MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 1402 December 1992

A Global Approach to Parameter Estimation of Chaotic Dynamical Systems

Athanassios G. Siapas[1]

Abstract

We present a novel approach to parameter estimation of systems with complicated dynamics, as well as evidence for the existence of a universal power law that enables us to quantify the dependence of global geometry on small changes in the parameters of the system. This power law gives rise to what seems to be a new dynamical system invariant.

Copyright © Massachusetts Institute of Technology, 1992

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-89-J-3202.

[1]Artificial Intelligence Laboratory, and Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

1 Introduction

> *"...Since all our measurements and observations are nothing more than approximations to the truth, the same must be true of all calculations resting upon them, and the highest aim of all computations made concerning concrete phenomena must be to approximate, as nearly as practicable, to the truth."*
>
> *K. F. Gauss, "Theoria Motus Corporum Coelestium" (1809).*

Estimation theory deals with the problem of estimating the state of a stochastic dynamical system from noisy observations. The earliest stimulus for its development was apparently provided by astronomical studies of planet and comet motion in the 18^{th} century. The motion of these bodies can be completely characterized by a finite number of parameters and the estimation problem that was considered was that of inferring the values of these parameters from telescopic measurement data.

To be more precise, suppose m measurement vector quantities $y_1, \ldots, y_m \in \mathbb{R}^l$ are available at discrete instants of time $t_1, \ldots, t_m$. The parameter vector $x \in \mathbb{R}^n$ which we wish to determine is assumed to be linearly related to the measured data, i.e.

$$y_k = M_k x + v_k \tag{1}$$

where v_k represent the measurement errors that occur at each observation time. Let $\hat{x}_m$ denote the the estimate of x based on the m data samples $\{y_1, \ldots, y_m\}$. Then the *residual* r_k $(1 \leq k \leq m)$ associated with the k^{th} measurement is defined to be the difference between the observed value y_k and the value predicted from the estimate $\hat{x}_m$:

$$r_k = y_k - M_k \hat{x}_m \tag{2}$$

Around 1795 Karl Friedrich Gauss invented the revolutionary *method of least-squares* for attacking the above problem. The method was independently invented and published by Legendre in 1806 in his book *"Nouvelles méthodes pour la determination des orbites des comètes"*. A detailed description of the method was published by Gauss in 1809 in his book *"Theoria Motus Corporum Coelestium"*. The name "least-squares method" comes from the fact that the optimal estimate $\hat{x}_m$ for x based on the observations $Y_m = \{y_i\}_{1 \leq i \leq m}$ is the value of x which minimizes an appropriately weighted sum of the squares of the residuals

$$L_m = \sum_{k=1}^{m} r_k^T W_k r_k \tag{3}$$

where the elements of the matrices W_k are selected to indicate the degree of confidence that one can place on the individual measurements.

We notice that the state/parameter vector x is assumed to be fixed in the above description. Moreover, the least-squares approach has no probabilistic meaning. [1]

[1] It should be noted that Gauss also considered a probabilistic approach to the estimation problem, but rejected in favor of minimizing function 3.

The general shift of attention to dynamical systems in the 20^{th} century, as well as the introduction of the *maximum likelihood method* by R. A. Fisher in 1912 led to new developments in the field of estimation theory. Kolmogorov in 1941 and Wiener in 1942 independently developed a linear minimum mean square estimation technique that received considerable attention and laid the foundations for the development of Kalman filter theory. This approach allowed for systems with state changing with time, as well as both continuous and discrete observations. Kolmogorov's and Wiener's work focuses on the analysis and synthesis of systems in terms of their input-output characteristics, reflecting the general trend in the scientific community at that time. The problems were formulated in terms of integral equations and the main tools used were the Laplace and Fourier transforms.

Subsequent scientific developments have stressed the *state space* approach, which uses difference and differential rather than integral equations for describing a system. Although both these approaches are mathematically equivalent the latter proved much more convenient and useful, opening the door to many new developments.

The state space reformulation of the estimation problem suggested a recursive approach, first attempted for a specific system in 1955, by J. W. Follin at Johns Hopkins University. Five years later, R. E. Kalman published a very influential paper ([20]), in which he described an optimal recursive algorithm for solving the linear estimation problem using a general state space approach. This became known as the *Kalman filter*. In a sense, the Kalman filter is nothing but an efficient computational solution of Gauss's least-squares problem in a more general state space setting.

The value of Kalman's reformulation was that it led to significant new insights and had the effect of unifying all previous results. Moreover, the Kalman filter equations provided an extremely convenient procedure for digital computer implementation, and its recursive structure, together with the fact that the Kalman gain is independent of the observations and can be precomputed, opened the door for real time estimation.

Nature is inherently nonlinear. Hence, in order to apply estimation techniques to real physical systems, it was necessary to develop algorithms that can deal with nonlinear dynamical systems. There are many reasons why nonlinear filtering theory is much harder than linear theory. The main difference is that the linear filter has essentially a finite description: the *state* of the system consists of the mean and the covariance matrix, which are enough to completely determine the conditional probability density that contains all relevant information. In the nonlinear case, the state of the filter is *infinite*, consisting of the whole conditional density function used to compute oprimal estimates. The numerical problems involved in computing the whole probability density function are, in general, intractable, since they involve the solution of complicated integro-differential equations or functional integral equations.

In practice, finite approximations to the conditional probability density are considered. An approximate nonlinear filter is obtained by parametrizing the conditional

density via a finite set of parameters, and computing equations for the evolution of these parameters, which comprise the state of the system. One of the most popular approaches used in practice, is to linearize the system in question and apply linear filtering theory. This gives rise to the Extended Kalman Filter and its variants. The main characteristic of all these approaches is that they try to follow *locally* a nominal trajectory, keeping track of how the state (finite approximation to an infinite conditional probability density) changes. This local character imposes serious limits to the applicability and effectiveness of these algorithms.

We notice that the traditional estimation algorithms are in effect attempts to extend inherently linear ideas to the nonlinear setting. One can completely understand linear systems by looking at isolated integral curves, but this is hopeless for the nonlinear case, because of the immense complexity of nonlinear evolution.

Henri Poincaré was the first to realize that the attention should be shifted from an analysis of isolated trajectories and local characterizations, to a more global topological and geometric understanding of the phase space of nonlinear dynamical systems[2]. At the present time we are witnessing a spectacular blossoming of nonlinear dynamics, made possible on the one hand by great theoretical developments on global topological and geometrical analysis, initiated by Poincaré's revolutionary work in the 19^{th} century, and on the other hand by the wide availability of increasingly powerful computers.

We believe that very interesting new insights and extremely accurate novel algorithms can be obtained by attacking parameter estimation problems using a global geometrical point of view. Moving up one level of abstraction we wish to consider parameter estimation algorithms whose primitive objects are geometric structures in phase space (represented as points in an appropriate function space), rather than points on isolated trajectories.

We demonstrate how to exploit the complexity of global geometrical phase space structures of nonlinear dynamical systems and their dependence on parameter variations in order to obtain extremely accurate parameter estimation algorithms that do not depend on local approximations, in the context of complex analytic dynamics.

In particular, the global algorithm was tested on the family of quadratic maps of the Riemann sphere and rational maps obtained from Newton's method on complex cubic polynomials. We show how to transform the estimation problem into a problem of minimizing a dissimilarity measure between images containing global dynamical information. Our experiments indicate that the dissimilarity function to be minimized is locally unimodal. In fact it seems to obey an exact power law locally. The exponent appears to be a new invariant of these dynamical systems, which we call the *parameter sensitivity exponent*, and which characterizes the dependence of global geometric structures on parameter variations.

The global algorithm gives extremely accurate estimates for the parameters of these systems systems (improving the initial estimate by more than 13 orders of magnitude in a certain case). Moreover, appears to be very robust with respect both to observation noise and dynamical noise.

2 Parameter Estimation and Global Geometry

The study of how geometric structures in phase space change as system parameters vary is of great interest and has received much attention. The main focus so far has been the study of changes in the *topology* of phase space structures (bifurcation theory).

Our objective is to exploit the way *exact geometrical* rather than topological features of phase space structures change as system parameters vary slightly, in order to obtain novel extremely accurate parameter estimation algorithms that do not depend on local approximations.

This approach led us to the discovery of what seems to be a very general power law that enables us to quantify the dependence of global geometry on small changes in the parameters of the system.

In order to demonstrate our approach we will restrict our attention to how basins of attraction change as parameters vary, and show how to transform the parameter estimation problem into an optimization problem over an appropriate space of functions containing global dynamical information.

Our discussion in this paper will be restricted to estimating a single parameter of a system, but our techniques can be readily generalized to higher dimensional problems.

We begin by demonstrating this approach for the family of complex quadratic polynomials.

2.1 The Quadratic Family

Given any complex quadratic polynomial $p(z) = az^2 + 2bz + d$, let $M(z) = az + b$ and $c = ad + b - b^2$. If $f_c : \overline{\mathbb{C}} \to \overline{\mathbb{C}}$ denotes the map $f_c(z) = z^2 + c$, where $\overline{\mathbb{C}}$ is the Riemann sphere, then:

$$
\begin{aligned}
(4)\quad M^{-1} \circ f_c \circ M(z) &= M^{-1}((az+b)^2 + c) \\
&= M^{-1}(a^2z^2 + 2abz + b^2 + c) \\
&= \frac{(a^2z^2 + 2abz + b^2 + c) - b}{a} \\
&= p(z)
\end{aligned}
$$

i.e. p and f_c are (analytically) conjugate. It follows that in order to understand the dynamics of all complex quadratic polynomials, it is enough to understand the dynamics of the complex one-parameter family

$$\mathcal{Q} = \{f_c : \overline{\mathbb{C}} \to \overline{\mathbb{C}}, c \in \mathbb{C}, f_c(z) = z^2 + c\}$$

Both the variable z and the parameter c fill out a complex plane. We will refer to the z-plane as the *dynamical plane* and to the c-plane as the *parameter plane*.

[2] It is interesting to note that, just like Gauss' idea in the case of estimation theory, the motivation for Poincaré's development of global geometrical dynamics comes from the study of the motion of celestial bodies.

2.1.1 Julia Sets

Given a rational map $f : \overline{\mathbb{C}} \to \overline{\mathbb{C}}$ of the Riemann sphere $\overline{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$, we can get a dynamical system by repeated application of f. In the begining of the twentieth century, the French mathematicians P. Fatou and G. Julia studied the iteration of complex polynomials of degree $d \geq 2$. Having at their disposal a powerful theorem of Montel that gave a sufficient condition for the normality of a family of meromorphic functions, they realized that it is very interesting to consider the following decomposition of the dynamical plane:

Definition 2.1 *A point $z \in \overline{\mathbb{C}}$ is an element of the Fatou set, F_f of a rational mapping f, if there exists a neighborhood U of z, such that the family of iterates $\{f^n\}$ is a normal family on U. The Julia set J_f of f is the complement of the Fatou set.*

where, f^n denotes n-fold functional composition of f by itself.

Without recalling the exact definitions, let us only remark that normal families have values that do not diverge under iteration. So, in some sense the Fatou and Julia sets of f are the sets of *stable*, *unstable* points of $\overline{\mathbb{C}}$ with respect to f, respectively.

We notice that the point at infinity ∞ is always an attracting fixed point for quadratic maps[3]. Let

$$A_c(\infty) = \{z \in \mathbb{C} : f_c^n \to \infty \ \ as \ \ n \to \infty\}$$

be the basin of attraction of infinity. We have the following result:

Proposition 2.1 *The Julia set J_{f_c} of $f_c : \overline{\mathbb{C}} \to \overline{\mathbb{C}}$, $f_c(z) = z^2 + c$, is equal to $\partial A_c(\infty)$.*

The proof of this statement is a direct consequence of the fact that the boundary of any completely invariant component (here $A_c(\infty)$) of the complement of the Julia set has to equal the Julia set.

The set $K_{f_c} = \mathbb{C} - A_c(\infty)$ is called the *filled in Julia set*. We clearly have $\partial K_c = J_{f_c} = \partial A_c(\infty)$, i.e. J_{f_c} separates competition between orbits being attracted to ∞ and orbits remaining bounded for all time.

Example 1: Consider the map $f_0 : \overline{\mathbb{C}} \to \overline{\mathbb{C}}$, $f_0(z) = z^2$. Clearly any point outside the unit circle S^1 has an orbit that escapes to infinity. Moreover, any point inside the unit circle has an orbit converging to 0. Consequently, the Julia set J_{f_0} of f_0 is the unit circle S^1.

