a low informational content. Iterated Systems has developed custom hardware and software which helps an operator to develop a "template" -- a rough fractal description -- for each object or group of related objects in a picture. The commercially available system is capable of 1,000:1 compression. (These "templates" can then be fine tuned to make changes in the picture as required.) Although the process is currently manual in nature, Barnsley is in the process of automating it in order to increase the speed of template development and compression. One of the more lucrative areas upon which Barnsley is focusing is the area of High Definition Television (HDTV). By designing an attachment for the HDTV sets which can convert the broadcast image to fractal form and then enhance the background images based upon information from foreground images, the HDTV picture would look perfectly convincing to a viewer. Although the picture wouldn't truly represent what was in front of the camera, the viewer wouldn't be able to discern the "forgery" and the resultant image would look perfectly convincing.<sup>19</sup>

The Defense Advanced Research Projects Agency (DARPA) is interested in Barnsley's work to obtain better image compression techniques for storing digital reconnaissance pictures as well as assisting in image recognition of the satellite images. Since nature produces fractal images while Euclidean shapes are evidence of people, a compressed image of a reconnaissance photo would contain affine maps of fractal images as well as non-fractal images. Algorithms could be used to determine the presence of the Euclidean shapes very rapidly and compare the images to a database of known items for comparison. This technique would be considerably faster than manual interpretation of the photographs. Better compression techniques will also allow for faster interpretation and subsequent transmission of images to the battlefield commands where they are needed during a combat situation.

### *Mechanical Engineering*

Vibrating objects will eventually succumb to fatigue. If the vibration can be predicted through periodicity, failure can be predicted and possibly prevented. If the vibrations are chaotic, failure prediction is almost impossible. The use of chaos theory to predict these chaotic vibrations may eventually enable engineers to create designs which account for them.<sup>20</sup> Fractals can unlock behaviors which appear to be random and unpredictable. As an example, if the vibrations of a buckled beam were completely random, the distribution of points on a Poincare map would be uniform. Since the distribution isn't uniform and tends towards a strange attractor, the vibrations are not truly random in nature.<sup>21</sup>

---

<sup>19</sup>T. Waters, "Fractals in Your Future," *Discover*, Vol. 10 No. 3, pp. 26-28, March 1989

<sup>20</sup>G. Goldstein, "Francis Moon: Coming to Terms with Chaos," *Mechanical Engineering*, Vol. 112 No. 1, pp. 41-42, January 1990

<sup>21</sup>G. Goldstein, "Francis Moon: Coming to Terms with Chaos," *Mechanical Engineering*, Vol. 112 No. 1, pp. 44, January 1990

One engineering professor in Munich is trying to use chaotic dynamics to improve the transmissions in BMW automobiles which are too noisy at low speeds. By relating the rattling of the gears to chaotic theory, he has been able to reorganize the gears to reduce the noise level.<sup>22</sup> This is one of the many applications where the use of chaos theory can be applied to consumer products in order to improve the comfort level of products as well as the safety and reliability aspects.

### *Medicine*

William M. Schaffer is using strange attractors to predict the epidemiology of childhood diseases such as measles and chicken pox. He has created dynamical models which show that, based upon data collected, the traditionally unexplainable yearly variations in measles, which seem random and noisy, follow a strange attractor with a fractal dimension of 2.5.<sup>23</sup> If we are able to predict potential outbreaks of diseases, it would be possible to provide preventative measures before an outbreak occurs and thus hopefully save lives.

Many parts of the body such as neural networks, the bile-duct system, and blood vessels leading to the heart have been found to be fractal in nature. It has been suggested that if the structure of the physiological systems is fractal, then why not their functions?<sup>24</sup> Physicians had always thought that the rhythm of the heart became more chaotic with age. Recent data shows that the older heart tends to be less chaotic. By plotting heart fluctuations, a fractal dimension can be calculated and this dimension decreases with age. This could allow doctors to relate physical age to chronological age of the body. We could have our fractal rate taken in the future instead of our pulse.

The application of fractals and chaos in medicine could eventually prolong life by allowing doctors to predict when and where disease will strike as well as predicting when particular parts of the body will fail. If we can understand more about how the body functions, it becomes easier to understand how certain diseases affect the body and thereby focus on a more effective means of combating those diseases. Who knows, one day the use of fractals and chaos theory may show how cancer attacks the body and eventually lead to a way of curing this disease.

