CHAOTIC DESCENT METHOD AND FRACTAL CONJECTURE

Vojin Jovanovic
Design and Manufacturing Institute
Stevens Institute of Technology
Hoboken, New Jersey

ABSTRACT

Very often, when dealing with computational methods in engineering analysis, the final state depends so sensitively on the system's precise initial conditions that the behavior becomes unpredictable and cannot be distinguished from a random process. This outcome is rooted in an intricate phenomenon labeled "chaos", which is a synonym for unpredictable events in nature. In contrast, chaos is a deterministic feature that can be utilized for problems of finding global solutions in both nonlinear systems of equations as well as optimization. The focus of this paper is an attempt to utilize computational instabilities in solving systems of nonlinear equations and optimization theory that resulted in development of a new method, Chaotic Descent. The method is based on descending to global minima via regions that are the source of computational chaos. Also, one very important conjecture is presented that in the future might lead the way towards direct solving of the systems of simultaneous nonlinear equations for all the solutions.

Keywords: optimization, chaos, fractals, Newton-Raphson, nonlinear equations.

1. INTRODUCTION

Finding global solutions of nonlinear optimization problems is a difficult task still not resolved to a satisfactory level by the general theory of optimization. The problem is difficult due to the fact that in the general case the behavior of the function throughout the space is unknown. Nonlinear programming methods are based on exploiting the local properties of the objective function and converging to the nearest solution that in many cases is only a local minimum. However, several local minima may exist and the corresponding function values may differ substantially. We can only determine its local characteristics, which is why there are so many well-developed local optimization methods [5]. When the function is well-behaved, i.e. convex, quadratic functions, the chance of finding the global optima is greatly enhanced. Situations where practical results may be achieved are described in [11].

Designing algorithms that obtain global solutions is very difficult, since there is no local criterion for deciding whether a local solution is global. During the past three decades, the subject of global optimization and related problems resulted in the development of a variety of deterministic and stochastic methods for finding global solutions. Deterministic methods include enumerative techniques, cutting planes, branch and bound, decomposition based approaches, bilinear programming, interval analysis, interior point methods, and approximate algorithms for large-scale problems. Stochastic methods include simulating annealing, pure random search techniques, and the clustering method. Due to the problems intrinsic complexity, most of these methods are based on numerical iterative techniques that are tuned up for the fastest possible convergence and the satisfaction of different criteria. However, iteration, as a dynamic process, is greatly overlooked among optimization experts, even though iteration is the heart of nonlinear programming. The surprise is even greater when it is learned that the tool for understanding iteration of functions, at least in one-dimensional case, was developed by mathematicians Julia and Fatou in the beginning of this century. This paper is intended to show how this tool can be utilized in search for global minima even though the tool relies only on local characteristics of the objective functions. It will be described how sensitivity to initial conditions influences numerical computations as pointed out in [8].

In this paper, some of the concepts from the Theory of Iterations of Functions are introduced with a little help from computer graphics, which is one reason for the recent popularity of this theory. It is unfortunate that its originators [9,11] whose work was motivated by a one-page paper by Cayley [2], have never been able to see the pictures seen today with the assistance of computers.

Throughout the paper the analyticity of the functions (the satisfaction of Cauchy-Riemann equations) will be emphasized, since this is the main reason for obtaining some remarkable results about iterations. An intriguing conjecture that arises from the theory of analytic functions will be shown where readers' familiarity with complex numbers will be assumed. The Chaotic Descent method and all the reasoning used to develop it will also be presented. This theoretical discussion will be supported with a numerical experiment.

2. ITERATION OF COMPLEX ANALYTIC FUNCTIONS

As a first step, it is necessary to define the notion of iteration of a complex analytic function $f(z)$, where z is a complex number. The iteration is a repeated application of function f on a complex variable z. Specifically, if we select a starting point z_0 in the complex plane $\boldsymbol{C}$ and then apply f repeatedly on z_0, we will be constructing, in turn, the points z_0, $z_1=f(z_0)$, $z_2=f(z_1)$, etc. In this fashion, iteration can be viewed simply as a feedback process, the behavior of which depends on the operator f. When this operator is linear, it is quite easy to predict the outcome of the iteration process. However, complications arise when f is a nonlinear operator. In these instances, the outcome of iterations is unpredictable, and its study reveals quite intricate properties.

We begin with the simplest possible nonlinear operator $f(z)=z^2$. There are two possible outcomes of its iteration process. Namely, with repeated application of $f(z)$ starting with a point z_0 inside of the unit circle, the iteration converges to 0. When starting with a point outside of the unit circle, the iteration converges to infinity. These two final states of the process are called the fixed points of f because $f(0)=0$ and $f(\infty)=\infty$. In general, fixed points mean $f(z^*)=z^*$ and periodic points mean $f^{\,n}(z)=z$, where $f^{\,n}$ represents n-fold composition of f, i.e. f o f o…$f(z)$. Therefore, $f^{\,n}(z)$ is the n^{th} iterate $f(f(f(\ldots(f(z)))$ of z. The least positive integer n for which $f^n(z)=z$ is called the period of the periodic orbit, where given z_0 belongs to $\boldsymbol{C}$, the orbit of z_0 under f is the sequence of points z_0, z_1, z_2, and $z_n=f(z_{n-1})$ for $n=1,2,3$, the fixed point z^* is

1. Attractive if $0<|f'(z^*)|<1$,
2. Superattractive if $f'(z^*)=0$,
3. Repelling if $|f'(z^*)|>1$
4. Neutral if $|f'(z^*)|=1$.