Example 2: The Julia set of the map $f_{-2} : \overline{\mathbb{C}} \to \overline{\mathbb{C}}$, $f_{-2}(z) = z^2 - 2$ is the interval $[-2, 2]$ on the real line[4].

In general, Julia sets are not smooth, but very complicated fractal objects, exhibiting an amazing variety of geometric forms as the parameter c changes. Figure 1 (taken from [26]) shows examples of Julia sets of complex quadratic polynomials corresponding to various points c on the complex plane, along with the Mandelbrot set which controls their topological structure.

[3] In fact ∞ is an attracting fixed point for all complex polynomial maps on $\overline{\mathbb{C}}$

[4] The proof is not obvious. See [6].

2.1.2 The Mandelbrot Set

In 1905, P. Fatou proved the following very surprising result[5]:

Theorem 2.1 *Every attracting cycle for a polynomial or rational function attracts at least one critical point.*

Each quadratic polynomial f_c in $\mathcal{Q}$ has a unique critical point, namely $z_0 = 0$. The corresponding critical value is $f_c(z_0) = f_c(0) = c$. It follows that f_c can have at most one attracting cycle in the complex plane. More generally, a polynomial of degree $d \geq 2$ can have at most $d - 1$ attracting cycles.

In 1918-1919 P. Fatou and G. Julia proved another result which further supported their conjecture that the dynamical behavior is dominated by the behavior of critical points:

Theorem 2.2 *Let Ω_f denote the set of critical points for a polynomial $f : \overline{\mathbb{C}} \to \overline{\mathbb{C}}$, and let K_f be the set of all points in $\mathbb{C}$ which do not escape to infinity under f, i.e. $K_f = \overline{\mathbb{C}} - A(\infty)$. Then:*

1. *$\Omega_f \subset K_f \Leftrightarrow J_f$ is connected.*
2. *$\Omega_f \cap K_f = \emptyset \Rightarrow J_f$ is a Cantor Set.*

Since for a quadratic map f_c, there exists only one critical point namely $z_0 = 0$, an immediate corollary of theorem 2.2 is the following:

Corollary 2.1 *The Julia set J_{f_c} of the quadratic map f_c is either connected or a Cantor set. Moreover, J_{f_c} is connected if and only if $f_c^n(0)$ does not tend to ∞ as $n \to \infty$.*

The above corollary suggests a natural decomposition of the parameter plane into the Mandelbrot set

$$M = \{c \in \mathbb{C} : J_{f_c} \ \ is \ \ connected\} \tag{5}$$

and its complement $\mathbb{C} - M$. Moreover, corollary 2.1 gives us a way to compute the Mandelbrot set: in order to check whether a point c of the parameter plane is in M, it is enough to check whether the orbit of 0 under f_c does not tend to infinity.

We remark that sets similar to the Mandelbrot set occur in many other families of complex analytic maps. This happens since many mappings or their iterates locally behave like a quadratic polynomial. Hence the Mandelbrot set is in some sense a universal object.

The boundary ∂M of the Mandelbrot set is a *bifurcation set*, i.e. the topological nature of the Julia set changes as we cross this set in the parameter plane. In the next sections we will investigate how the geometry rather that the topology of Julia sets depends on parameter variations.

[5] The proof can be found in [9].

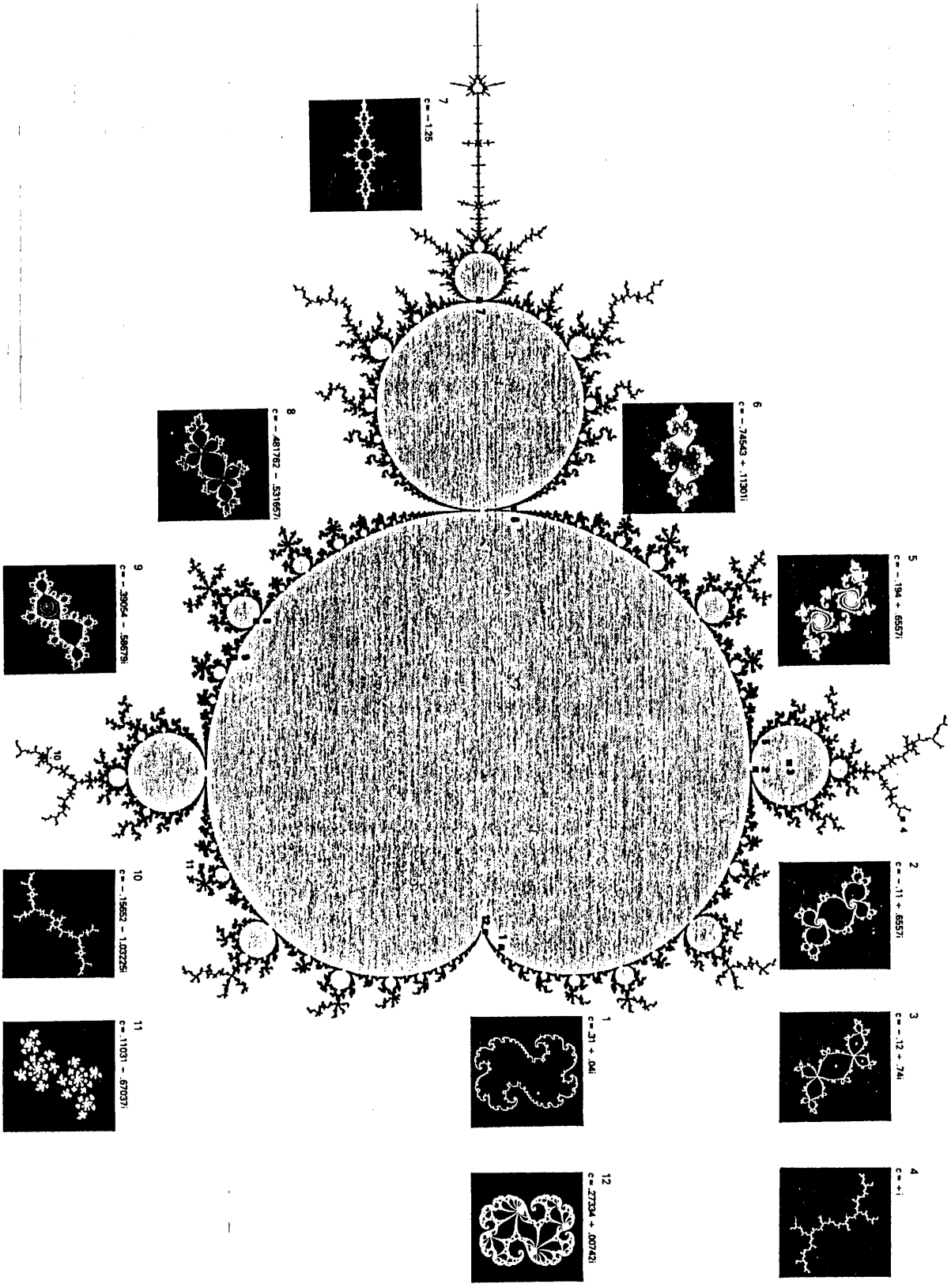


Figure 1: Julia sets of complex quadratic polynomials surrounding the Mandelbrot Set which controls their structure. The picture is taken from [26].

2.2 Global Parameter Estimation for the Quadratic Family

The complex one-parameter family of dynamical systems

$$z_{n+1} = z_n^2 + c \tag{6}$$

can be considered as the following real two-parameter family

$$x_{n+1} = x_n^2 - y_n^2 + \lambda \tag{7}$$

$$y_{n+1} = 2x_n y_n + \xi \tag{8}$$

under the usual identification of $\mathbb{R}^2$ with $\mathbb{C}$ sending z_n to $x_n + y_n i$ and c to $\lambda + \xi i$. We want to consider the problem of estimating λ, when ξ is held constant, in the presence of observation and dynamic noise. In particular, suppose the real system has noisy dynamical evolution:

$$x_{n+1} = x_n^2 - y_n^2 + \lambda + v_{d,x}(n) \tag{9}$$

$$y_{n+1} = 2x_n y_n + \xi + v_{d,y}(n) \tag{10}$$

and we actually observe:

$$\overline{x}_n = x_n + v_{o,x}(n) \tag{11}$$

$$\overline{y}_n = y_n + v_{o,y}(n) \tag{12}$$

where
both that the dynamical $\{v_{d,x}(n)\}_{n\in\mathbb{Z}_+}$, $\{v_{d,y}(n)\}_{n\in\mathbb{Z}_+}$ as well as the observation $\{v_{o,x}(n)\}_{n\in\mathbb{Z}_+}$, $\{v_{o,y}(n)\}_{n\in\mathbb{Z}_+}$ noise sequences are white Gaussian random sequences.

2.2.1 Setting up global functions for quadratic maps

The primitive objects used in local methods are points on isolated trajectories. In the next section, we will show how to obtain extremely accurate estimates of the parameter λ by moving up one level of abstraction and consider an algorithm that uses representations of the Julia sets of the maps as primitive objects. In this section, we describe how to use proposition 2.1 in order to obtain a discrete representation of Julia sets of quadratic maps $f_c : \overline{\mathbb{C}} \to \overline{\mathbb{C}}$, $f_c(z) = z^2 + c$. The method we will describe is often refered to as the Level Set Method (LSM) ([26], [27]).

We restrict our attention to a domain $D \subset \mathbb{C} \approx \mathbb{R} \times \mathbb{R}$, on which we impose a grid of $n \times m$ cells. Choose a large integer N_{max} (iteration resolution) and an arbitrary set T (target set) containing ∞. We will take $T = \{z : \|z\| \geq R\}$, where R is a large number[6]. For each cell in the decomposition of the domain D assign an integer label $l_c(p) = l_c(p, T)$, where p is the centerpoint of the cell, in the following way:

$$l_c(p) = \begin{cases} k & \text{provided } f_c^i(p) \notin T \text{ and } f_c^k(p) \in T, \\ & \text{for } 0 \leq i < k \text{ and } k \leq N_{max} \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

If $l_c(p)$ is nonzero then p escapes to infinity and $l_c(p)$ is the *escape time* (measured in the number of iterations) needed to hit the target set T around ∞. The contours obtained by the above algorithm are approximations of the equipotential curves of the filled in Julia set K_c when J_c is connected (see for example [27]).

As c moves on the parameter plane where the Mandelbrot set lives, the corresponding Julia set changes from shape to shape producing an immense variety of possible geometric forms (figure 1)

Let us fix ξ to the value $\xi = 0.3$, and consider how the geometry of the Julia sets changes as λ varies locally around the value $\lambda = -1$. Figures 2, 3, 4 show a window of the Julia set for $\lambda = -1.0, \lambda = -1.00001, \lambda = -1.0001$ respectively. We see that the human eye can clearly sense changes of the order 10^{-4} or so and tell which phase image from 3 and 4 is closer to 2. We expect that by comparing these images we can sense very small changes in parameters and obtain extremely accurate estimation algorithms. In the next section we describe how to turn the above intuitive approach of comparing images to get an estimate of a parameter into a precise algorithm.

2.2.2 The Global Approach

In order to be able to store in the computer and manipulate the images that we get by running LSM on a domain $D \subset \mathbb{R} \times \mathbb{R}$, we chose to represent them as two dimensional arrays $A = A_\lambda(D) = (\alpha_{i,j}^\lambda)_{1\leq i\leq n, 1\leq j\leq m}$.

The array $A = A_\lambda(D)$ is a lookup table (a discrete representation) of a function:

$$F_\lambda : D \to \mathbb{R}$$

which we will refer to as the *global function* for f_c on D.

Before we proceed, let us recall that if (X, μ) is a measure space, for $p \geq 1$, $L^p = L^p(X, \mu) = \{f : X \to \mathbb{R}$ measurable such that $\int_X \|f\| d\mu < \infty\}$. It is a standard result of functional analysis that $L^p(X, \mu)$ is a vector space and $\|.\|_p$ is a norm on $L^p(X, \mu)$. For $f \in L^p$ the value:

$$\|f\|_p = \left(\int_X \|f\|^p d\mu \right)^{1/p} \tag{14}$$

is called the L^p-norm of f. Let us define the L^p-distance between two functions $f, g \in L^p$ to be:

$$d_p(f, g) = \|f - g\|_p \tag{15}$$

Let us choose the domain D to be a compact rectangle in $\mathbb{R}^2$. Then the function $F_\lambda \in L^p$, where $X = D$ and μ is the Lebesgue measure on $D \subset \mathbb{R}^2$. The distance d_p gives us a measure of how different F_λ, F_μ are for λ, μ two different values of the parameter. If

$$A_\lambda = (\alpha_{i,j}^\lambda)_{1\leq i\leq n, 1\leq j\leq m}$$

$$A_\mu = (\alpha_{i,j}^\mu)_{1\leq i\leq n, 1\leq j\leq m}$$

are the discrete representations of F_λ, F_μ respectively, a natural measure of their difference, is the discrete L^p-distance ($p < \infty$):

$$d_p(A_\lambda, A_\mu) = \|A_\lambda - A_\mu\|_p = \left[\sum_{i=1}^{n} \sum_{j=1}^{m} \|\alpha_{i,j}^\lambda - \alpha_{i,j}^\mu\|^p \right]^{1/p} \tag{16}$$

From now on we will tend to use the same notation for both the continuous quantities and their discrete representations.