### *Meteorology*

One of the first examples of chaos theory to be described was by Edward Lorenz at MIT in the early 1960s. Lorenz was working on computer models for weather and climate and found that very small changes in the initial values of his variables could lead to extreme differences in the outcome of the simulation. Although the initial equations being used were very simple and well-behaved, Lorenz was observing an occurrence of "sensitive dependence on initial conditions" which is a characteristic of chaotic systems.<sup>25</sup> Lorenz's

---

<sup>22</sup>G. Goldstein, "Francis Moon: Coming to Terms with Chaos," *Mechanical Engineering*, Vol. 112 No. 1, pp. 47, January 1990

<sup>23</sup>J. Gleick, *Chaos: Making a New Science*, Viking Penguin Inc., New York, p. 316, 1987

<sup>24</sup>J. McDermott, "Dancing to Fractal Time," *Technology Review*, Vol. 92 No. 1, p. 8, January 1989

<sup>25</sup>A.M. Killian, "Playing Dice with the Solar System," *Sky and Telescope*, Vol. 78 No. 2, p.136, August 1989



original graphs showed what has been defined as a strange attractor. Figures 3-5 show the Lorenz attractor as viewed from three different perspectives. Although it appears that the attractors are tracing a path or orbit, they are actually built up iteratively within the phase space.

If it were possible to accurately predict weather patterns, many billions of dollars and thousands of lives could be saved annually by the prediction of hurricanes, tornados, and other natural forces which wreak havoc with our planet.