Periodic points of period n are similarly classified by replacing f with f_n in the above definition. The reason for this terminology is as follows: If z_0 is an attracting or superattracting fixed point, then there is an open neighborhood $\boldsymbol{U}$ of z_0 having the property that $f^{\,n}(z) \to z_0$ as $n \to \infty$ for each z in $\boldsymbol{U}$. The set of all points whose orbits converge to z_0 is called the ***basin of attraction*** of z_0. In view of these definitions, and using the example $f(z)=z^2$, we can classify the two fixed points 0 and ∞ as superattracting. The fixed point $z_0=1$ is repelling. This point belongs to the unit circle and it is interesting to examine the dynamics of f on the unit circle $\{z: |z| =1\}$. The circle has the striking property of being both forward and backward invariant under f; that is, each point of the circle has its entire history and future lying on it. It can be shown that all the points on the unit circle are repelling periodic points. The closure of all repelling points is called a Julia set and denoted with ***J***(*f*). The complement of the Julia set is called the Fatou set, or stable set, and it is denoted with ***F***(*f*). The dynamics of f on the unit circle are defined as chaotic because it is unpredictable, indecomposable and recurrent. An iterated map, or Dynamical system, is considered unpredictable if it exhibits sensitive dependence on initial conditions. In other words, given any initial state z_0, there must be a nearby state w_0 whose orbit diverges from that of z_0. To be defined as indecomposable, a Dynamical system must have an orbit that eventually enters any pre-assigned region, no matter how small, in the plane. Finally, a Dynamical system exhibits recurrence if, given an initial condition z_0, there is another initial condition w_0 arbitrarily close to z_0 that is periodic. The dynamics of f on the unit circle are chaotic.

The study of the iterations is greatly aided by examining the basins of attraction for the functions in question. To illustrate this, the basin of attraction for $f(z)=z^2$ is represented in Figure 1.

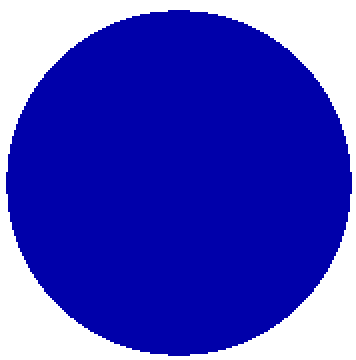

Figure 1. Basin of attraction for $f(z)=z^2$.

Its construction is simple, but it offers a lot of insight into the iteration process. Every initial point for which its orbit converges to z=0 is colored with a dark color, while every point for which its orbit converges to infinity is left uncolored. Because there are only two possible outcomes, the basin of attraction contains two regions:

colored, and uncolored. Note that the boundary between these two regions is smooth (the unit circle) and, as mentioned before, this represents the Julia set. The two regions also represent the Fatou set, which is the complement of the Julia set. Looking at Figure 1 it might appear that nothing significant is happening; however, things change when we add a bit more nonlinearity to our operator. Suppose that we modify this operator slightly, taking $f(z)=z^2+c$ where c is a small complex number. It is easy to see that we still have $f^n(z) \to w$ if z is small, where w is the fixed point of f close to 0, and that $f^n(z) \to \infty$ if z is large. Again, the Julia set is the boundary between these two types of behavior, but it turns out that $\boldsymbol{J}$ is now a fractal curve (Figure 2)

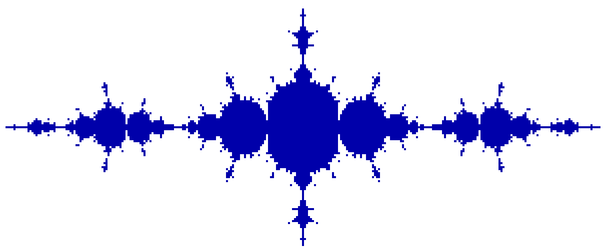

Figure 2. Basin of attraction for $f(z)=z^2 - 1.25$.

This remarkable property is the reason for such interest in the science of iteration. A fractal curve is a special kind, because it is neither a line nor a surface, but something in between. Its main characteristic is that it is self-similar on an infinitely small scale. No matter how close we zoom into a small window around a boundary point that belongs to the Julia set, we see shapes of original objects repeating themselves (Figure 3). A good collection of fractal images can be found in [10].

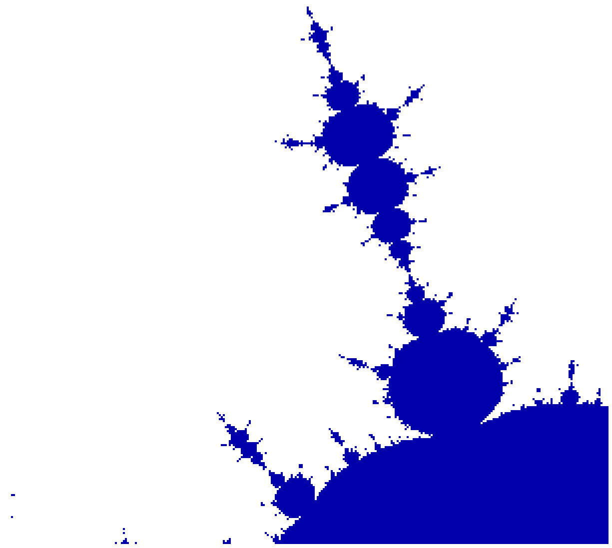

Figure 3. Zooming into the basin of attraction

for $f(z)=z^2 - 1.25$.

To understand this phenomenon completely, one theorem exists which belongs to the realm of advanced complex analysis. This remarkable theorem states the result on which this application is based, and for which the entire above introduction was necessary. Before stating this theorem and showing its consequences, we need to define a family of normal analytic functions and present some helpful propositions. For details refer to [3] and [4].

Let $\boldsymbol{U}$ be an open set in $\boldsymbol{C}$, and let $f_n : \boldsymbol{U} \to \boldsymbol{C}$ be a family of complex analytic functions (i.e. functions differentiable on $\boldsymbol{U}$ in the complex sense). The family $\{f_n\}$ is said to be normal on $\boldsymbol{U}$ if every sequence of functions selected from $\{f_n\}$ has a subsequence, which either 1) converges uniformly on every compact subset of $\boldsymbol{U}$, or 2) converges uniformly to ∞ on $\boldsymbol{U}$. Also, the family $\{f_n\}$ is not normal at z_0 if the family fails to be a normal family in every neighborhood of z_0. To evaluate whether a family of analytic functions is normal at a point, the following two propositions are most useful.

Proposition. Suppose $\{f_n\}$ is a sequence of analytic functions that converges uniformly on a domain U to a map f. Then f is analytic in U and moreover $\lim_{n\to\infty} f_k^{(k)}(z) = f^{(k)}(z)$.