[6] In most of our experiments we take $R = 100$.

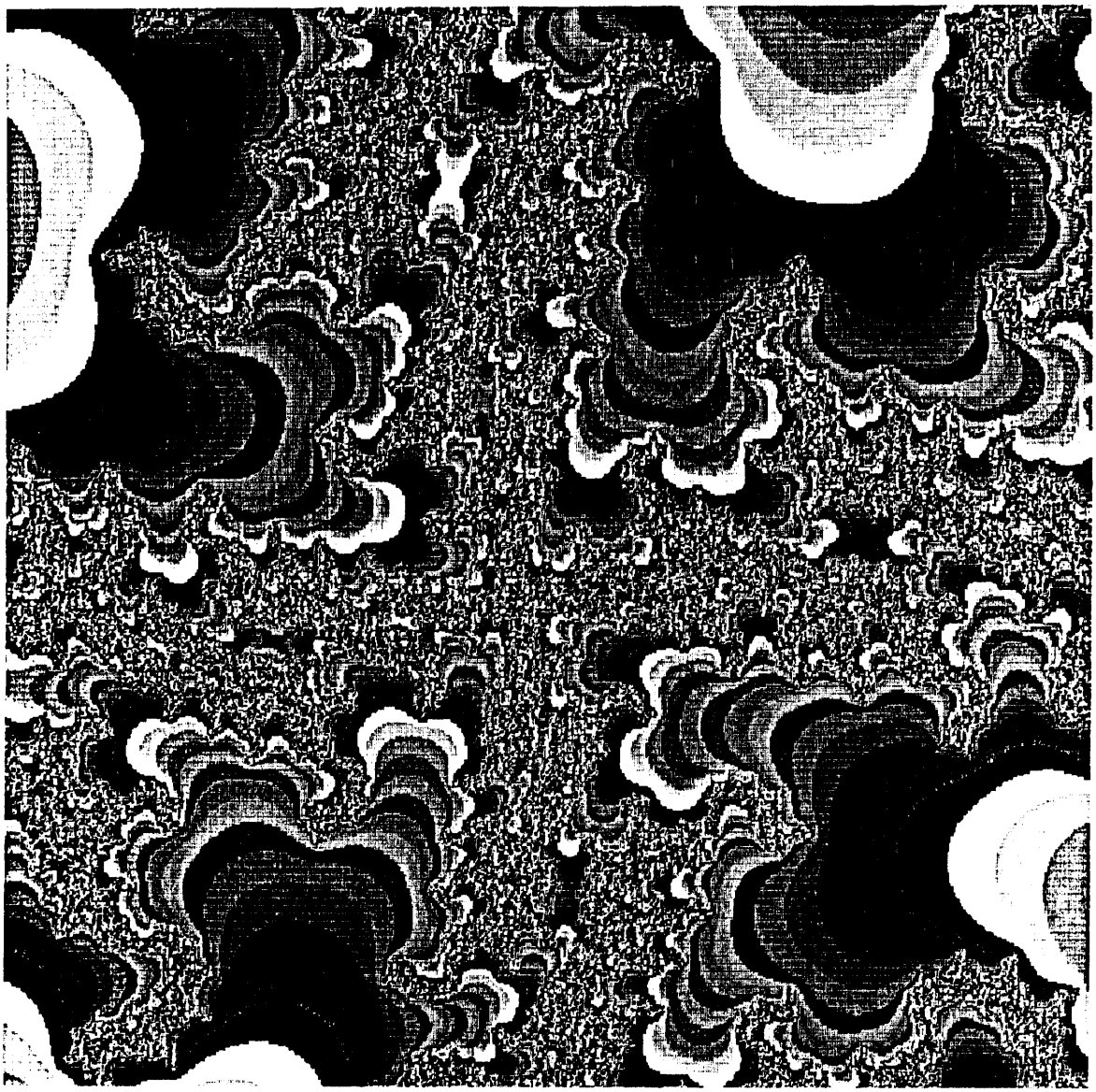


Figure 2: Julia Set for $\lambda = -1.0, \xi = 0.3$, $D = [-0.079555, -0.079525] \times [0.265320, 0.265350]$.

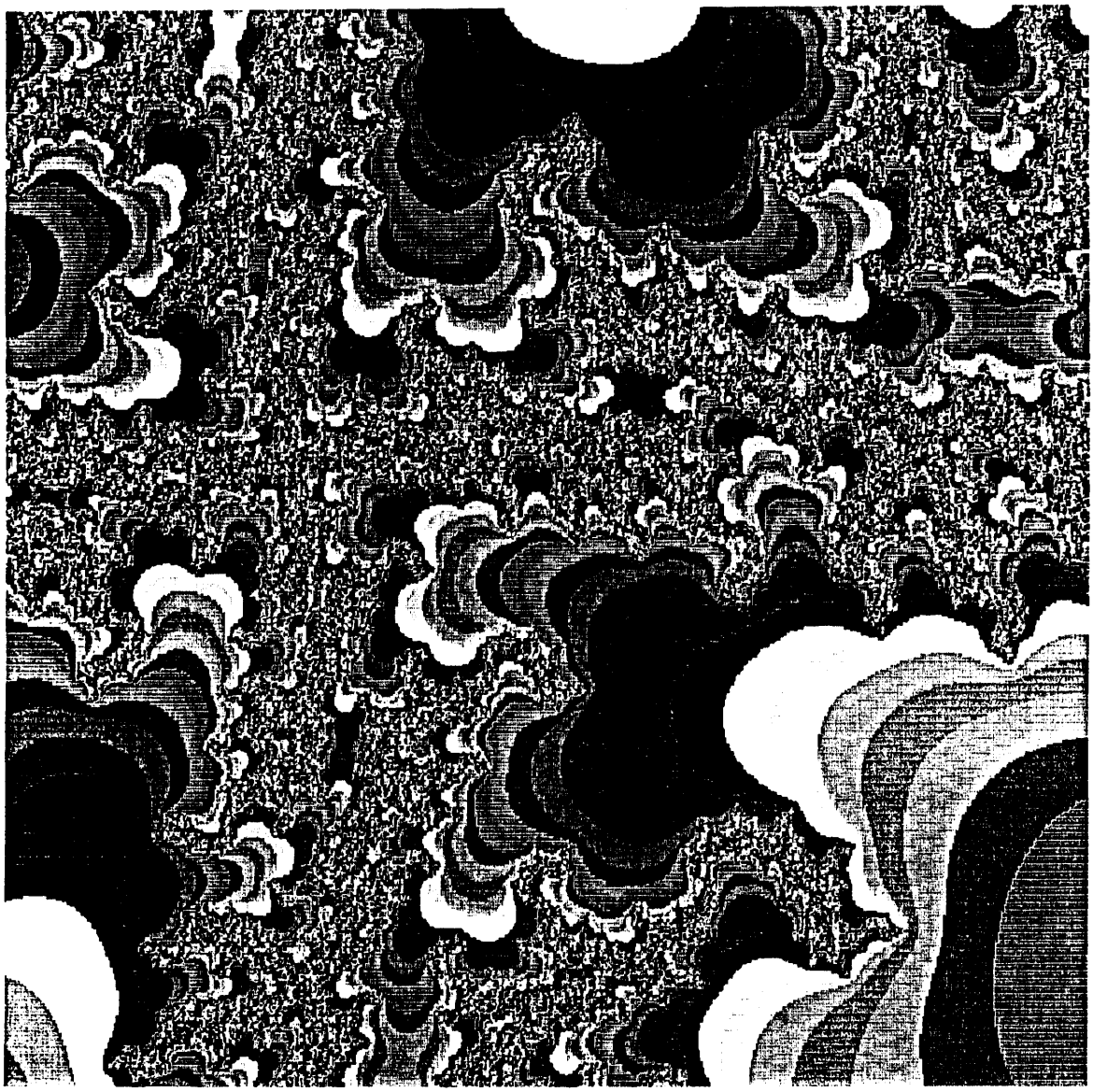


Figure 3: Julia Set for $\lambda = -1.00001, \xi = 0.3$, $D = [-0.079555, -0.079525] \times [0.265320, 0.265350]$.

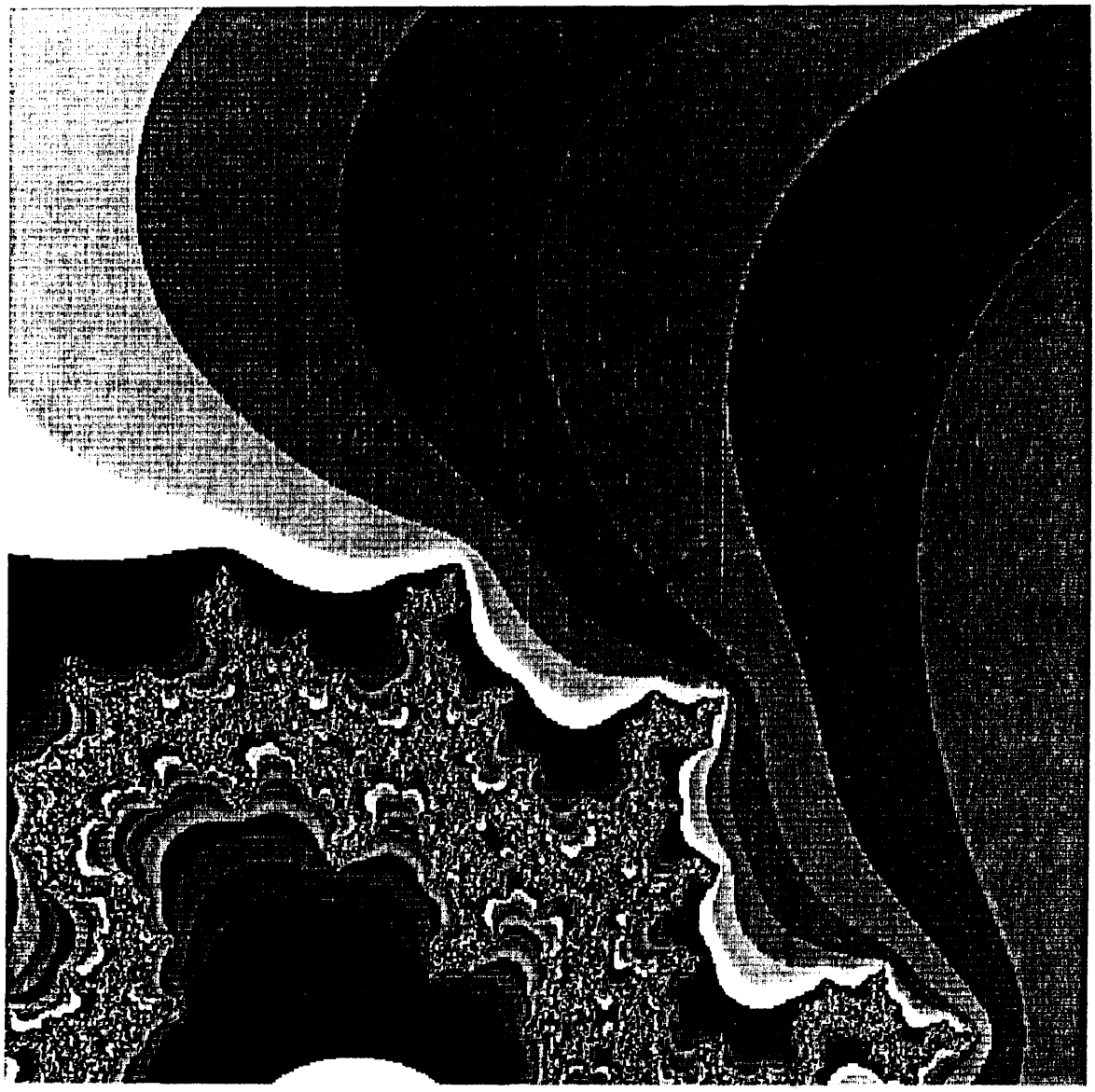


Figure 4: Julia Set for $\lambda = -1.0001, \xi = 0.3$, $D = [-0.079555, -0.079525] \times [0.265320, 0.265350]$.