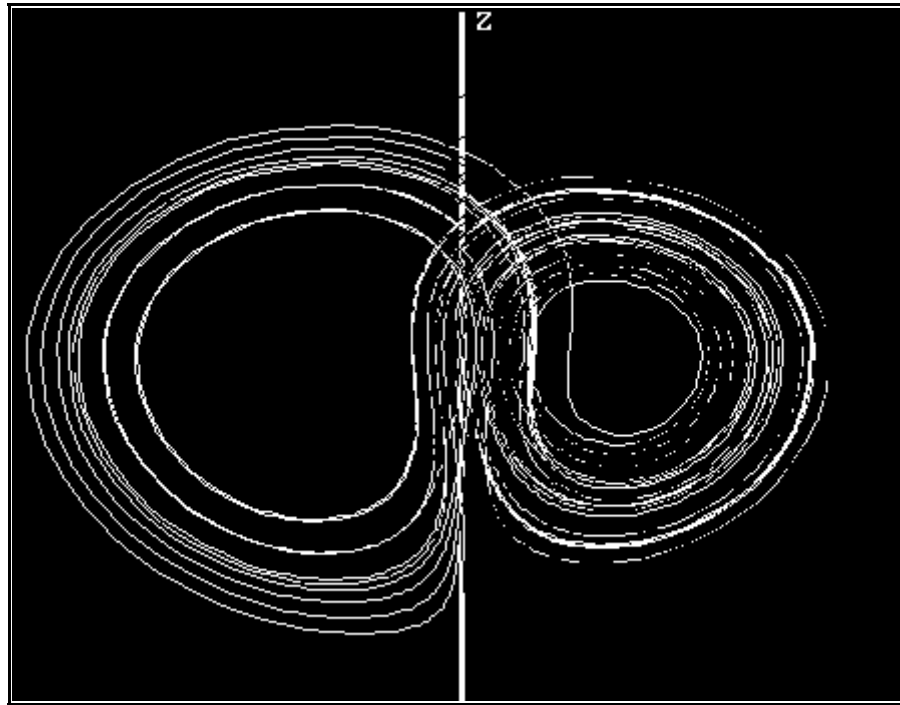


Figure 3

Lorenz Attractor Projected on YZ plane

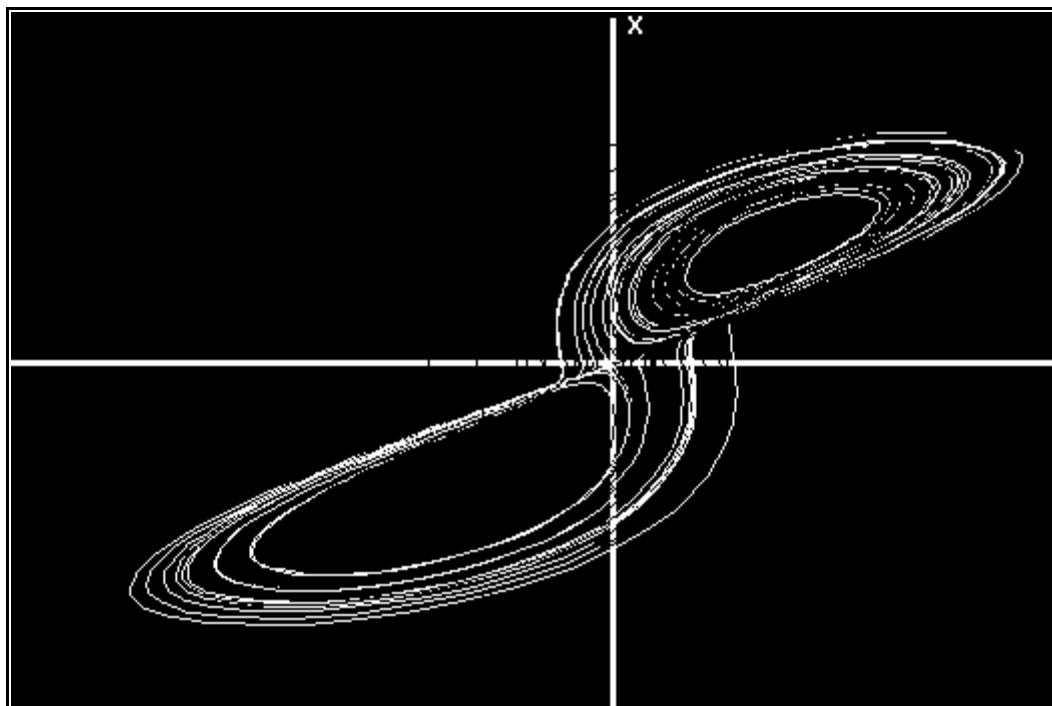


Figure 4

Lorenz Attractor Projected on XY plane

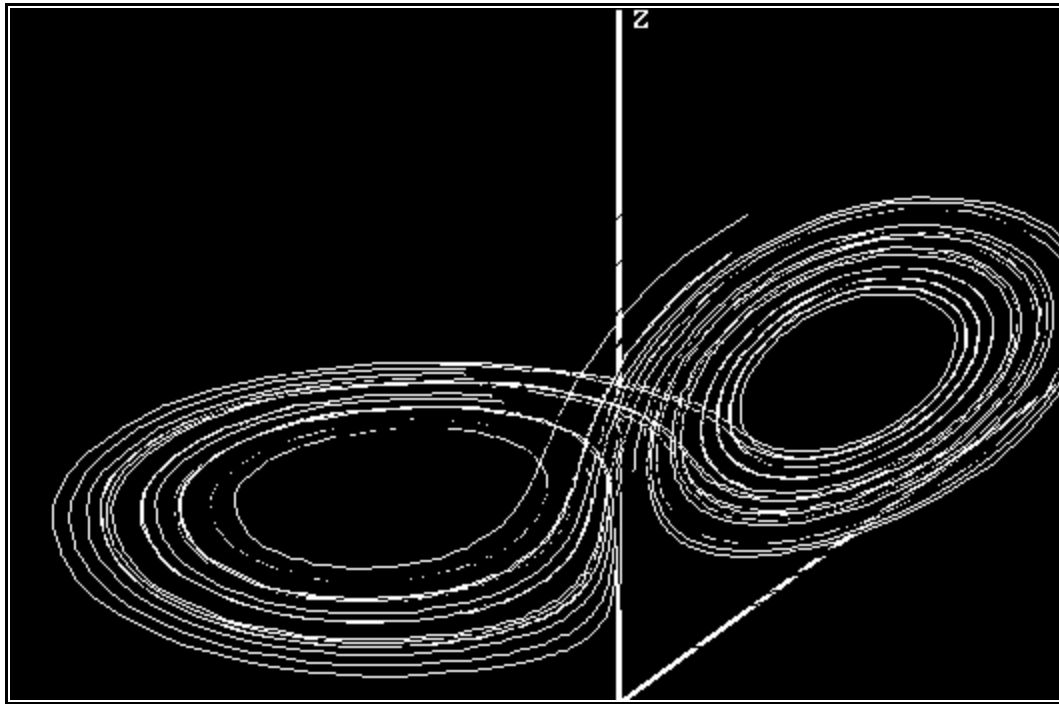


Figure 5

### Three-Dimensional View of Lorenz Attractor

#### *Military*

Fractals are being investigated for use in modeling command and control as well as predicting the outcome of battles. The issues of command and control involve complexities which cannot be modeled by current computer simulations of combat. The self-similarity of fractals is ideally suited to modeling the hierarchy of the military command and control structure which is virtually identical at all levels. The flow of battlefield information is fractal in nature. A battlefield may be looked at as a complex non-linear system with the order of the military units and the chaos of the battle interacting. The boundary between the chaos and order is fractal in nature. By measuring the fractal boundary, it may be possible to predict where the order ends and the chaos begins. Chaos in higher level units might be catastrophic and could mean the collapse of a unit. Predicting when this would happen could allow an upper level unit to regain control and restore order among the unit.<sup>26</sup> Application of these principles could save many lives in a combat situation and allow for redistribution of logistics elements on an accurate basis instead of through crisis management.

#### *Nuclear reactor safety*

---

<sup>26</sup>G. Zorpette, "Fractals: Not Just Another Pretty Picture," *IEEE Spectrum*, p. 31, October 1988

At General Electric, secret work was conducted with fractals in nuclear reactor safety.<sup>27</sup> The flow of gases in heat exchangers and cooling water flowing around fuel rods can cause vibrations which lead to fatigue and eventually failure. Engineers are looking at chaotic behavior to try and explain these flow induced vibrations.<sup>28</sup> This work could greatly enhance the safety of nuclear reactors and prevent disasters such as Three Mile Island and Chernobyl.