Proposition. Let f be analytic and suppose that z_0 is a repelling periodic point for f. Then the family of iterates of f is not normal at z_0.

Since $\boldsymbol{J}(f)$ is the closure of all the repelling points, by applying the above propositions it follows that the family of iterates fails to be normal at any point in $\boldsymbol{J}(f)$. One of the most important consequences of the failure to be a normal family at a given point is that the family of functions must then assume virtually every value in any neighborhood of the point. This result is known as Montel's theorem.

Montel's theorem. Suppose $\{f_n\}$ is a family of analytic functions defined on a domain U. Suppose there exist $a,b \in \boldsymbol{C}$, $a \neq b$, such that $f_n(z) \neq a$ or b for any n and any $z \in U$. Then $\{f_n\}$ is a normal family in U.

It is important to note here that Montel's theorem is applicable to **any** analytic function. The consequence of this theorem is the following corollary that is the basis for the method represented in this paper.

Corollary. Let f be an analytic map. Let z_0 belong to $\boldsymbol{J}(f)$ and let $\boldsymbol{U}$ be a neighborhood of z_0. Then the iterates $f^n(\boldsymbol{U})$ omit, at most, one point in $\boldsymbol{C}$.

Proof. If $f^n(\boldsymbol{U})$ omitted two points, then $\{f_n\}$ would be a normal family in U.

This is a very strong result and in the following section the practical consequences it has on solving the system of nonlinear equations and optimization theory will be shown.

3. COMPLEX NEWTON-RAPHSON METHOD

Perhaps the most widely used of all root-locating formulas is the Newton-Raphson method. This method can be derived from the Taylor series expansion and is represented by Equation (1)

$$x_{n+1} = f(x_n) = x_n - \frac{g(x_n)}{g'(x_n)} \tag{1}$$

where $f(x_n)$ is called the Newton-Raphson function. Geometrically, this formula gives an estimate of a root of a nonlinear function g. We take its linear approximation at some initial point x_n and solve for an intersection with the x-axis. We use this intersection x_{n+1} later as a new initial point for a new linear approximation. Eventually, if we are "sufficiently close" to a root of the original function, the iteration process will settle in that root. This approximation process brings very rapid convergence of the method, but it also creates some problems widely stated in numerical textbooks. Experiments show that the Newton-Raphson (NR) has the tendency to oscillate around a local maximum or minimum where such oscillations may persist. Also, it is possible that for an initial guess that is close to one root, the iteration may jump to a location several roots away. This tendency to move away from the area of interest is due to the fact that near zero slopes are encountered. However, in view of the previous section, it is quite easy to make sense of the aforementioned problems. Additionally, as shown before, these problems are useful when one searches for all the roots of a nonlinear equation.

Let us begin with a simple example. It is required to solve equation

$$g(x) = x^3 - x = 0 \tag{2}$$

For this simple function all of the solutions are x_1 x_2 $x_{3.}$ Constructing the Newton-Raphson function, we obtain $x_{n+1} = x_n - (x_n^3 - x_n)/\ (2\ x_n^2 - 1)$. Now, instead of considering only real numbers, we can substitute z for x to obtain $z_{n+1} = z_n - (z_n^3 - z_n)/(2\ z_n^2 - 1)$. Therefore, initial points for iterating Equation (2) can be chosen anywhere in the complex plane. Equation (2) can be viewed now as an iteration of an analytic function $f(z_n)= z_n\text{-}(z_n^3\text{-}z_n)/\ (2\ z_n^2\text{-}1)$. This is where the discussion from the previous section will be beneficial. For a better understanding of the iterative process of this function, it is useful to create a picture of its basins of attraction (Figure 4).

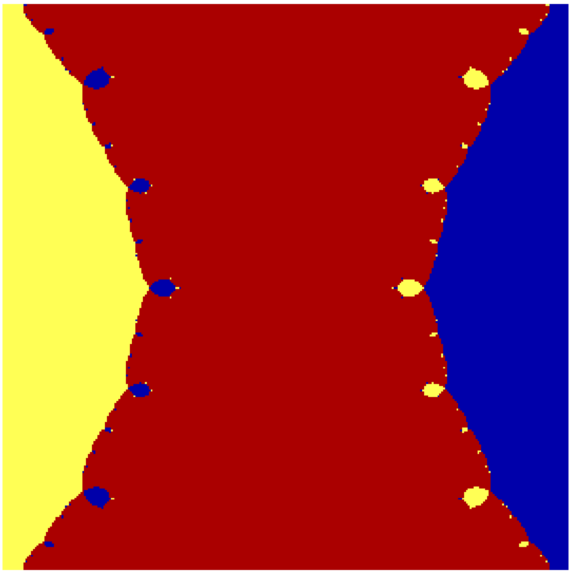

Figure 4. Basin of attraction for NR method

for $g(z)=z^3 - z$.

In Figure 4, three shades of gray represent three possible outcomes of the iteration, i.e., the three roots of the equation we want to solve $z_1=1$, $z_2= -1$, $z_3=0$. Note the ragged boundary between each of the basins of attraction and all three shades of gray on the boundary. The boundary represents the points that belong to the Julia set of *f*. The important characteristics of Julia sets that must hold in any case of iteration of an analytic function have already been discussed. Zooming into this intricate boundary at a given point, we obtain a snowflake picture of self-similar regions of the whole domain. In fact, no matter how close we zoom in we "see" the whole plane through the iterative eyes of the function *f* because of Montel's theorem. In other words, we know that when starting from any neighborhood of the point that belongs to the Julia set, iterates of *f* will cover the whole complex plane and we will find all the roots of the equation that we want to solve. Moreover, the pictures of basins of attraction we see will assume fractal shapes.

Therefore, if one chooses to solve an equation for all the roots, it is not important to be "sufficiently close" to a root, as stated in numerous numerical texts. Rather, it is necessary to find a point that belongs to a Julia set and examine its neighborhood. When such a point is located we know that we will "in the iterative sense" be infinitely close to all the roots of the equation. Another important property of the Julia set should be noted; a point on the boundary of one of the domains of attraction will be on the boundary of all of them. Mathematically, this statement is represented with Equation (3)

$$J(f) = \partial A(z_1) = \ldots = \partial A(z_n) \tag{3}$$

where ∂A is the boundary of the domain of attraction of root z_n. This is another remarkable result and, again, it is a consequence of Montel's theorem.