It would seem that the L^p distance of global functions F_λ, F_μ is a rather coarse characterization of the differences of the corresponding images. It turns out that it is ideally suited for doing parameter estimation in the presence of noise. The reason is that, intuitively, taking the L^p distance of global functions has the effect of *averaging away the effects of noise, returning a measure of the difference between the images which is rather noise insensitive.*

Given a parameter value λ, let us define the following functional:

$$\psi_\lambda : L^p \to \mathbb{R} \tag{17}$$

$$\psi_\lambda(G) = d_p(F_\lambda, G) \tag{18}$$

Let us now define the *dissimilarity function* for parameter λ to be the function:

$$\varphi_\lambda : \mathbb{R} \to \mathbb{R} \tag{19}$$

$$\varphi_\lambda(\mu) = \psi_\lambda(F_\mu) = d_p(F_\lambda, F_\mu) \tag{20}$$

As $\mu \to \lambda$ we expect $\varphi_\lambda(\mu) = d_p(A_\lambda, A_\mu) \to 0$. Intuitively, we expect the dissimilarity function φ_λ to be unimodal[7]. locally around λ with a local minimum at $\mu = \lambda$.

Our experiments not only confirm the above conjecture, but indicate that there is a lot of structure in the way the dissimilarity function decreases to the local minimum: an exact power law is obeyed around λ. Moreover, local unimodality of f_λ around λ seems to be very robust with respect to dynamical and especially observation noise.

Once we know the dissimilarity function φ_λ is locally unimodal, with the real value of the parameter λ being the local minimum, we can use one of the standard optimization algorithms to determine λ. In our experiments we use the Golden Section Method, described in appendix A. This method uses the fact that we can bracket the location of the minimum of a unimodal function by evaluating the function at two distinct points in the region L of unimodality.

2.3 Performance of the Global Algorithm for the Quadratic Family

In this section we give a list of sample runs of the global algorithm for quadratic maps $f_c : \overline{\mathbb{C}} \to \overline{\mathbb{C}}, f_c(z) = z^2 + c$. If $c = \lambda + \xi i$ we want to consider the problem of estimating λ assuming ξ is fixed, in the presence of observation and/or dynamical noise (equations 11, 9).

Let $\tilde{F}_\lambda$ be the (noisy) global function whose discrete representation $\tilde{A}_\lambda$ is obtained by performing measurements (according to equation 11) on the noisy real system. Local minimization of the dissimilarity function

$$\varphi_\lambda : \mathbb{R} \to \mathbb{R}$$

$$f_\lambda(\mu) = d_1(\tilde{F}_\lambda, F_\mu)$$

is obtained using the Golden Section Method. Phase windows are represented as 512×512 arrays, and the Boundary Scanning algorithm used, checks for escape or orbits out of a circle of radius 100 centered at the origin. The noisy phase portrait $\tilde{A}_\lambda$ corresponds to $\lambda = -1.0$ and is obtained with Gaussian observation noise with variance $\sigma_o = 10^{-3}$ and no dynamic noise. The accuracy achieved gives an upper bound on the distance of the global method estimate from the real value of the parameter.

- $\lambda = 0.3$
  1. Domain : $[-1.5, -1.0] \times [0.0, 0.5]$
     Accuracy Achieved : 10^{-10}
     Number of Orbit Points in typical image : 4120137
     Average Number of Orbit Points per cell : 15
  2. Domain : $[-1.445, -1.37] \times [0.05, 0.2]$
     Accuracy Achieved : 10^{-11}
     Number of Orbit Points in typical image : 7150541
     Average Number of Orbit Points per cell : 27
  3. Domain : $[-1.40507, -1.40506] \times [0.100155, 0.100165]$
     Accuracy Achieved : 10^{-13}
     Number of Orbit Points in typical image : 17446074
     Average Number of Orbit Points per cell : 66
  4. Domain : $[-0.079555, -0.079525] \times [0.265320, 0.265350]$
     Accuracy Achieved : 10^{-15}
     Number of Orbit Points in typical image : 33692290
     Average Number of Orbit Points per cell : 128
  5. Domain : $[-0.079545, -0.079535] \times [0.265330, 0.265340]$
     Accuracy Achieved : 10^{-16}
     Number of Orbit Points in typical image : 36951018
     Average Number of Orbit Points per cell : 140
- λ=0.35
  1. Domain : $[-1.445, -1.37] \times [0.05, 0.2]$
     Accuracy Achieved : 10^{-9}
     Number of Orbit Points in typical image : 3905591
     Average Number of Orbit Points per cell : 15
  2. Domain : $[-1.4075, -1.4070] \times [0.0992, 0.0997]$
     Accuracy Achieved : 10^{-10}
     Number of Orbit Points in typical image : 9058660
     Average Number of Orbit Points per cell : 35
  3. Domain : $[-1.40715, -1.40710] \times [0.09957, 0.09962]$
     Accuracy Achieved : 10^{-11}
     Number of Orbit Points in typical image : 12225105
     Average Number of Orbit Points per cell : 47
- λ=0.40

[7] A function $\varphi : \mathbb{R} \to \mathbb{R}$ is said to be *unimodal* on a closed interval $L \subset \mathbb{R}$ if there is an $x^* \in L$ such that x^* minimizes φ on L and, for any two points $x_1, x_2 \in L$, such that $x_1 < x_2$ we have:

$$x_2 \leq x^* \Rightarrow f(x_1) > f(x_2)$$
$$x^* \leq x_1 \Rightarrow f(x_2) > f(x_1)$$

Note that unimodal functions are not necessarily differentiable or even continuous.Strictly convex functions and most of their generalizations are unimodal.

1. Domain : $[-1.445, -1.37] \times [0.05, 0.2]$
   Accuracy Achieved : 10^{-8}
   Number of Orbit Points in typical image : 3100158
   Average Number of Orbit Points per cell : 12

2.4 Cayley's Problem and Newton Basins

In this section, we consider the problem of estimating a parameter in a dynamical system obtained from Newton's method for cubic complex polynomials.

Newton's method and its variants are among the most prominent numerical methods for finding solutions to nonlinear equations. From a numerical point of view Newton's method has always been considered a local method, i.e. one assumes that the initial guess is sufficiently close to a root, and then the orbit under Newton's iteration scheme tends to this root.

In 1879 A. Cayley, in a one page paper [10], suggested the extension of what he calls the Newton-Fourier method

$$x_{k+1} = N(x_k) = x_k - h\frac{f(x_k)}{f'(x_k)} \tag{21}$$

applied to complex polynomials $f : \overline{\mathbb{C}} \to \overline{\mathbb{C}}$, where h is a real number. It is interesting to note that 21 can be interpreted as the Euler method with stepsize h for the initial value problem:

$$\dot{x}(t) = -\frac{f(x(t))}{f'(x(t))} \tag{22}$$

$$x(0) = x_0 \tag{23}$$

Each of the roots of f is an attracting fixed point of the Newton-Fourier iteration. Cayley suggested studying the method globally, i.e. understanding the geometry of the basins of attraction of the roots in phase space.

The problem is easy in the case of quadratic polynomials: we have seen that any quadratic map is analytically conjugate to one of the form $f_c(z) = z^2 + c$. Newton's method for f_c is a rational map of degree 2:

$$N(x) = \frac{x^2 + c}{2x} \tag{24}$$

It can be shown that the Julia set J_N of N is the perpendicular bisector of the segment joining the roots $\pm\sqrt{c}$. Thus the basins of attraction of the two roots are the half planes defined by J_N.

The geometry of the problem for higher degree polynomials is extremely complicated. To get a feeling for why that should be so, it is enough to note that we have an poynomial f of degree n, then if A_i is the basin of attraction of the i^{th} root ρ_i of f, we must have:

$$J_f = \partial A_i, \quad i = 1, \ldots, n$$

i.e. each point in the Julia set J_f must touch simultaneously all basins of attraction. Figure 5 shows the Newton basin portrait for the cubic

$$q(z) = (z - 0.5i)(z + 0.5i)(z - 1)$$

The values assigned to each cell in phase space correspond to convergence time to a root. If k is convergence time of the center point of a cell to ρ_i where $1 \leq i \leq 3$, then the value assigned to the cell in question is $3k + (i - 1)$.

Suppose ρ_1, ρ_2 are known. We want to consider the problem of estimating ρ_3 in the presence of observation and dynamical noise. We let

$$q_{\rho_1,\rho_2,\rho_3}(z) = (z - \rho_1)(z - \rho_2)(z - \rho_3)$$

and define $F_{\rho_1,\rho_2,\lambda}$ to be the Newton basin global image corresponding the polynomial $q_{\rho_1,\rho_2,\lambda}$. Suppose the real value for λ is $\lambda = 1.0$ and let us restrict λ to move on the real axis. This gives us a situation exactly analogous to the one for Julia sets of complex quadratic maps. Again we get an estimation algorithm by trying to minimize:

$$\varphi_{\rho_1,\rho_2} : \mathbb{R} \to \mathbb{R}$$

$$\varphi_{\rho_1,\rho_2}(\mu) = d_1(\tilde{F}_{\rho_1,\rho_2,\lambda}, F_{\rho_1,\rho_2,\mu})$$

where $\tilde{F}_{\rho_1,\rho_2,\lambda}$ is the noisy Newton basin global function for the third root equal to λ.

2.5 Performance of the Global Algorithm for Newton's Basins

In this section we give a list of sample runs of the global algorithm for the case of dynamical systems obtained by Newton's method. Local minimization of the dissimilarity function φ_{ρ_1,ρ_2} where $\rho_1 = 0.5i$ and $\rho_2 = -0.5i$, is obtained using the Golden Section Method. Global functions are represented as 512×512 arrays. If the number of iterations it takes for the centerpoint of a given cell to enter a neighborhood of one of the roots, say ρ_i (where $\rho_3 = \mu$ is the parameter) is k, then the value assigned to each the cell in question is $3k + (i - 1)$. The noisy phase portrait $\tilde{F}_{\rho_1,\rho_2,\lambda}$ corresponds to $\lambda = 1.0$ and is obtained with Gaussian observation noise with variance $\sigma_o = 10^{-3}$ and no dynamic noise. The accuracy achieved gives an upper bound on the distance of the global method estimate from the real value of the parameter.

1. Domain : $[-0.044, -0.024] \times [-0.105, -0.085]$
   Accuracy Achieved : 10^{-9}
   Number of Orbit Points in typical image : 10538413
   Average Number of Orbit Points per cell : 40

2. Domain : $[-0.033842, -0.033822] \times [-0.093942, -0.093922]$
   Accuracy Achieved : 10^{-13}
   Number of Orbit Points in typical image : 25907786
   Average Number of Orbit Points per cell : 99

3. Domain : $[-0.033833, -0.033832] \times [-0.0939325, -0.0939315]$
   Accuracy Achieved : 10^{-14}
   Number of Orbit Points in typical image : 34889936
   Average Number of Orbit Points per cell : 133

4. Domain : $[0.075, 0.080] \times [0.077, 0.080]$
   Accuracy Achieved : 10^{-8}
   Number of Orbit Points in typical image : 12464288
   Average Number of Orbit Points per cell : 48