### *Oil Pumping*

The development of a "diffusion limited aggregation" model to describe chaotic growth is helping oil companies to recover oil from wells which are almost empty. Typically, water is pumped into the wells to displace the remaining oil. The boundary between the oil and water often breaks into fractal-shaped fingers and the displacement won't occur. Addition of a polymer to restrict the mixing and create a less fractal interface tends to stabilize the displacement process and allow the oil to be recovered.<sup>29</sup> The events of recent months in the Middle East have served to show us that our dependence on foreign oil must be reduced. Methods for increasing our own oil production will serve to reduce this dependence. If methods such as discussed above will enhance our ability to remove additional oil from older wells, we could recover more product per well than originally estimated.

### *Simulations*

The use of chaos theory in simulations has potential due to the non-linearity of many simulations. Due to the sensitive dependence on initial conditions, a simulation could give widely varying results if run as a linear system and so many sets of initial data surrounding the initial conditions must be run to determine the divergence of the results. Running the simulation as a non-linear system could give more credible results to the model.

Fractal landscapes used in training simulators can be extremely advantageous in training fighter pilots. Current simulators have a set number of situations which are run repeatedly during the training situation from a database of available choices. In order to train a pilot to cope with all of the terrain where they will be flying, the entire terrain must be filmed and stored as data. The maintenance of a database for this type of information is not economically feasible. An additional drawback to the current type of simulator is that the pilots eventually become familiar with the simulation and learn to react to the set of conditions being presented. We need to train pilots to react to general flying conditions instead of a particular set of conditions. In an actual combat situation with new and constantly changing situations, the pilot may not know how to react if they have only trained to a particular set of scenarios. The use of fractals to generate artificial reality on demand lends an element of realism to a simulator since the scenario can be varied

---

<sup>27</sup>J. Gleick, *Chaos: Making a New Science*, Viking Penguin Inc., New York, p. 114, 1987

<sup>28</sup>G. Goldstein, "Francis Moon: Coming to Terms with Chaos," *Mechanical Engineering*, Vol. 112 No. 1, pp. 43, January 1990

<sup>29</sup>J. McDermott, "Dancing to Fractal Time," *Technology Review*, Vol. 92 No. 1, p. 8, January 1989

endlessly as required by the training situation and the trainee learns to react to random situations instead of a predetermined set which is repeated over and over.<sup>30</sup>

### **Technical Obstacles/Implications to a Manager in a High Technology Organization**

There are several key obstacles to the use of fractals and chaos theory in the majority of areas where they may be applied. The basic principles of the field concern the application of higher level mathematics to ordinary processes. In order to be able to apply the mathematics correctly, it is necessary to understand both the principles of fractals and chaos as well as the field to which they are being applied. The majority of the engineers in industry today have little or no knowledge of chaos theory since most of the major research universities teach nonlinear dynamics or chaos theory at the graduate level while most of the engineers come out of undergraduate programs and are not exposed to the work in the graduate courses.

Equipment required to work in the field of chaos and fractals includes high speed digital computers and high resolution output devices. The cost of the hardware required could be out of the reach of many companies. Installation of high end computers requires additional support networks which can be costly in themselves.

The personnel working on applications of chaos and fractals are many and varied. They range from individuals in the ivory towers of IBM's research centers to hobbyists with interests in generating pretty artwork. A company desiring to get into working with chaos and fractals must be able to find scientists or engineers who have expertise in the area of business which is of concern as well as the field of chaos. It is uncertain whether chaos theory applied in one specific area can be easily transferred to other areas. Matching the proper people with projects has always been a key concern of the technical manager and this becomes even more critical with the field of chaos theory. One way of getting involved with fractals and chaos theory might be to develop a partnership with a university working in the field and allowing graduate students to assist your company with its entry into the field. These graduate students might eventually be hired by the company to begin a program for looking into future applications of the technology.

Chaos theory as applied to everyday life is a relatively new branch of science and thus there are very few scientists and engineers from most areas attempting to apply it to their situations. Since there is very little support for the technology, it would not be cost effective for an engineer to try applying it to their work unless they are willing to become totally immersed in the field of chaos and set up their own network of supporting entities. Agencies such as DARPA are committed to funding technologies which have no immediate value but could have potential value in the future. The majority of our corporations are profit driven and cannot afford to become involved in a technology which won't show an immediate increase in profitability.

One other area of concern is the proprietary nature of work done in the area of fractals and chaos theory. Since the application of this technology is so new, any discoveries by a company will tend to be kept proprietary in order to maintain a

---

<sup>30</sup>J.C. Sommerer, "Chaos and Fractals: A 'Heads Up' for Technical Managers", Lecture to JHU/APL Advanced Technology Seminar, November 20, 1990

competitive advantage. This veil of secrecy will tend to eliminate the usual "networking" among high technology workers which helps to advance the technology.