To summarize, a numerical method for finding all the roots of a nonlinear equation has existed for a long time. However, it was necessary that Julia and Fatou developed the theory of iterations so that it could be realized that the Newton-Raphson method is a greater tool than one might imagine.

4. SOLVING FOR ALL THE ROOTS OF A NONLINEAR EQUATION BY LOCATING $z \in J(F)$

In the preceding section, we found that in order to solve for all the roots of a nonlinear equation, we need to locate **only one** point that belongs to the Julia set of the Newton-Raphson function, and then iterate a small neighborhood of this point to obtain all the roots of the equation. Using the iteration theory we are guaranteed to find all of them, because Montel's theorem states that the neighborhoods of points in J are spread right across the complex plane by iterates of the NR function.

The question that arises now is how to find a point that belongs to a Julia set. To answer this question, we should keep in mind that the Julia set is the closure of all repelling periodic points. Therefore, we should search for one of the periodic points within any period. In [6] and [7] a method that locates a period two repelling point was presented. The primary interest there was in finding all the roots along the real line. However, if there are no real solutions to the equation, the formula developed in that paper could also be used with complex numbers to locate a period two repelling point which, in this case, will be a complex number.

One important advantage of the period two point is that in sampling along the real line in its vicinity, we will find all the ***real*** roots of the equation, which might be desirable if we do not need the complex roots. Locating a period two point is fine for this application, however, it might be computationally involved. On the other hand, the Newton-Raphson method offers another point that belongs to a Julia set and can be easily found. To understand this we should recall that the Julia set is a backward invariant under an iterated function, in this case

NR function. This means that all the points that iterate to repelling points of NR also belong to the Julia set. When the Newton-Raphson method is applied to polynomials it can be shown that infinity is a fixed repelling point ($f(\infty)=\infty$) and therefore belongs to the Julia set. Now, all the points that iterate to infinity must also belong to the Julia set. These points are called critical points and occur where the slope of the polynomial is zero. They are called this because at these points the NR function "blows up".

Using our example, we find critical points by solving $\frac{dg(z)}{dz} = 3z^2 - 1 = 0$ which yields $z_1=1/\sqrt{3}$ and $z_2=-1/\sqrt{3}$. Now it can be seen that $f(\pm 1/\sqrt{3})=\infty$. Therefore, both of these points belong to the Julia set. In the case we are to solve non-polynomial functions, it cannot be stated that infinity is a fixed repelling point (i.e. $tan(\infty)$ does not exist), but that the points that lead to ∞ are the points where the family of iterates fails to be normal. According to the definition (i.e. $tan(\pi/2)=\infty$ is a valid statement; at $\pi/2$ family of *tan* functions fails to be normal), we can again apply Mantel's theorem. Therefore, to solve for the roots of $g = 0$ we only need to examine the neighborhood of one of the roots of $\frac{dg(z)}{dz} = 0$.

In summary, to find all the roots of a nonlinear equation g = 0 we need to find any solution z_c *to* $\frac{dg(z)}{dz}=0$ *(by above* z_c *is guarantied to belong to the Julia set) and use any neighborhood of* z_c *to iterate f(z) from it in order to find all the roots of g = 0.*

The cost of this procedure is executing only one extra Newton-Raphson search on $\frac{dg(z)}{dz} = 0$. Which represents a minimal preprocessing since the Newton-Raphson method has quadratic convergence. Once this Julia set point is located one performs the search for the roots of the equation as specified before.

It is important to stress that, as opposed to the period two point, the point that leads to infinity will not provide all the real roots if we sample only along the real line in its vicinity, so we must sample using complex numbers around z_c. It bears mentioning again that any drawing of basins of attraction would yield fractal pictures. Both procedures appear very simple; however, their foundation rests on the very deep and complicated results from the theory of iteration of analytic functions. Numerical examples once a point in a Julia set is located can be found in [6].

5. ITERATION OF VECTOR FUNCTIONS

In this section, we will build on the previous results and introduce a further extension which enables us to obtain all the solutions to a system of nonlinear equations as well as finding a global minimum to the nonlinear programming (NLP) problem.

One might expect that solving a system of equations can be achieved by locating a point that belongs to the boundary of basins of attraction and by using the multivariable Newton-Raphson. Unfortunately, such a direct analogy is impossible. The previous method works out at this point only for an analytic function in the complex plane. For vectors of analytic functions, the situation is different and it is not known where the mathematical theory of iterating vector analytical functions stands. Nevertheless, it is worth examining situations with the multivariable Newton-Raphson method in order to gain some insight into what is happening.

Going back to solving g = 0 by the Newton-Raphson method, it is useful to take a closer look at the real and imaginary part of iterative Newton-Raphson function. This makes it more clear how great a role analyticity of g plays; i.e. satisfaction of Cauchy-Riemann (CR) equations.

It is required to solve g(z) = u + v i = 0, where u and v are a real and imaginary part respectively. Cauchy-Riemann equations are $u_x = v_y \qquad u_y = -v_x$. The derivative of a complex function is $g'(z) = u_x + i\,v_x$ or $g'(z) = v_y - i\,u_y$ where $z = x + i\,y$. To solve the problem by the Newton-Raphson method, one needs to construct a NR function that is $f(z) = z - \dfrac{g(z)}{g'(z)}$. Broken down to functions u and v we have

$$x_{k+1} + i\,y_{k+1} = x_k + i\,y_k - \frac{u + i\,v}{u_x + i\,v_x} \tag{4}$$

Multiplying with conjugate pairs yields

$$x_{k+1} + i\,y_{k+1} = x_k + i\,y_k - \frac{u + i\,v}{u_x + i\,v_x}\,\frac{u_x - i\,v_x}{u_x - i\,v_x} \tag{5}$$

$$x_{k+1} + i\,y_{k+1} = x_k + i\,y_k - \frac{u\,u_x + v\,v_x + i\,(v\,u_x - u\,v_x)}{{u_x}^2 + {v_x}^2} \tag{6}$$

Therefore, this actually represents the iteration of two functions

$$\begin{aligned} x_{k+1} &= x_k - \frac{u\,u_x + v\,v_x}{{u_x}^2 + {v_x}^2} \\ y_{k+1} &= y_k - \frac{v\,u_x - u\,v_x}{{u_x}^2 + {v_x}^2} \end{aligned} \tag{7}$$

By iterating NR of $g(z)$ we actually iterate a system of two equations NR_x and NR_y where u and v satisfy the CR equation as well as NR_x and NR_y.