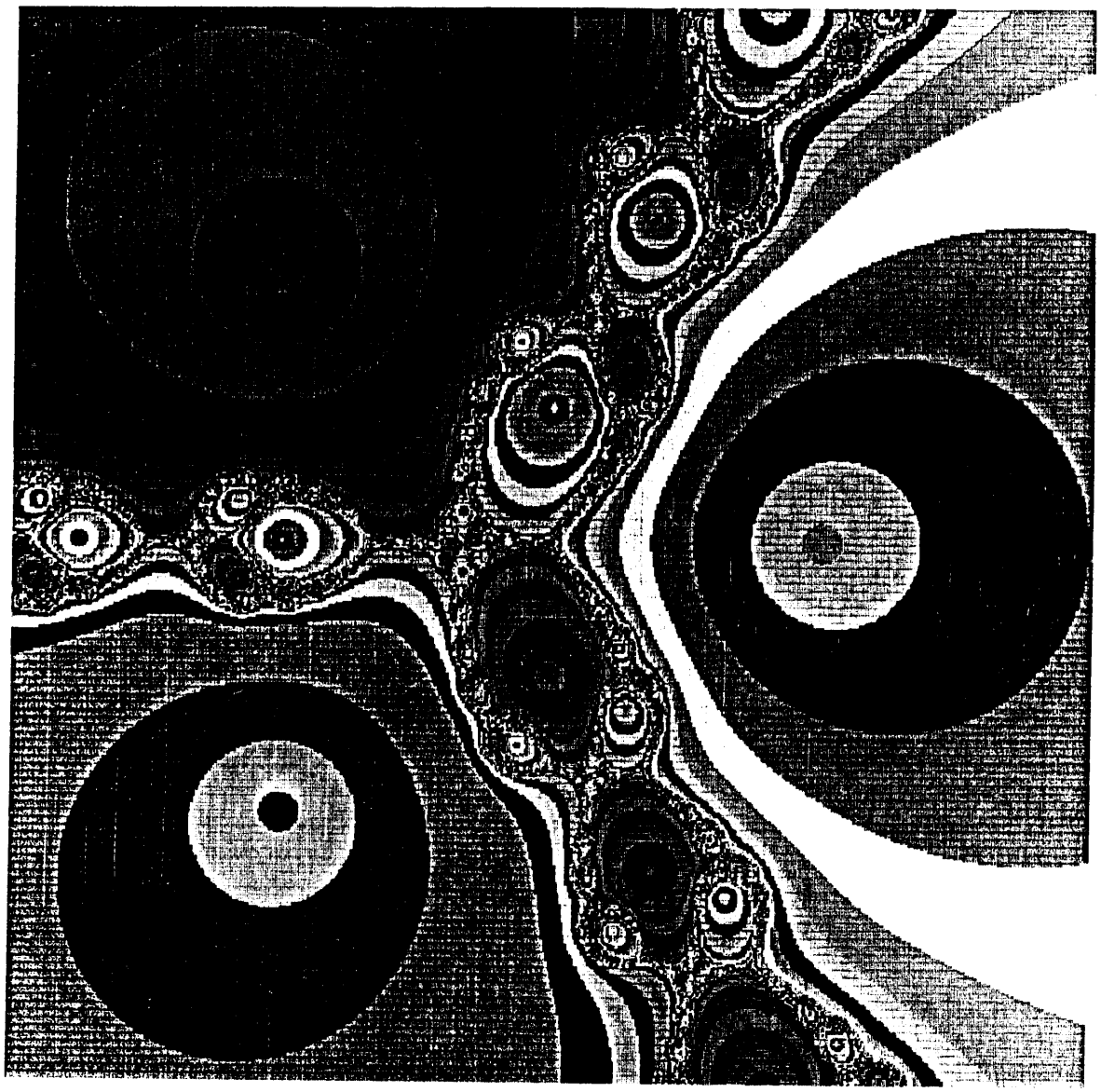


Figure 5: Newton Basins for $f(z) = (z - 0.5i)(z + 0.5i)(z - 1)$, and domain $D = [-0.5, 1.5] \times [-1.0, 1.0]$.

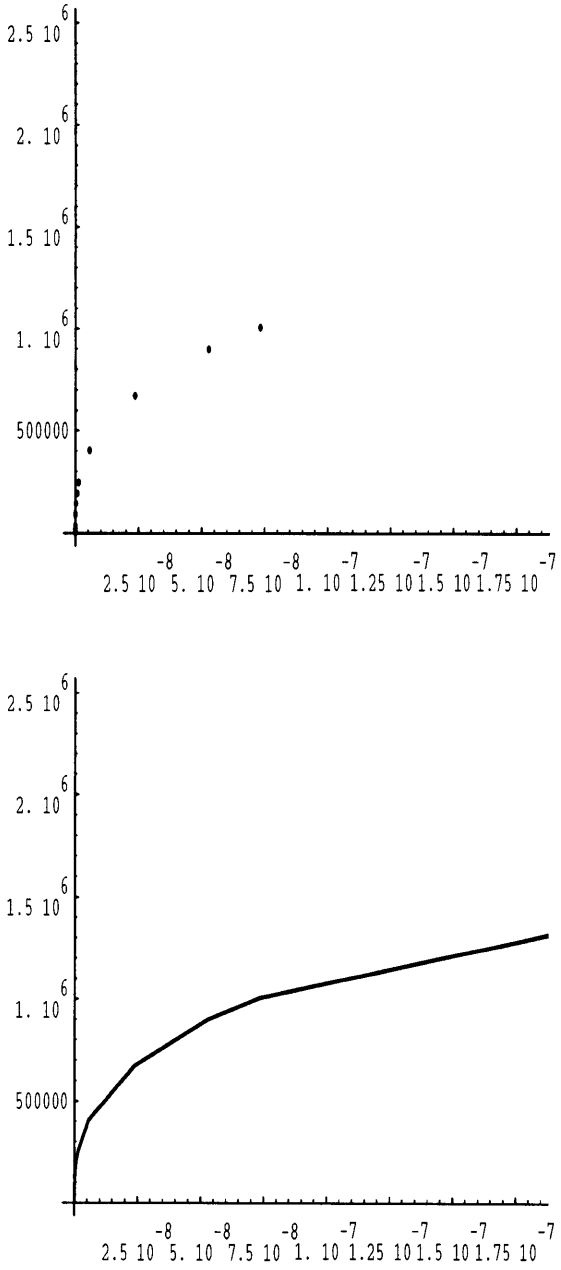
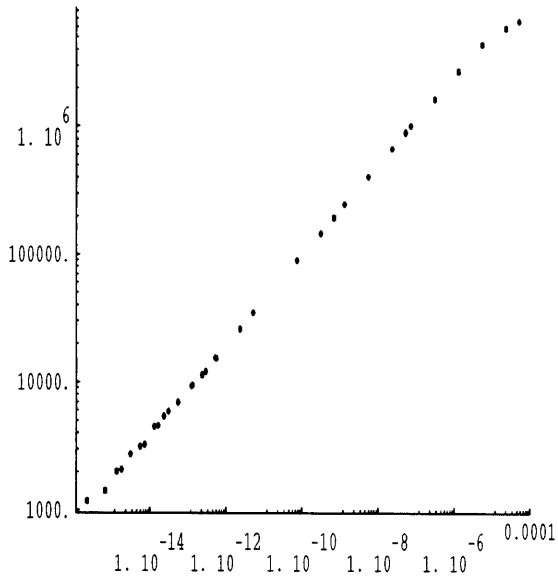


Figure 6: Quadratic Family: Plot of $\delta F_{\lambda,\delta\lambda}$ as a function of $||\delta\lambda||$, $\delta\lambda > 0$, for $\lambda = -1$, $\xi = 0.3$. The maximum number of iterations is 100, and the domain is $D = \{(x,y) : x \in [-0.079555, -0.079525], y \in [0.265320, 0.265350]\}$. A 512×512 cell resolution is used. The real image is measured with Gaussian observation noise with variance $\sigma_o = 10^{-3}$ and no dynamical noise ($\sigma_d = 0$). The plot on the top shows some sample points, and the one on the bottom is the same plot with straight lines connecting the sample points.



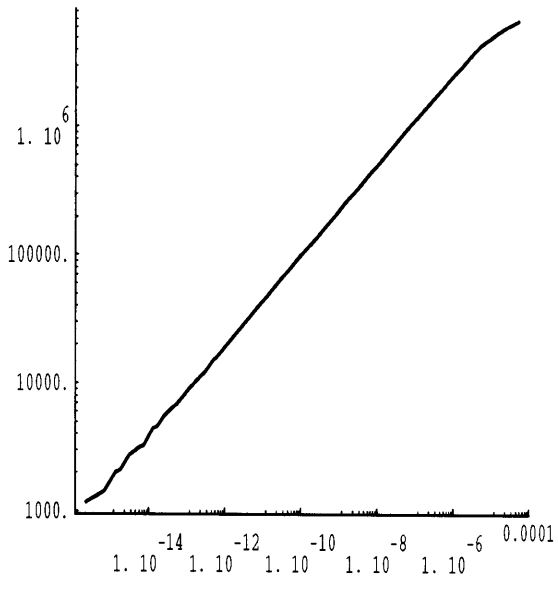


Figure 7: Quadratic Family: Plot of $\log(\delta F_{\lambda,\delta\lambda})$ as a function of $\log||\delta\lambda||$, $\delta\lambda > 0$, for $\lambda = -1$, $\xi = 0.3$. The maximum number of iterations is 100, and the domain is $D = \{(x,y) : x \in [-0.079555, -0.079525], y \in [0.265320, 0.265350]\}$. A 512×512 cell resolution is used. The real image is measured with Gaussian observation noise with variance $\sigma_o = 10^{-3}$ and no dynamical noise ($\sigma_d = 0$). The plot on the top shows some sample points, and the one on the bottom is the same plot with straight lines connecting the sample points.

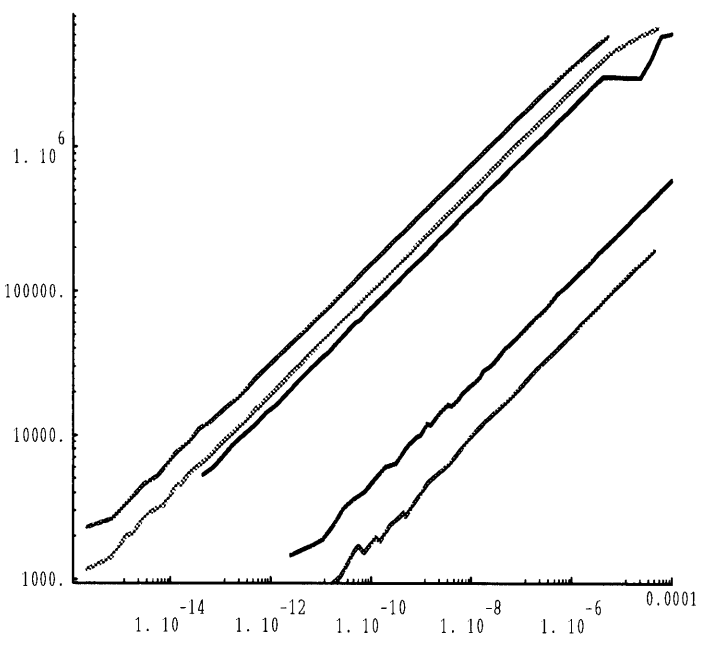


Figure 8: Quadratic Family: Plot of $\log(\delta F_{\lambda,\delta\lambda})$ as a function of $\log\|\delta\lambda\|$, $\delta\lambda > 0$, for $\lambda = -1$, $\xi = 0.3$. Each line corresponds to a different domain. The resolution increases from right to left. The real image is measured with Gaussian observation noise with variance $\sigma_o = 10^{-3}$ and no dynamical noise.

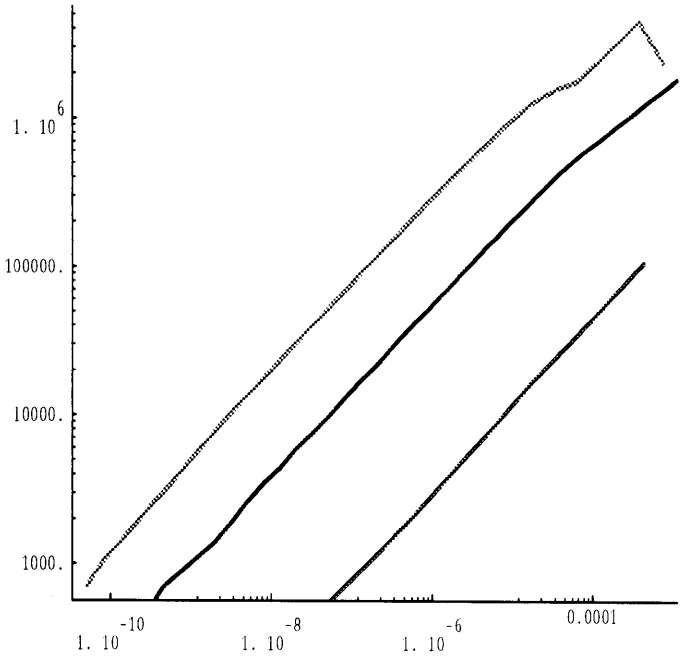


Figure 9: Quadratic Family: Plot of $\log(\delta F_{\lambda,\delta\lambda})$ as a function of $\log\|\delta\lambda\|$, $\delta\lambda > 0$, for $\lambda = -1$, $\xi = 0.3$. Each line corresponds to a different domain. The resolution increases from right to left. The real image is measured with Gaussian observation noise with variance $\sigma_o = 10^{-3}$ and no dynamical noise.