Once a manager decides to try applying fractals and chaos theory to their area of work, a support network must be established for the technology. A plan must be put together for investigating potential uses of the technology and a method for obtaining the required resources must be established. A methodology must be established for evaluating the results of the initial research in order to decide where the major efforts will be focused. Attention should be paid to looking at applying the new science to an area which will show the best return on investment in the shortest time. If one or two small successes can be shown, it will be easier to obtain additional support for the work. Many engineers are quick to try to apply a new technology to their work simply to be able to say that they are using that technology. This type of situation can be dangerous if the technology is not fully understood. Application of fractals to oil drilling or image compression is of relatively low risk and a failure of the technology in either of these areas would not be a major safety problem. Application of fractals/chaos theory to nuclear reactor safety could lead to potentially devastating results if the technology caused problems during its application to the existing processes.

The normal cultural aspects of an engineering environment may not be conducive to the application of chaos theory since many of the ideas are outside the realm of the ordinary engineering education. The individuals working on applying the new concepts must be given the latitude and support to do their work outside of the normal organizational structure. However, when the time comes for applying the results of the work, even more effort must be placed upon helping to integrate the new technology within existing structures.+

## APPENDIX A

### Program listings for generation of figures

(Listings are reprinted from R.T. Stevens, *Fractal Programming in C*, M&T Publishing, Redwood City, 1989)

```
/*
cmandel = ADVANCED PROGRAM TO MAP THE MANDELBROT SET
*/
#include <stdio.h>
#include <math.h>
#include <dos.h>
#include <process.h>
#include "tools.h"

const int maxcol = 639;
const int maxrow = 349;
const int max_colors = 16;
int CURSOR_X=0,CURSOR_Y=0,col,row;
int max_iterations = 512;
int max_size = 4;
int LINEWIDTH=1, OPERATOR=0x00, ANGLE, XCENTER, YCENTER;
unsigned long int PATTERN=0xFFFFFFFF;
unsigned char PALETTE[16]={0,1,2,3,4,5,20,7,56,57,58,59,60,61,62,63};
int colors[16]={0,1,2,3,4,5,20,7,56,57,58,59,60,61,62,63};
double Q[350];
float TXMax,TXMin,TYMax,TYMin;
union LIMIT XMax,YMax,XMin,YMin,Pval,Qval;

char file_name[13] = {"mandel00.pcx"};

main()
{
    double P,deltaP, deltaQ, X, Y, Xsquare, Ysquare;
    int i,color, row, col,error,response,repeat=0x30,start_col;

    printf("\nEnter background and 15 other colors separated by"
           " commas: ");
    scanf("%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d",
          &colors[0],&colors[1],&colors[2], &colors[3],
          &colors[4],&colors[5],&colors[6],&colors[7],
```

```

        &colors[8],&colors[9],&colors[10], &colors[11],
        &colors[12],&colors[13],&colors[14],&colors[15]);
printf("Enter screen number (00 - 99): ");
file_name[6] = getche();
file_name[7] = getche();
getch();
error = restore_screen(file_name);
for (i=0; i<16; i++)
    setEGAPalette(i,colors[i]);
if (error == 0)
{
    printf("\nEnter 0 to generate first Mandelbrot screen");
    printf("\nEnter 1 to exit: ");
    response = getche();
    if (response != 0x30)
        exit(0);
    else
    {
        XMax.f = 1.2;
        XMin.f = -2.0;
        YMax.f = 1.2;
        YMin.f = -1.2;
        start_col = 0;
        setMode(16);
        cls(7);
    }
}
else
{
    if (error < 639)
    {
        start_col = 8 * (error/8);
        remove(file_name);
    }
    else
    {
        start_col = 0;
        move_cursor(0,15,0,0);
        move_cursor(1,15,CURS_X,CURS_Y);
        cls(7);
        XMax.f = TXMax;
        XMin.f = TXMin;
        YMax.f = TYMax;
        YMin.f = TYMin;
    }
}

```



```

}

while (repeat == 0x30)
{
    deltaP = (XMax.f - XMin.f)/(maxcol);
    deltaQ = (YMax.f - YMin.f)/(maxrow);
    Q[0] = YMax.f;
    for (row=1; row<=maxrow; row++)
        Q[row] = Q[row-1] - deltaQ;
    P = XMin.f + start_col * deltaP;

    for (col=start_col; col<maxcol; col++)
    {
        if (kbhit() != 0)
        {
            save_screen(0,0,col,349,file_name);
            exit(0);
        }
        for (row=0; row<=maxrow; row++)
        {
            X = Y = Xsquare = Ysquare = 0.0;
            color = 1;
            while ((color<max_iterations) && ((Xsquare +
            Ysquare) < max_size))
            {
                Xsquare = X*X;
                Ysquare = Y*Y;
                Y = 2*X*Y + Q[row];
                X = Xsquare - Ysquare + P;
                color ++;
            }
            plot(col, row, (color % max_colors));
        }
        P += deltaP;
    }
    save_screen(0,0,639,349,file_name);
    getch();
    gotoxy(10,24);
    printf("Enter '0' to run another plot, '1' to quit: ");
    repeat = getche();
    if (repeat == 0x30)
    {
        move_cursor(0,15,0,0);
        move_cursor(1,15,CURSOR_X,CURSOR_Y);
    }
}