Now, let us examine what happens when we try to solve a system of two equations $u = 0$ and $v = 0$, which satisfy CR equations with the multivariable Newton-Raphson method. The Newton-Raphson method is given as $\begin{Bmatrix} x_{k+1} \\ y_{k+1} \end{Bmatrix} = \begin{Bmatrix} x_k \\ y_k \end{Bmatrix} - J^{-1} \begin{Bmatrix} u \\ v \end{Bmatrix}$, where J is a Jacobian matrix. Let us first construct the Jacobian matrix.

$$J = \begin{bmatrix} u_x & u_y \\ v_x & v_y \end{bmatrix} \text{ and } J^{-1} = \frac{1}{|J|}\,Adj\,J = \frac{1}{|J|} \begin{bmatrix} v_y & -u_y \\ -v_x & u_x \end{bmatrix}. \tag{8}$$

Using CR equations we have

$$J^{-1} = \frac{1}{u_x v_y - v_x u_y} \begin{bmatrix} v_y & -u_y \\ -v_x & u_x \end{bmatrix} = \frac{1}{{u_x}^2 + {v_x}^2} \begin{bmatrix} u_x & v_x \\ -v_x & u_x \end{bmatrix} \tag{9}$$

Substitution yields

$$\begin{Bmatrix} x_{k+1} \\ y_{k+1} \end{Bmatrix} = \begin{Bmatrix} x_k \\ y_k \end{Bmatrix} - \frac{1}{{u_x}^2 + {v_x}^2} \begin{bmatrix} u_x & v_x \\ -v_x & u_x \end{bmatrix} \begin{Bmatrix} u \\ v \end{Bmatrix} \tag{10}$$

which is the same as Equation (7). Therefore, using the Newton-Raphson method for two equations that satisfy CR conditions is equivalent to iterating the complex function to yield fractals.

Now, the following can be stated.

$$J^T = \begin{bmatrix} u_x & v_x \\ u_y & v_y \end{bmatrix} \tag{11}$$

Computing *Adj* of *J* and using CR we obtain

$$Adj\ J = \begin{bmatrix} v_y & -u_y \\ -v_x & u_x \end{bmatrix} = \begin{bmatrix} u_x & v_x \\ u_y & v_y \end{bmatrix} \tag{12}$$

In two equations case it is true that

$$J^T = Adj\ J \tag{13}$$

The preceding equation suggests a relationship in multivariable case.

Conjecture: Fractals exist if $J^T = Adj\ J$.

If this conjecture holds true in multidimensional cases, it should be possible to have a direct analogy to one-dimensional cases of Julia sets. However, this proves to be a very strict condition for problems with a general set of equations. It means that gradients of the functions that form the simultaneous equations should be orthogonal to each other to assure the existence of fractals. Such a condition is almost never met in practical problems. Therefore, a direct application of the theory of Julia sets appears to be limited to the one-dimensional iterating of analytic functions. However, if the conjecture could be proven, then it might be possible to devise an algorithm that transforms a general set of nonlinear equations to one that has mutually orthogonal gradients where all the results from the Julia sets theory will be applicable. It remains for future research to prove the above conjecture and therefore lead a way for an application of iteration theory to multidimensional problems.

6. CHAOTIC DESCENT

6.1 Theoretical Considerations

In order to demonstrate how the final goal can be achieved even though the condition given by the conjecture is not satisfied, roots will be sought using a two variable general problem. In other words, a system of two nonlinear equations for all the solutions $g_1(x_1,x_2)=0$ and $g_2(x_1,x_2)=0$ will be solved. In the previous section, it was shown that if g_1 and g_2 satisfy CR conditions one can simply form an analytic function $G(z)=g_1+g_2\,i$ and apply the NR method to this complex function. Also, one can choose to apply the NR multivariable method to g_1 and g_2. In both cases the result would be the same. We can then obtain fractal boundaries and, therefore, all the roots to the system of equations. It helps to visualize how this really happens, because in it lies the germ of applying the same process in the case that g_1 and g_2 do not satisfy CR conditions.

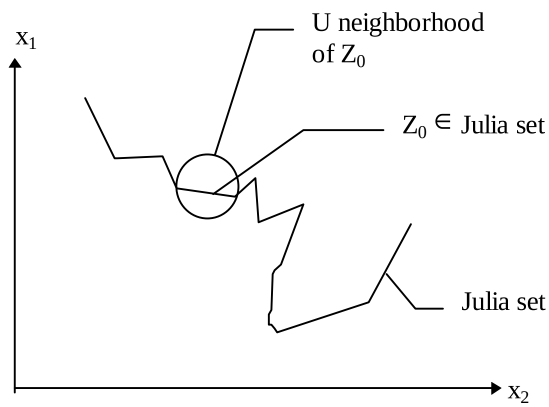

Figure 5. Julia set and the point that belongs to it.

Figure 5 shows some hypothetical basins of attractions. The procedure is to, through iteration, examine "every" point that belongs to ***U*** using the NR method. This would lead to all the roots because we would be iterating an analytic function. This is perhaps irrelevant now, but it will prove useful later in deciding what order we could examine the points in ***U***. We could cover ***U*** by a) circling around z_0, or b) examining along lines parallel to x_1 or x_2 axis, or c) examining along lines from z_0 to the boundary of ***U***. Any of these strategies will cover completely the whole of ***U,*** and we will have all of the initial points to lead us to the roots. It should be pointed out that if the direct extension to multivariable cases existed, we would be similarly examining every point in some small multidimensional volume ***U*** around a "gate" to all the roots using any of the above-mentioned methods.