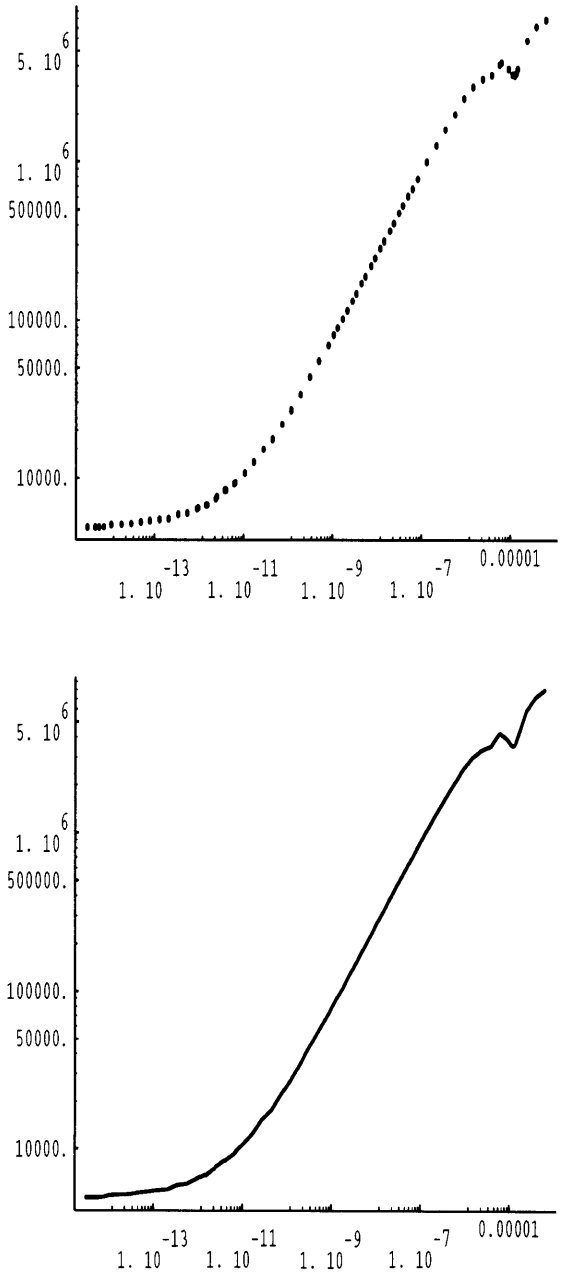


Figure 10: Newton Basins: Plot of $\log\varphi_{\rho_1,\rho_2}(\lambda+\delta\lambda)$ as a function of $||\delta\lambda||$, $\delta\lambda > 0$, for $\rho_1 = 0.5i, \rho_2 = -0.5i$, where $\lambda = 1$. The real image is measured with Gaussian observation noise with variance $\sigma_o = 10^{-3}$ and no dynamical noise ($\sigma_d = 0$). The plot on the top shows some sample points, and the one on the bottom is the same plot with straight lines connecting the sample points. A 512×512 cell resolution is used over the domain $D = \{(x,y) : x \in [-0.033842, -0.033822], y \in [-0.093942, -0.093922]$.

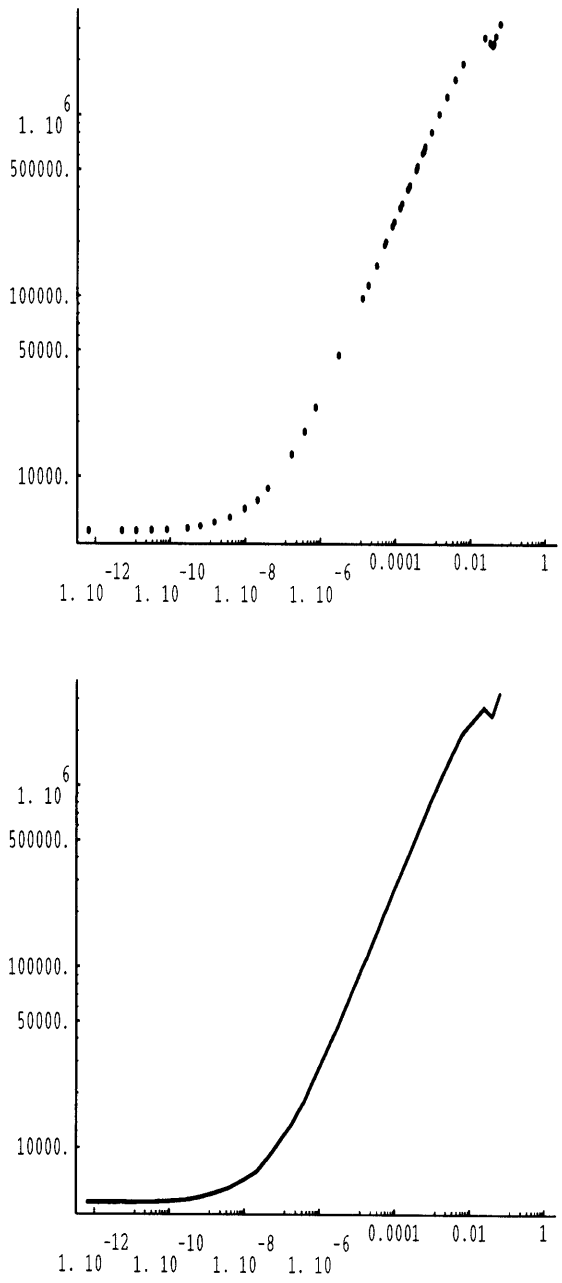


Figure 11: Newton Basins: Plot of $\log\varphi_{\rho_1,\rho_2}(\lambda+\delta\lambda)$ as a function of $||\delta\lambda||$, $\delta\lambda > 0$, for $\rho_1 = 0.5i, \rho_2 = -0.5i$, where $\lambda = 1$. The real image is measured with Gaussian observation noise with variance $\sigma_o = 10^{-3}$ and no dynamical noise ($\sigma_d = 0$). The plot on the top shows some sample points, and the one on the bottom is the same plot with straight lines connecting the sample points. A 512×512 cell resolution is used over the domain $D = \{(x,y) : x \in [-0.044, -0.024], y \in [-0.105, -0.085]\}$.

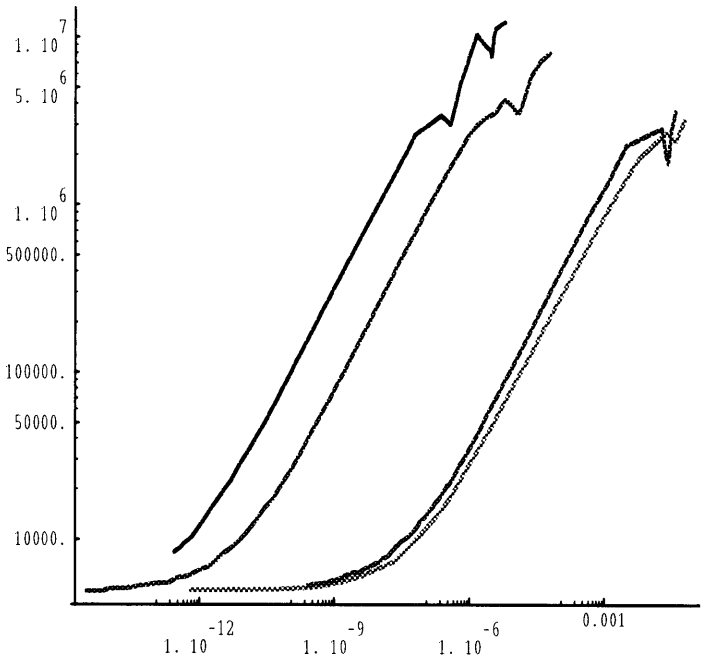


Figure 12: Newton Basins: Plot of $\log\varphi_{\rho_1,\rho_2}(\lambda+\delta\lambda)$ as a function as a function of $\log||\delta\lambda||$, $\delta\lambda > 0$, for $\rho_1 = 0.5i, \rho_2 = -0.5i$, $\lambda = 1$. Each line corresponds to a different domain. The resolution increases from right to left. The real image is measured with Gaussian observation noise with variance $\sigma_o = 10^{-3}$ and no dynamical noise.

3 The Parameter Sensitivity Power Law

"Let chaos storm!
Let cloud shapes swarm!
I wait for form."
R. Frost, "A Further Range: Ten Mills, (V) Pertinax" (1936).

The numerical experiments not only confirm that the dissimilarity function φ_λ is locally undimodal around λ, but indicate that there is a lot of structure in the way it decreases to the local minimum. In particular, if F_λ is the global function corresponding to λ, and $\mu = \lambda + \delta\lambda$, then for $||\delta\lambda||$ small enough, a power law of the form:

$$\boxed{\delta F_{\lambda,\delta\lambda} = d_p(F_{\lambda+\delta\lambda}, F_\lambda) = M||\delta\lambda||^d} \quad (25)$$

is obeyed. We will refer to equation 25 as the **parameter sensitivity power law**.

Let us consider, for example, the quadratic map $f_c(z) = z^2 + c$, $c = \lambda + \xi i$. Figure 6 is a plot of $d_1(F_{\lambda+\delta\lambda}, F_\lambda)$ as a function of $\delta\lambda$, for $\lambda = -1$, $\xi = 0.3$. They shape of the curve that we get immediately suggests a power law around $\lambda = -1$ with exponent less than 1. If in fact a power law of the form of equation 25 holds, then taking the logarithms of both sides gives a straight line of slope d:

$$\boxed{\log\delta F_{\lambda,\delta\lambda} = d\log||\delta\lambda|| + m} \quad (26)$$

where $m = \log M$. Figure 7 shows a log-log version of figure 6, which is in fact a straight line. We give the following definitions:

Definition 3.1 *Suppose that the local power law holds for some global function $F_\lambda = F_{\lambda,D} : D \to \mathbb{R}$. We define the parameter sensitivity exponent (p.s.e.) γ of the global function F_λ to be $\gamma = 1 - d$. Moreover, we define $m = \log M$ to be the resolution factor of F_λ.*

The parameter sensitivity exponent γ is a measure of the performance of the global algorithm, since it quantifies our ability to distinguish nearby global functions (images). If $\gamma = 0$, i.e. a linear power law is obeyed, the global function is rather insensitive to parameter variations. The performance of the global algorithm improves as γ gets closer to 1, i.e. as the slope d in the $\log - \log$ plot decreases towards 0.

Looking at the plot of $\log d_1(F_{\lambda+\delta\lambda}, F_\lambda)$ as a function of $\log||\delta\lambda||$ for different domains of a system and putting the resulting plots together (for example figures 8, 12), leads to the conclusion that the slope d changes very little as we change our focusing window, i.e. the domain D, in phase space! Hence it makes sense to talk about the *parameter sensitivity exponent of the system.* All experiments performed indicate that the following conjectures are true:

Conjecture 3.1 *The parameter sensitivity exponent of a system is the same for all typical domains*[8] .

[8] By *typical* we mean that the global function $F_\lambda : D \to \mathbb{R}$ is representative of the complexity of the system. For example, in the case of connected Julia sets a typical domain is any domain near the boundary $\partial A_c(\infty)$. A domain lying wholy in the interior of the Julia set, in which the global function is identically zero, is not a typical domain.

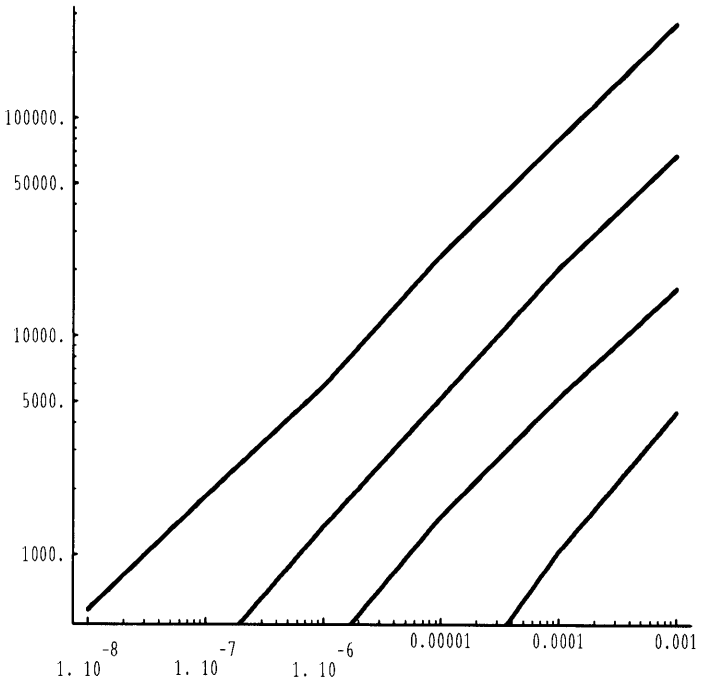


Figure 13: Quadratic Family: Plot of $\log(\delta F_{\lambda,\delta\lambda})$ as a function of $\log\|\delta\lambda\|$ for $\lambda = -0.12$ and $\xi = 0.74$. The maximum number of iterations is 100, and the domain is $D = \{(x,y) : x \in [0.3, 0.4], y \in [0.3, 0.4]\}$. The lines from left to right are for resolutions of 512×512, 256×256, 128×128, 64×64 cells.