```

```

    XMax.f = TXMax;
    XMin.f = TXMin;
    YMax.f = TYMax;
    YMin.f = TYMin;
    start_col = 0;
    setMode(16);
    cls(7);
  }
}
□

```

```

/*
cjulia = ADVANCED PROGRAM TO MAP THE JULIA SETS
*/
#include <conio.h>
#include <stdio.h>
#include <math.h>
#include <dos.h>
#include <process.h>
#include "tools.h"
#include <ctype.h>

const int maxcol = 639;
const int maxrow = 349;
const int max_colors = 16;
int CURSOR_X=0,CURSOR_Y=0,col,row;
int max_iterations;
int max_size = 4;
int LINEWIDTH=1, OPERATOR=0x00, ANGLE, XCENTER, YCENTER;
unsigned long int PATTERN=0xFFFFFFFF;
unsigned char PALETTE[16]={0,1,2,3,4,5,20,7,56,57,58,59,60,61,62,63};
int colors [8]= {0,1,2,3,4,5,20,7};
float P,Q;
float TXMax,TXMin,TYMax,TYMin;
union LIMIT XMax,YMax,XMin,YMin,Pval,Qval;

char file_name[13] = {"mandel00.pcx"};
char file_name2[13] = {"julia000.pcx"};

FILE *f1,*f2;

main()
{
    float deltaX, deltaY, X, Y, Xsquare, Ysquare;
    int i,color, row, col,error,response,repeat=0x30,start_col;

    printf("Enter '0' to quit, '1' to expand "
           "or finish julia, '2' for new plot: ");
    do
        repeat = getch();
    while ((repeat != 0x30) && (repeat != 0x31) && (repeat != 0x32));

    while (repeat != 0x30)
    {
        printf("\nEnter number of iterations: ");
        scanf("%d",&max_iterations);
    }

```

```

printf("\nEnter background and six other colors separated by"
      " commas: ");
scanf("%d,%d,%d,%d,%d,%d",&colors[0],&colors[1],
      &colors[2],&colors[3],&colors[4],&colors[5],
      &colors[6]);
if (repeat == 0x32)
{
    printf("\nEnter mandelbrot set number "
          "(00 - 99): ");
    file_name[6] = getche();
    file_name[7] = getche();
    setMode(16);
    if ((isdigit(file_name[6])) && (isdigit(file_name
        [7])))
    {
        error = restore_screen(file_name);
    }

    else
        exit(0);
    getch();
    start_col = 0;
    move_cursor(2,15,0,0);
    XMax.f = 1.8;
    XMin.f = -1.8;
    YMax.f = 1.2;
    YMin.f = -1.2;
    P = Pval.f;
    Q = Qval.f;
    cls(7);
}
else
{
    printf("\nEnter julia screen number "
          "(00 - 99): ");
    file_name2[6] = getche();
    file_name2[7] = getche();
    if ((isdigit(file_name2[6])) &&
        (isdigit(file_name2[7])))
        error = restore_screen(file_name2);
    else
        exit(0);
    if (error == 0)
        exit(0);
    else

```

```

        {
            if (error < 639)
            {
                start_col = 8 * (error/8);
                remove(file_name2);
                P = Pval.f;
                Q = Qval.f;
            }
            else
            {
                move_cursor(0,15,0,0);
                move_cursor(1,15,CURSOR_X,CURSOR_Y);
                XMax.f = TXMax;
                XMin.f = TXMin;
                YMax.f = TYMax;
                YMin.f = TYMin;
                start_col = 0;
                cls(7);
            }
        }
    }
    if (colors[0] <64)
    {
        setEGApalette(0,colors[0]);
        setEGApalette(1,colors[1]);
        setEGApalette(2,colors[2]);
        setEGApalette(3,colors[3]);
        setEGApalette(4,colors[4]);
        setEGApalette(5,colors[5]);
        setEGApalette(6,colors[6]);
    }
    deltaX = (XMax.f - XMin.f)/(maxcol);
    deltaY = (YMax.f - YMin.f)/(maxrow);
    for (col=start_col; col<=maxcol; col++)
    {
        if (kbhit() != 0)
        {
            Pval.f = P;
            Qval.f = Q;
            save_screen(0,0,col,349,file_name2);
            exit(0);
        }
        for (row=0; row<=maxrow; row++)
        {
            X = XMin.f + col * deltaX;