Now, let us assume that g_1 and g_2 do not satisfy CR conditions and that it is required to solve g_1=0 and g_2=0. We could do the following: we could transform the problem into an optimization problem, forming an objective function $G(x_1,x_2) = g_1^2+g_2^2$. It should be obvious that solving the previous problem is equivalent to minimizing G and vice versa. We also know that only global minima with value of $G = 0$ are the solutions we are interested in and that there is probably more than one global minimum. This fact usually complicates searching when using the classical optimization methods.

Let us now imagine G in a three dimensional space x_1, x_2, and z where $z = G(x_1,x_2)$. G certainly consists of hills and valleys, some of which are our global minima. A contour plot of G in x_1 and x_2 plane is shown in Figure 6.

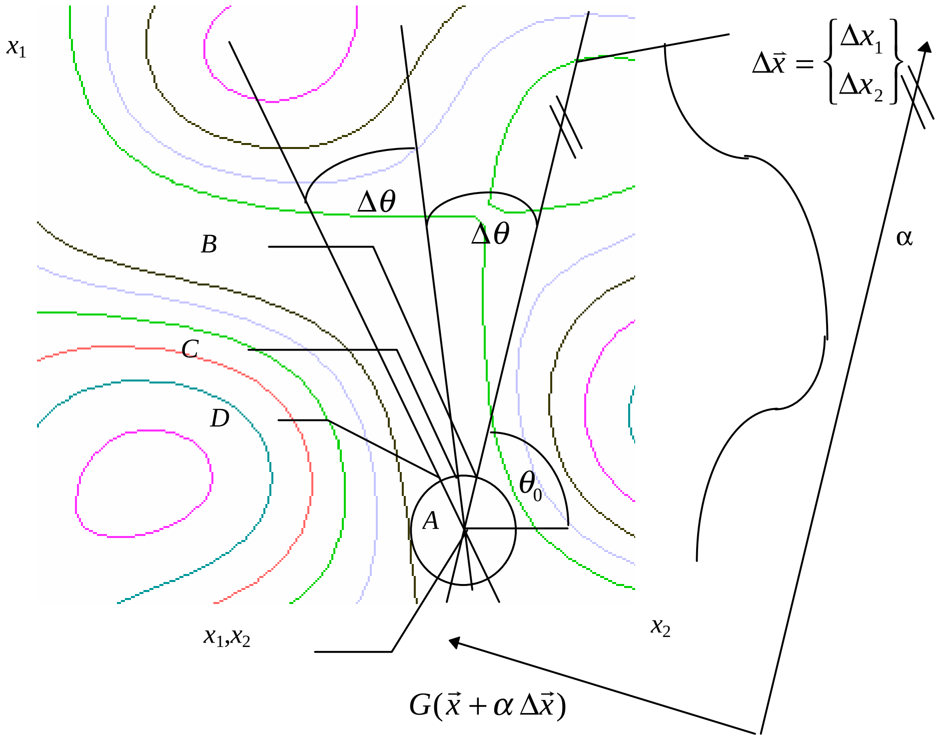

Figure 6. One-dimensional search through the variable space.

The curved boundaries in Figure 6 represent constant values of G. Let us now pick any point in the $x\ y$ plane and draw a circle around it. A point B is selected on that circle. A and B are connected to form a direction in the $x\ y$ plane. This direction now serves to represent a cut through some of the hills and valleys of $G(x_1,x_2)$. This cut G is a one-dimensional function that depends on α. Here, our knowledge from the previous section becomes handy. By taking the derivative of $G'_\alpha(\vec{x} + \alpha\Delta\vec{x})$ and solving $G'_\alpha(\alpha)$=0 for all the roots (α_1,α_2,etc.) we will be applying our knowledge of Julia sets. Note that we are able to locate all α_i, but we are only interested in real ones and those that represent a minimum of G. By checking the second derivative at real α_i, we memorize only those that satisfy G less than a small number to avoid local minima. We again continue in the same fashion by drawing another cut through D, offset by $\Delta\theta$ from C, and we repeat everything. Now, we know that when $\Delta\theta$ goes to 0 and we cover half of the circle from θ_0 to $\theta_0 + \pi$, and we also know that we actually covered the whole $x_1\ x_2$ plane by our one dimension cuts. Whatever we memorized with condition G less than some small number represents the global minima. Essentially, we were locating sensitive fractal areas along the line and using these areas to locate α_i. Now recall method c) from the previous section. Provided that we found Julia points, we would also examine every point along a line in ***U***. In that case, the region is fractal and immediately we iterate to a root. In the method just described we have to find a fractal region along the line and then examine every point in its vicinity. This shows the similarity between the two searches. However, the beauty of this simple procedure is that it could be directly applied to a multivariable problem whereas the previous direct procedure is limited to a strict Jacobian condition (Equation (13)). In summary, the first procedure could only be done when CR conditions are satisfied. For g_1 and g_2, we could form $G=g_1+g_2\, i$ and find any Julia point and encircle it by ***U*** to find all the roots. However, the procedure we have been developing can be used in any state of CR conditions. We form $G=g_1^2+g_2^2$, pick any point in x_1 x_2, find a Julia point along every direction and sample around it to find all the stationary points along that direction. In a multivariable problem case one would form $G=g_1^2+g_2^2+\ldots+g_i^2+\ldots$. and proceed with optimization by taking one-dimensional searches throughout the hyperspace as described previously.

6.2 Practical Considerations

The first objection one might have is that it is impossible to cut G with a line through all the space because there are an infinite number of directions, and one can never be sure if there is some direction that contains global minimum that we did not cover. This practical problem is actually easy to solve. Let us go back to the contour plot of G depicted in Figure 7.