Consequently, the **parameter sensitivity exponent seems to be a new dynamical system invariant** that quantifies the dependence of global geometry[9] on variations of the parameter.

Conjecture 3.2 *The parameter sensitivity exponent of a system is independent of the cell resolution*[10] *used in approximating the global function. Higher resolution only increases the resolution factor m.*

Of course the parameter sensitivity exponent (p.s.e.) is not exactly the same for all resolutions, because we get discretization errors in low resolutions. The p.s.e stays close to a dynamical system invariant and converges to it as the number of cells goes to infinity. An example on which conjecture 3.2 is tested is show in figure 13.

To gain some more insight into the power law let us rewrite equation 25 as:

$$\frac{\delta F_{\lambda,\delta\lambda}}{\delta\lambda} = \frac{M}{\|\delta\lambda\|^{1-d}} = \frac{M}{\|\delta\lambda\|^{\gamma}} \tag{27}$$

Consider the limit

$$\left(\frac{dF_\lambda}{d\lambda}\right)_{L^p} = \lim_{\delta\lambda\to 0} \frac{\delta F_{\lambda,\delta\lambda}}{\|\delta\lambda\|} \tag{28}$$

[9] or more generally, of global information as represented by the global function F_λ.

[10] The cell resolution is the number of cells used in the finite representation of the global function.

The above limit is a *mean derivative* in L^p-norm of the global function F_λ with respect to the parameter λ. We see that when $d = 1$, then the limit exists. When $d < 1$ the limit 28 does not converge, and the *parameter sensitivity exponent* $\gamma = 1 - d$ *measures its rate of divergence.*

Remark: The knowledge that a power law is obeyed locally around the true value λ of the parameter can be used to improve the performance (number of function evaluations) of the global approach enormously! Moreover, it can be used to check for errors and place a safe bound on the distance from the real value of the parameter.

3.1 The power law for smoothly changing global functions

In this section we demonstrate that if a global function is sufficiently smooth with respect to changes in the parameter, then a linear power law must be obeyed locally. In particular, we prove the following proposition:

Proposition 3.1 *Given a global function $F_\lambda : D \to \mathbb{R}$, if for every point $x \in D$, $F_\lambda(x)$ is twice differentiable with respect to the parameter λ and the derivative $d^2F_\lambda(x)/d\lambda^2$ is continuous and bounded in a neighborhood U of λ, then a linear power law is obeyed locally for any L^p-norm. Moreover, the L^p-derivative of F_λ with respect to λ exists and is equal to the space average $\|dF_\lambda/d\lambda\|_{L^p}$.*

Proof:

The proof is a straightforward application of Taylor's formula. Since $F_\lambda(x)$ has a continuous second derivative with respect to λ, then for any $\mu = \lambda + \delta\lambda \in U$ we have:

$$F_\mu(x) = \sum_{k=0}^{n} \frac{F_\lambda^k(x)}{k!}(\mu - \lambda)^k + E_{n,x}(\mu) \tag{29}$$

(30)

$$E_{n,x}(\mu) = \frac{1}{n!}\int_\lambda^\mu (\mu - t)^n F_t^{n+1}(x)dt$$

where $F_\lambda^k(x) = d^kF_\lambda(x)/d\lambda^k$ and $n = 1$. If A is an upper bound for $F_t^2(x)$ over U, then

$$\|E_{1,x}(\mu)\| \le \frac{A}{2!}\|\mu - \lambda\|^2 = \frac{A}{2}\|\delta\lambda\|^2 \tag{31}$$

Consequently, for $\delta\lambda$ small enough we have

$$\|F_{\lambda+\delta\lambda}(x) - F_\lambda(x)\| \approx \|\frac{dF_\lambda(x)}{d\lambda}\delta\lambda\| \tag{32}$$

The L^p-distance between F_λ and F_μ is:

(33) $\delta F(\lambda, \delta\lambda) = d_p(F_{\lambda+\delta\lambda}, F_\lambda)$

$$= \left(\int_D \|F_{\lambda+\delta\lambda}(x) - F_\lambda(x)\|^p dx\right)^{1/p}$$

Hence, for sufficiently small $\|\delta\lambda\|$:

$$\delta F(\lambda, \delta\lambda) \approx \left(\int_D \|\frac{dF_\lambda}{d\lambda}\|^p dx\right)^{1/p} \|\delta\lambda\| \tag{34}$$

The above linear power law implies that:

(35)

$$\left(\frac{dF_\lambda}{d\lambda}\right)_{L^p} = \lim_{\delta\lambda\to 0} \frac{\delta F_{\lambda,\delta\lambda}}{\|\delta\lambda\|} = \left(\int_D \|\frac{dF_\lambda}{d\lambda}\|^p dx\right)^{1/p}$$

□

The above proof not only shows that a power law holds locally, but actually it holds pointwise. The following lemma gives us a simple test for pointwise validity of the power law.

Lemma 3.1 *If a power law holds pointwise in norm L^{p_0}, then the power law with the same exponent holds for any norm L^p.*

The proof is trivial. It turns out that when the exponent of the parameter sensitivity power law is not 1, the law does not hold pointwise, but is the effect of spatial averaging in L^p-norm. Figure 14 gives an example of a system where this is tested using lemma 3.1.

Consider now the linear differential equation

$$\frac{dx}{dt} = h(\lambda)x \tag{36}$$

on the real line $\mathbb{R}$. The solution of the above equation (equation 36) satisfying the initial condition $x(0) = x_0$ is given by

$$x(t) = x_0 e^{h(\lambda)t} \tag{37}$$

Suppose $h(\lambda) > 0$. In this case, we have a repelling fixed point at $x = 0$. Given a domain $D = [a, b] \subset \mathbb{R}$, and we define a function $F_\lambda = F_{\lambda,M} : D \to \mathbb{R}$ assigning to each point in the domain D its escape time from a large interval $[-M, M]$. Let us ignore the point $x = 0$, where $F_\lambda(x) = 0$, since it does not change anything when we integrate over it. For x different from zero we have:

$$F_\lambda(x) = \frac{\ln M - \ln \|x\|}{g(\lambda)}$$

For most smooth functions g the above global function satisfies the requirements of proposition 3.1 hence a linear power law is valid.

For example, if $g(\lambda) = e^\lambda$, we get

$$\begin{aligned}\frac{d^2 F_\lambda(x)}{d\lambda^2} &= (\ln M - \ln \|x\|)\frac{d^2}{d\lambda^2}\left(\frac{1}{g(\lambda)}\right) \\ &= (\ln M - \ln \|x\|)e^{-\lambda}\end{aligned}$$

which is continuous and bounded for any λ. Similarly, for $g(\lambda) = \lambda$ a linear law holds around any λ different from 0.

The same reasoning applies to the case when $g(\lambda) < 0$. The only difference is that $F_\lambda(x)$ measures the time needed to enter some predetermined neighborhood of the attracting fixed point 0, instead of a neighborhood of infinity.

Let us now consider a linear system

$$\frac{dx}{dt} = A(\lambda)x \tag{38}$$

where $x \in \mathbb{R}^n$. A general solution to (38) can be obtained by a linear superposition of n linearly independent solutions $\{x^1(t), \ldots, x^n(t)\}$:

$$x(t) = \sum_{j=1}^{n} c_j x^j(t) \tag{39}$$

where the n unknown constants c_j are to be determined by initial conditions. If $A(\lambda)$ has n linearly independent eigenvectors v^j, $1 \le j \le n$, then we may take as a basis for the space of solutions the vector valued functions

$$x^j(t) = e^{\lambda_j t} v^j \tag{40}$$

where λ_j is the eigenvalue associated with v^j. Let us assume that al least one of the eigenvalues is real and positive. Let λ_1 be the largest of the positives eigenvalues. Then for t very large we have

$$x(t) \approx c_1 e^{\lambda_1 t} v^1$$

Hence escape time from a very large circle around the origin is approximately

$$F_\lambda(x) \approx \frac{\ln M - \ln \|c_1 v^1\|}{\lambda_1(\lambda)}$$

Essentially the problem is reduced to the one-dimensional problem considered above. If A is a sufficiently smooth function of λ, the maximum eigenvalue λ_1 is a sufficiently smooth function of λ and the requirements of proposition 3.1 are satisfied. All this can be made more rigorous.

It seems that almost all linear systems of differential equations would give rise to dynamical systems exhibiting a linear parameter sensitivity law with respect to global functions measuring convergence time to the fixed point and infinity. A more precise and rigorous theory of global estimation on linear systems can easily be developed.

In the previous section we have discussed discrete maps rather than continuous ones. In those cases time is discrete and hence F_λ might not be differentiable with respect to λ. For example, if we take the discrete dynamical system:

$$x_{n+1} = \lambda x_n, \quad \lambda > 1$$

then escape time out of a circle of radius M is given by

$$F_\lambda(x) = \left\lceil \frac{\log \|\frac{M}{x}\|}{\log \lambda} \right\rceil$$

Proposition 3.1 can still be applied in the sense that the curves of equal discrete escape time are just approximations to curves of the continuous escape time $G_\lambda(x) = \log \|\frac{x}{M}\| / \log \lambda$ which satisfies the requirements of the proposition.

3.2 Additional Examples

3.2.1 The Forced Pendulum

So far we have only seen discrete dynamical systems from complex analytic dynamics exhibiting a positive parameter sensitivity exponent (p.s.e.). In this section, we provide an example of a continuous time dynamical system with high p.s.e. which has a completely different dynamical structure, and enforces the belief for the universality of the parameter sensitivity power law.

In particular, we consider the forced pendulum described by the equation (see [3, 23])

$$\frac{d^2\theta}{dt^2} + \lambda\frac{d\theta}{dt} + \beta \sin\theta = \gamma \cos t \tag{41}$$

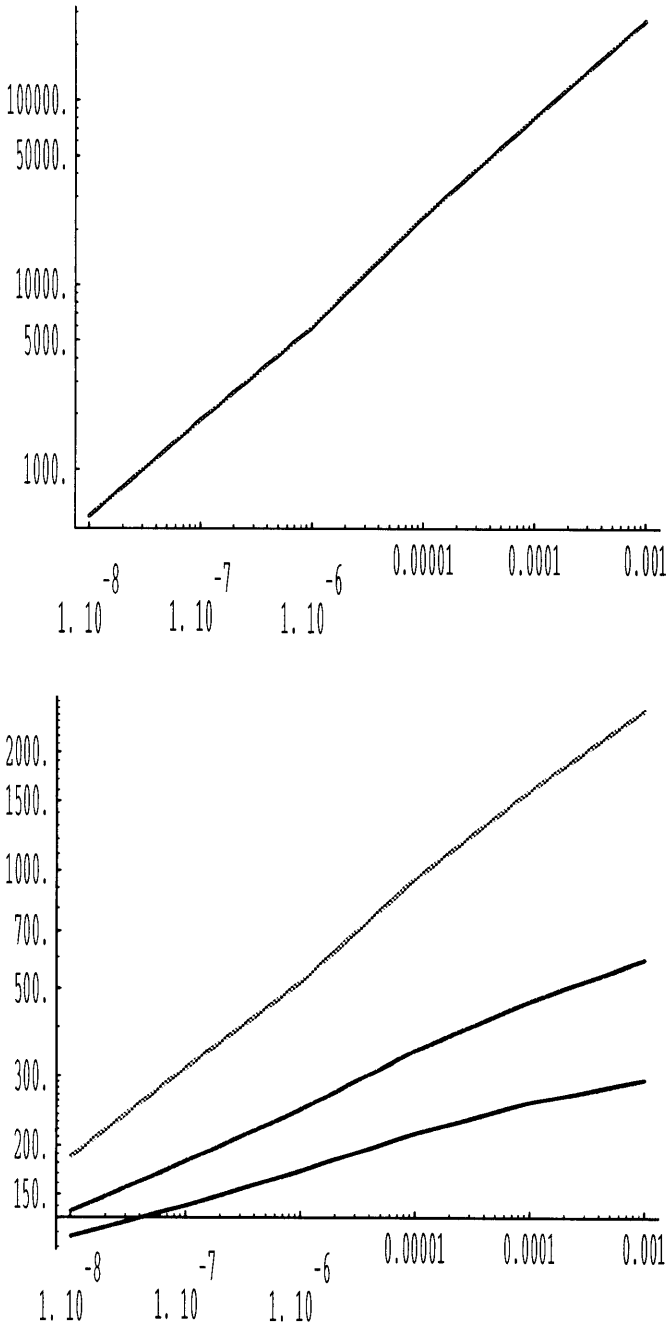


Figure 14: Plot of $\log(\delta F_{\lambda,\delta\lambda})$ vs. $\log\|\delta\lambda\|$ for the quadratic map with $c = \lambda + 0.74i$, $\lambda = -0.12$, where F_λ is escape time out of a circle of radius $R = 100$. A resolution of 512×512 cells is used, and the domain chosen is $D = [0.3, 0.4] \times [0.3, 0.4]$. The plot on the top is a display of the data in L^1-norm. The slope is $d \approx 0.54$. The plot on the bottom shows the data in L^2-, L^3-, L^4-norm (from top to bottom) with slopes $\approx 0.23, 0.13, 0.08$ respectively.