```

```

        Y = YMax.f - row * deltaY;
        Xsquare = 0;
        Ysquare = 0;
        color = 0;
        while ((color < max_iterations) &&
                ((Xsquare + Ysquare) < max_size))
        {
            Xsquare = X*X;
            Ysquare = Y*Y;
            Y = 2*X*Y + Q;
            X = Xsquare - Ysquare + P;
            color++;
        }
        gotoxy(24,24);
        if (color >= max_iterations)
        {
            color = ((int)((Xsquare + Ysquare)
                           *6.0))%6 + 1;
        }
        else
            color = 0;
        plot(col, row, color);
    }
}
Pval.f = P;
Qval.f = Q;
save_screen(0,0,639,349,file_name2);
getch();
gotoxy(1,24);
printf("File Name: %s Enter '0' = quit, '1' = expand "
       "plot, '2' = new plot: ",file_name2);
repeat = getche();
}
getch();
}

```

□

```

/*
lorenz = PROGRAM TO PLOT LORENZ ATTRACTOR
*/
#include <dos.h>
#include <stdio.h>
#include <math.h>
#include "tools.h"

float radians_to_degrees(float degrees);

const int maxcol = 639;
const int maxrow = 349;

int LINEWIDTH = 3, OPERATOR = 0;
int color = 15;
unsigned long int PATTERN = 0xFFFFFFFF;
unsigned char PALETTE[16]={0,1,2,3,4,5,20,7,56,57,58,59,60,61,62,63};
float rad_per_degree=0.0174533,x_angle=45,y_angle=0,z_angle=90;
union LIMIT XMax,YMax,XMin,YMin,Pval,Qval;
char file_name[13] = {"lorenz00.pcx"};

main()
{
    double x,y,z,d0_x,d0_y,d0_z,d1_x,d1_y,d1_z,d2_x,d2_y,d2_z,
           d3_x,d3_y,d3_z,xt,yt,zt,dt,dt2,third=0.333333333,
           sx,sy,sz,cx,cy,cz,temp_x,temp_y,old_y;
    int i, j, row, col, old_row, old_col;

    x_angle = radians_to_degrees(x_angle);
    sx = sin(x_angle);
    cx = cos(x_angle);
    y_angle = radians_to_degrees(y_angle);
    sy = sin(y_angle);
    cy = cos(y_angle);
    z_angle = radians_to_degrees(z_angle);
    sz = sin(z_angle);
    cz = cos(z_angle);
    for (j=0; j<3; j++)
    {
        color = 4;
        LINEWIDTH = 3;
        x = 0;
        y = 1;
        z = 0;
        setMode(16);
    }
}

```

```

if (j == 0)
{
    old_col = y*9;
    old_row = 9*z - 240;
    drawLine(-320,-238,319,-238,15);
    drawLine(0,-238,0,239,15);
    gotoxy(79,24);
    printf("Y");
    gotoxy(42,1);
    printf("Z");
}
if (j == 1)
{
    old_col = y*10;
    old_row = 10*x;
    drawLine(-320,0,319,0,15);
    drawLine(0,-238,0,238,15);
    gotoxy(79,12);
    printf("Y");
    gotoxy(42,1);
    printf("X");
}
if (j == 2)
{
    old_col = y*9;
    old_row = 9*z - 240;
    drawLine(-320,-238,319,-238,15);
    drawLine(0,-238,0,239,15);
    drawLine(0,-238,319,82,15);
    gotoxy(79,24);
    printf("Y");
    gotoxy(42,1);
    printf("Z");
    gotoxy(79,8);
    printf("X");
}

}

LINEWIDTH = 1;
dt = 0.01;
dt2 = dt/2;
for (i=0; i<8000; i++)
{
    d0_x = 10*(y-x)*dt2;
    d0_y = (-x*z + 28*x - y)*dt2;
    d0_z = (x*y - 8*z/3)*dt2;