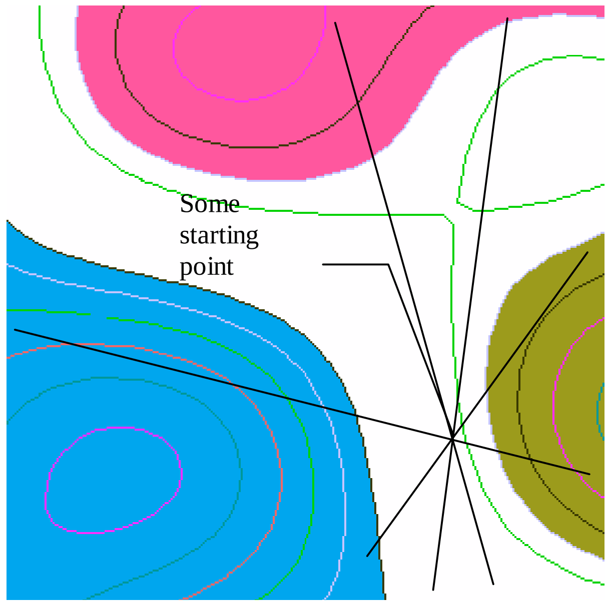

Figure 7. Cutting the basins of attraction formed by CG method.

Now, let us imagine that we are trying to solve the problem with a classical method, the conjugate gradient (CG) method for example. We know that by choosing some point close to a minimum, the CG will converge to that minimum. Assigning a color to all the points that lead to that minimum would generate a basin of attraction. It is also known that there are basins of attraction for other minima using CG to which we could assign different shades of gray, and that in starting our new procedure from some point to cut through G we will intersect some of the basins of attraction. Acknowledging this we can do the following. When we cut through the variable space and locate all the stationary minima along one direction, we can supply these locations to the CG method to converge to a local minimum. Now, with this possibility it should be clear that we do not have to have our $\Delta\theta$ small at all because basins of attraction formed by CG are generally large.

Nevertheless, the question remains - which $\Delta\theta$ do we choose? We can start with $\Delta\theta = 180^o$, and by cutting it in half and continuing halving the interval we will be getting to smaller $\Delta\theta$s. When to stop? Simply stop when no new minima are found. Besides, it is convenient to randomly choose a direction in the π segment. Random tries will uniformly distribute directions over the whole π segment. This is effective because one may find immediately a direction, which contains basins of global minima, just through luck.

When solving a usual optimization problem, there is often a unique global minimum. At such instance the following could be done. We start from a point and along a randomly chosen direction find a stationary point that leads to a best local minimum. Now, we can choose a new random direction from that local minimum and again find the best local minimum in the same fashion. It should be clear that eventually this process must stop at the global minimum. Due to its nature this process was termed Chaotic Descent.

7. NUMERICAL ALGORITHMS

Based on the discussion in previous sections, it is now possible to construct two algorithms. One lends itself to solving optimization problems; the other can be employed in solving a system of nonlinear equations.

7.1 Optimization Algorithm

The task is to minimize $G(\boldsymbol{x}):R^n \rightarrow R$ with any constraints where $\boldsymbol{x} \in R^n$.

1. Form a penalty function according to all the rules specified in the Optimization theory.
2. Choose a starting vector x^*.
3. Randomly choose a direction vector x_u.
4. Find all the minima (all αs) of one-dimensional function $G(x^* + \alpha\, x_u)$ using the Julia set theory (find a Julia point and use NR method to iterate from this point and solve for all the roots of $G'(x^* + \alpha\, x_u)=0$). If no minimum is found go to 3.
5. Use all the αs found in step 4 and form the starting points $x^* + \alpha\, x_u$ for CG method. Memorize the best local minimum and its location that CG finds.
6. If no new minimum is found after MAX steps, print the last minimum found and STOP
7. Take the location found in 5. as a new starting point x^* and go to 3.

7.2 Solving the System of Nonlinear Equations Algorithm

The task is to solve $g_i(\boldsymbol{x})=0$ $i=1\ldots n$ for all the solutions. $\boldsymbol{x} \in R^n$.

1. Form an objective function $G(\boldsymbol{x})=\Sigma\, g_i(\boldsymbol{x})^2$ which is to be minimized.

2. Choose a starting vector x^*.

3. Randomly choose a direction vector $x_{u.}$

4. Find all the minima (all αs) of one-dimensional function $G(x^*+ \alpha\, x_u)$ using the Julia set theory (find a Julia point and use NR method to iterate from this point and solve for all the roots of $G'(x^*+ \alpha\, x_u)=0$). If no minimum is found go to 3.

5. Use all the αs found in step 4 and form the starting points $x^*+ \alpha\, x_u$ for CG method. Memorize all the local minima and their locations that CG finds.

6. Those minima that are less than some very small number are the solutions to the system of equations, so keep them in memory; otherwise forget them.

7. If no new minimum is found after MAX steps, print all memorized minima and STOP.

8. Go to 3.

8. NUMERICAL EXPERIMENT AND ANALYSIS

In testing the Chaotic Descent method from the previous section, we need an optimization problem that possesses the characteristics which frequently make searching for global minima a difficult task. Such difficulties are a) numerous local minima within the feasible region, and b) irregular boundaries of the feasible region. These difficulties are more than sufficient to seriously affect the outcome of any existing global optimization procedure, and locating the global minimum is seriously jeopardized. Why would that be the case? Simply due to the usual nonlinearity of optimization problems, all of the optimization procedures are necessarily numerical and could only determine the local characteristics of the objective function. Therefore, to find the global behavior of the objective function, there is nothing better than to exhaustively search the whole variable space. These are the main ingredients of any numerical procedure including the Chaotic Descent. However, this procedure searches exhaustively in a very efficient manner.

The main problem for any global optimization method is uncertainty of locations and boundaries of a feasible region, which means that it is completely unknown where to start the exhaustive search and how fine a distribution of initial points to generate. These kinds of problems are completely eliminated with the Chaotic Descent. The algorithm automatically locates the best regions that provide the possibility of locating all of the roots along one direction. Successive repetition of this procedure for as many directions as needed quickly locates the global minimum.

To illustrate a procedure an example of a practical value was selected. The task is to find a rectangle with the smallest area that encloses n circles. Similar problems arise in the packaging industry where the necessity of the least expenditure of material is a priority. A mathematical description of the problem depicted in Figure 8 follows.

Minimize (14)

$$F = x_c y_c$$

subject to

$$\begin{aligned} &x_i \geq r_i, y_i \geq r_i \\ &x_c - x_i \geq r_i, y_c - y_i \geq r_i \\ &(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2 \geq (r_i + r_{i+1})^2 \end{aligned}$$

To solve this optimization problem a penalty function approach is used when a constrained problem is transformed into an unconstrained one. The objective function obtained in this way possesses many local minima due to many possible arrangements of the circles within the rectangle as well as a few “equivalent” global minima. This should not escape one’s observation since the optimal configuration of the circles within the rectangle might be oriented in several ways in xy plane which gives the same value for the objective function.

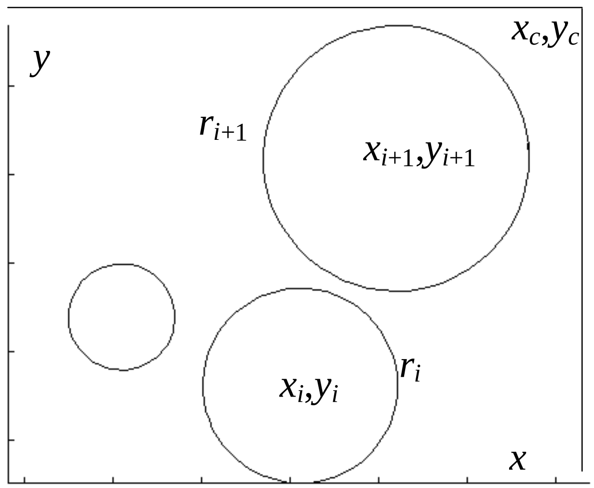

Figure 8. **Problem of finding a rectangle with the smallest area that encloses *n* circles.**

For numerical testing we chose 5 circles to be enclosed within a rectangle. This gives an optimization problem with 12 variables. Radii of the circles were chosen to be r_1=1.2, r_2=2.2, r_3=3, r_4=3.5, r_5=4. The best solution found by the Chaotic Descent method with the area of the rectangle 177.65 where the corner opposite of 0,0 is x_c=12.91, y_c=13.75 is depicted in Figure 9. The solution was confirmed with the Genetic Algorithms and it is indeed a global solution. Note that the solution shown in Figure 9 is the only one of 8 solutions with the same area of the rectangle for the mathematical model described with Equation (14) due to different orientation and folding in the plane, however, physically it is the only solution of the problem. Numerical calculations were done on PC 486 66MHz Windows 95 Turbo C++ 3.0 environment. The time needed for the Chaotic Descent to locate the solution was anything between 1 to 6 minutes, 10 to 70 directions searched through variable space and 300 to 1100 evaluations of local minima.

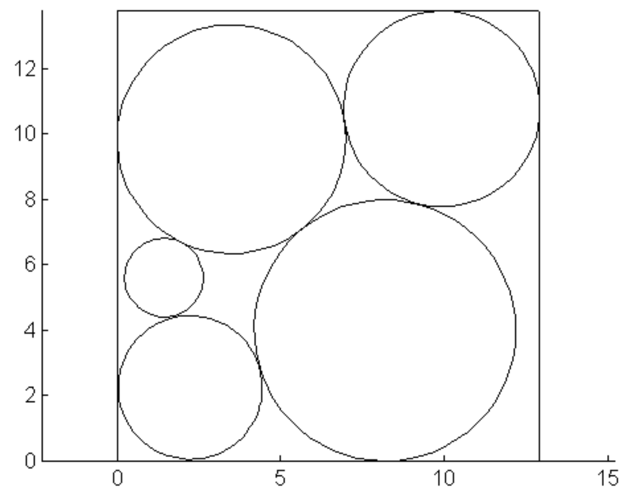

Figure 9. The best solution for the rectangle problem.

9. CONCLUSION

This paper presents a method that enhances the likelihood of locating global solutions in optimization problems and solving for all of the solutions for a set of nonlinear equations. The method incorporates a novel idea based on using the sensitivity of numerical computations to locate all of the roots of a nonlinear equation. Specifically, we first locate a point that belongs to the Julia set and then iterate from its neighborhood using the Newton-Raphson method to locate all of the solutions to the equation. To make this procedure useful for multivariable problems we span the variable space with one dimensional lines along which we try to locate better minima for the given objective function until we descent to a global one. The performance of the algorithm was observed on an optimization problem where the global solution was known in advance. To determine the value of this procedure a comparison of this approach to established algorithms is needed. This will be the focus of future reports. Also, a resolution of the conjecture presented would be the most useful to determine a possibility of solving sets of nonlinear equations for all the solutions without transforming the problem to an optimization one.

10. REFERENCES

1. Fatou M.P. Sur les equations fonctionelles. Bull. Soc. Math. France 1919; **47**: 161-271.

2. Cayley A. The Newton-Fourier Imaginary Problem. American Journal of Mathematics 1879

3. Conway JB. *Functions of One Complex Variable*. Springer-Verlag: New York, 1978.

4. Devaney RL. *An Introduction to Chaotic Dynamical System.* Perseus Publishing Co., a division of Harper/Colins: 1989.

5. Jacoby SL, et al. *Iterative methods for Nonlinear Optimization Problems*. Prentice-Hall: Englewood Cliffs, New Jersey, 1972.

6. Jovanovic V, *Identifying, Utilizing and Improving Chaotic Numerical Instabilities in Computational Kinematics.* Ph.D. Thesis; The University of Connecticut, 1998.

7. Jovanovic VT, Kazerounian K. Using Chaos to Obtain Global Solutions in Computational Kinematics. *ASME Journal of Mechanical Design* 1998; **120**(2): 299-304.

8. Jovanovic VT, Kazerounian K. On the Metrics and Coordinate-System-Induced Sensitivity. *International Journal of Numerical Methods in Engineering* 1998; **42**(4): 729-747.

9. Julia G. Memoire sur l'iteration des fonctions rationelles. *J. Math. Pures Appl.*1918; **8**: 47-245.

10. Peitgen H, et. al. *Chaos and Fractals: New Frontiers of Science.* Springer-Verlag: New York. 1992

11. Wilde DJ. *Globally optimal Design*. Wiley: New York, 1978.