```



```

    xt = x + d0_x;
    yt = y + d0_y;
    zt = z + d0_z;
    d1_x = (10*(yt-xt))*dt2;
    d1_y = (-xt*zt + 28*xt - yt)*dt2;
    d1_z = (xt*yt - 8*zt/3)*dt2;
    xt = x + d1_x;
    yt = y + d1_y;
    zt = z + d1_z;
    d2_x = (10*(yt-xt))*dt;
    d2_y = (-xt*zt + 28*xt - yt)*dt;
    d2_z = (xt*yt - 8*zt/3)*dt;
    xt = x + d2_x;
    yt = y + d2_y;
    zt = z + d2_z;
    d3_x = (10*(yt - xt))*dt2;
    d3_y = (-xt*zt + 28*xt - yt)*dt2;
    d3_z = (xt*yt - 8*zt/3)*dt2;
old_y = y;
    x += (d0_x + d1_x + d1_x + d2_x + d3_x) * third;
    y += (d0_y + d1_y + d1_y + d2_y + d3_y) * third;
    z += (d0_z + d1_z + d1_z + d2_z + d3_z) * third;
    if (j == 0)
    {
        col = y*9;
        row = 9*z - 240;
    if (((col<0) && (old_col >= 0)) || ((col > 0)
        && (old_col <= 0)))
        color++;
    }
    if (j == 1)
    {
        col = y*10;
        row = 10*x;
    if (((col<0) && (old_col >= 0)) || ((col > 0)
        && (old_col <= 0)))
        color++;
    }
    if (j == 2)
    {
    if (((y<0) && (old_y >=0)) || ((y > 0)
        && (old_y <=0)))
        color++;
        temp_x = x*cx + y*cy + z*cz;
        temp_y = x*sx + y*sy + z*sz;

```

```

        col = temp_x*8;
        row = temp_y*7-240;
    }
    drawLine(old_col,old_row,col,row,color);
    old_row = row;
    old_col = col;
}
save_screen(0,0,639,349,file_name);
getch();
}
}

```

```

float radians_to_degrees(float degrees)
{
    float angle;

    while (degrees >= 360)
        degrees -= 360;
    while (degrees < 0)
        degrees += 360;
    angle = rad_per_degree*degrees;
    return angle;
}

```

```

/*
save_screen() = save screen to disk file
*/

```

```

void save_screen(int x1, int y1, int x2, int y2, char file_name[])
{

    extern union LIMIT XMax,XMin,YMax,YMin,Pval,Qval;
    extern unsigned char PALETTE[16];
    int i,j,k,add1,add2,number,num_out, line_length, end,
        start_line, end_line;
    unsigned char ch,ch1,old_ch,red,green,blue;
    FILE *fsave;

    while (file_name[6] < 0x3A)
    {
        if (fsave = fopen (file_name,"rb") != NULL)
        {

```

```

        file_name[7]++;
        if (file_name[7] >= 0x3A)
        {
            file_name[7] = 0x30;
            file_name[6]++;
        }
    }
    else
    {
        fsave = fopen(file_name,"wb");
        fputc(0x0A,fsave);
        fputc(0x05,fsave);
        fputc(0x01,fsave);
        fputc(0x04,fsave);
        putw(x1,fsave);
        putw(y1,fsave);
        putw(x2,fsave);
        putw(y2,fsave);
        putw(640,fsave);
        putw(350,fsave);
        ch = 0x00;
        for (i=0; i<16; i++)
        {
            red = (((PALETTE[i] & 0x20) >> 5) |
                    ((PALETTE[i] & 0x04) >> 1)) * 85;
            green = (((PALETTE[i] & 0x10) >> 4) |
                    (PALETTE[i] & 0x02)) * 85;
            blue = (((PALETTE[i] & 0x08) >> 3) |
                    ((PALETTE[i] & 0x01) << 1)) * 85;
            fputc(red,fsave);
            fputc(green,fsave);
            fputc(blue,fsave);
        }
        fputc(0x00,fsave);
        fputc(0x04,fsave);
        start_line = x1/8;
        end_line = x2/8 + 1;
        line_length = end_line - start_line;
        end = start_line + line_length * 4 + 1;
        putw(line_length,fsave);
        for (i=0; i<4; i++)
            fputc(XMax.c[i],fsave);
        for (i=0; i<4; i++)
            fputc(XMin.c[i],fsave);
        for (i=0; i<4; i++)

```

```

        fputc(YMax.c[i],fsave);
for (i=0; i<4; i++)
    fputc(YMin.c[i],fsave);
for (i=0; i<4; i++)
    fputc(Pval.c[i],fsave);
for (i=0; i<4; i++)
    fputc(Qval.c[i],fsave);
for (i=92; i<128; i++)
    fputc(' ',fsave);
for (k=y1; k<y2; k++)
{
    add1 = 80*k;
    number = 1;
    j = 0;
    add2 = (start_line);
    old_ch = read_screen(add1 + add2++,0);
    for (i=add2; i<end; i++)
    {
        if (i == end - 1)
            ch = old_ch - 1;
        else
        {
            if ((add2) == end_line)
            {
                j++;
                add2 = (start_line);
            }
            ch = read_screen(add1 + add2,
                               j);
        }
        if ((ch == old_ch) && number < 63)
            number++;
        else
        {
            num_out = ((unsigned char)
                        number | 0xC0);
            if ((number != 1) ||
                ((old_ch & 0xC0) ==
                 0xC0))
                fputc(num_out,fsave);
            fputc(old_ch,fsave);
            old_ch = ch;
            number = 1;
        }
        add2++;
    }
}

```

```

        }
    }
    fclose(fsave);
    break;
}
}
nosound();
} □

```