For parameter values $\lambda = 0.1, \beta = 1.0, \gamma = 1.75$ the system has (at least) four attracting periodic orbits (each having period 2π). For the time 2π Poincaré return map, these orbits are attracting fixed points located at:

$$p_1 \approx (3.287, 0.262) \tag{42}$$

$$p_2 \approx (4.301, 0.397) \tag{43}$$

$$p_3 \approx (0.053, -1.070) \tag{44}$$

$$p_4 \approx (0.084, 1.608) \tag{45}$$

Let us assume the unknown parameter is λ, for a given a domain D define the global function $F_\lambda : D \to \mathbb{R}$ to have discrete representation A_λ assigning to each cell the label of the fixed point its centerpoint is attracted to. This means that if the centerpoint p is attracted to p_i we assign to the cell the value i. This is a *very coarse* representation of a global function: we do not record convergence time or any such information; just the label of the attractor. Figure 15 shows a picture of the resulting image for a 128×128 cell decomposition of the domain $D = [0, 2\pi] \times [-2, 4]$. All computations have been made with a Bulirsch-Stoer integrator.

Figure 16 demonstrates that the local parameter sensitivity power law is obeyed in a very impressive way around $\lambda = 0.1$. The parameter sensitivity exponent seems to be $\gamma = 1 - d \approx 0.907$ This is an enormous p.s.e., giving a global parameter estimation algorithm with very impressive performance.

3.2.2 The tent map

The next step is to consider the simplest possible systems for which the parameter sensitivity power law holds, and for which an analytic proof is possible.

To that end, consider the tent map:

$$f_\lambda = f : I \to I$$

$$f_\lambda(x) = \begin{cases} \lambda x & \text{if } x < 0.5, \\ \lambda(1-x) & \text{if } x > 0.5. \end{cases} \tag{46}$$

where $I = [0, 1]$ is the unit interval. The global function $F_\lambda : D \to \mathbb{R}, D \subset I$ that we will consider, just like in previous cases, measures escape time from an interval $[-M, M]$. In particular, we will take $M = 1$, i.e. measure escape times from the unit interval.

If the slope λ is less than 1, then no points ever escape from the unit interval. Consequently, $F_\lambda(x) = 0$, for all $x \in I$.

Let us restrict to the case $\lambda > 1$. If we define α to be $1/\lambda$, then we notice that the points in the interval

$$I_1(\lambda) = [\alpha, 1 - \alpha]$$

escape after one iteration of the tent map f_λ. Moreover, points in the intervals

$$I_2^1(\lambda) = [\alpha^2, \alpha - \alpha^2], \quad I_2^2(\lambda) = [1 - (\alpha - \alpha^2), 1 - \alpha^2]$$

escape after 2 iterations, and points in

$$I_3^1(\lambda) = [\alpha^3, \alpha^2 - \alpha^3]$$

$$I_3^2(\lambda) = [\alpha - (\alpha^2 - \alpha^3), \alpha - \alpha^3]$$

$$I_3^3(\lambda) = [1 - (\alpha^2 - \alpha^3), 1 - \alpha^3]$$

$$I_3^4(\lambda) = [1 - (\alpha - \alpha^3), 1 - (\alpha - (\alpha^2 - \alpha^3))]$$

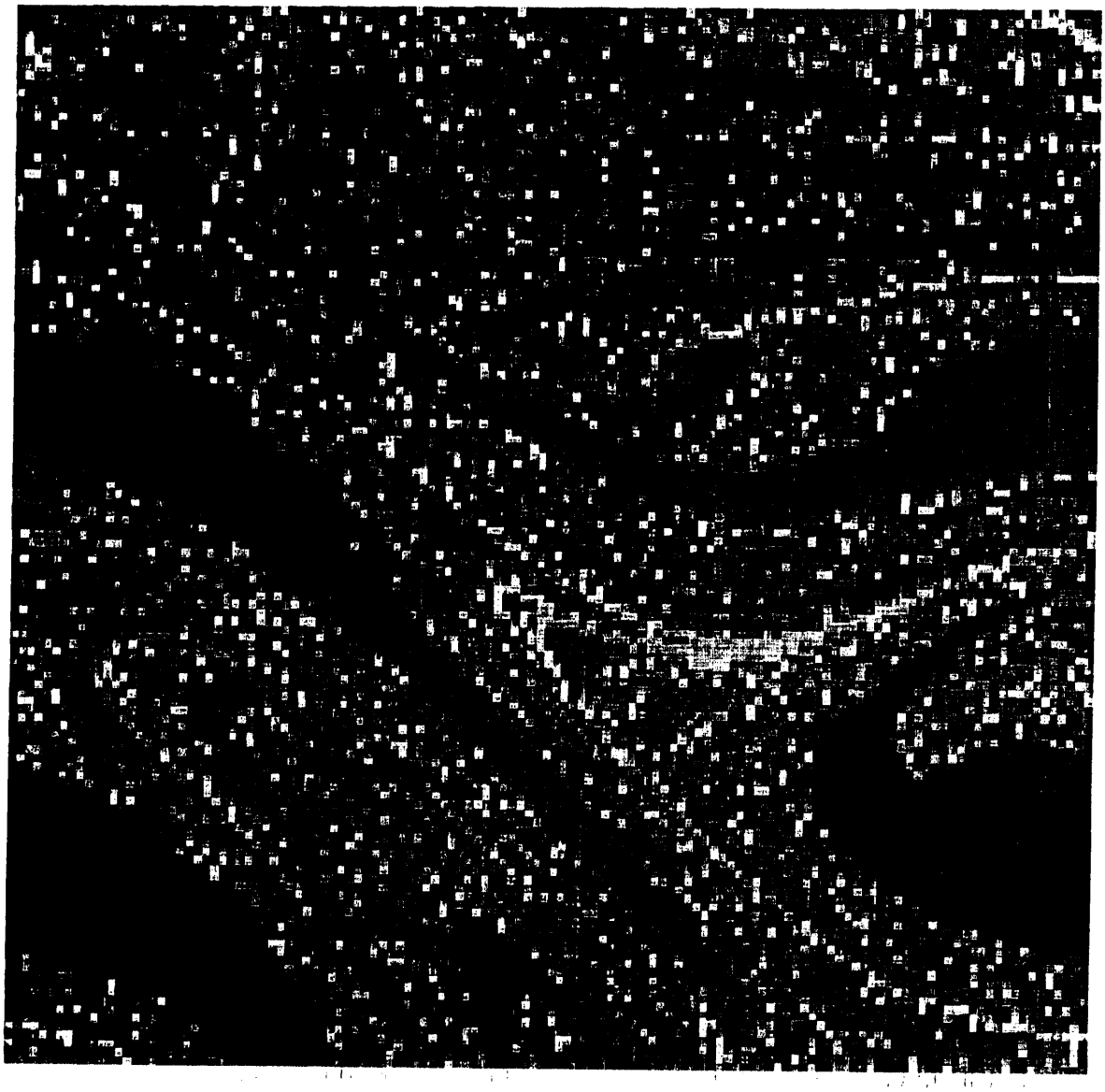


Figure 15: The basins of attraction of the four fixed points of the 2π Poincaré return map for the forced pendulum with $\lambda = 0.1, \beta = 1.0, \gamma = 1.75$, over the domain $D = [0, 2\pi] \times [-2, 4]$.

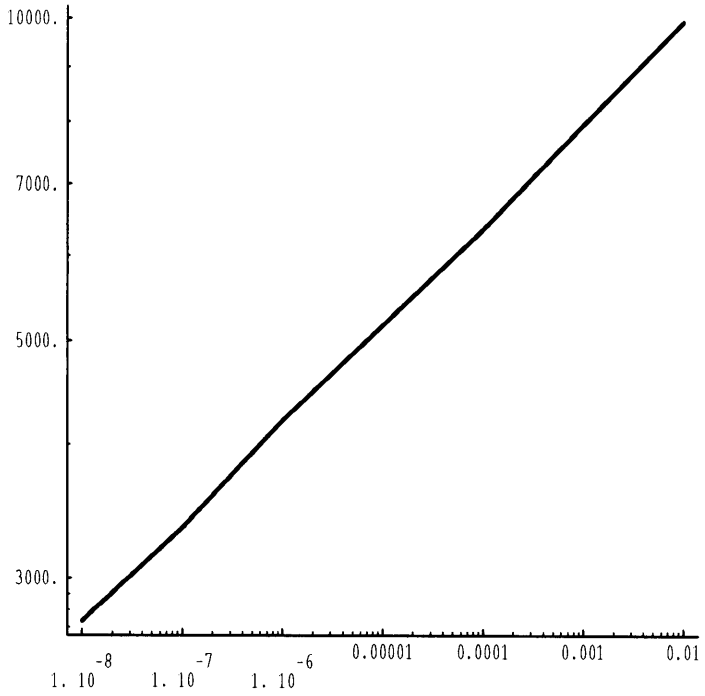


Figure 16: Plot of $\log(\delta F_{\lambda,\delta\lambda})$ vs. $\log||\delta\lambda||$ for the forced pendulum for $\lambda = 0.1, \beta = 1.0, \gamma = 1.75$. The are four attracting fixed points $\{p_1, p_2, p_3, p_4\}$ for the 2π Poincaré return map. $F_\lambda(x)$ is just the label i of the fixed point to which x is attracted ($1 \leq i \leq 4$). A resolution of 128×128 cells is used, and the domain chosen is $D = [0, 2\pi] \times [-2, 4]$. L^1-norm is used and the slope of the line is approximately 0.093!!!

escape after 3 iterations. In general, there are 2^{k-1} intervals each of which has length $l_k = \alpha^{k-1}(1 - 2\alpha)$ containing points that escape after k iterations.

Figure 17 clearly shows that the parameter sensitivity power law holds, and that the p.s.e. (the slope of the lines in the log-log plots) is an invariant of the system independent of the resolution.

3.2.3 A fractal curve boundary of attraction

Given $1 < \lambda < 2$, consider the map ([14], [18]) $M_\lambda = M : \mathbb{R} \times S^1 \to \mathbb{R} \times S^1$, defined by

$$M(x_k, \theta_k) = (x_{k+1}, \theta_{k+1}) \tag{47}$$

where

$$x_{k+1} = \lambda x_k + \cos\theta_k \tag{48}$$

$$\theta_{k+1} = 2\theta_k \pmod{2\pi} \tag{49}$$

This map has two attractors $+\infty$ and $-\infty$ for the first component, meaning, if $f^n(x_0, \theta_0) = (x_n, \theta_n)$ then $\lim_{n\to\infty} x_n = \pm\infty$. M has no finite attractors since the eigenvalues of the Jacobian matrix are 2 and $\lambda > 1$.

To calculate what the boundary between the basins of attraction $A_\lambda(\pm\infty)$, we proceed as in [18]. We first notice that given any initial point (x_0, θ_0), we have $\theta_k = 2^k\theta_0$. The map M is two to one (and hence noninvertible), but given any point x_N and $\theta_N = 2^N\theta_0$ we can

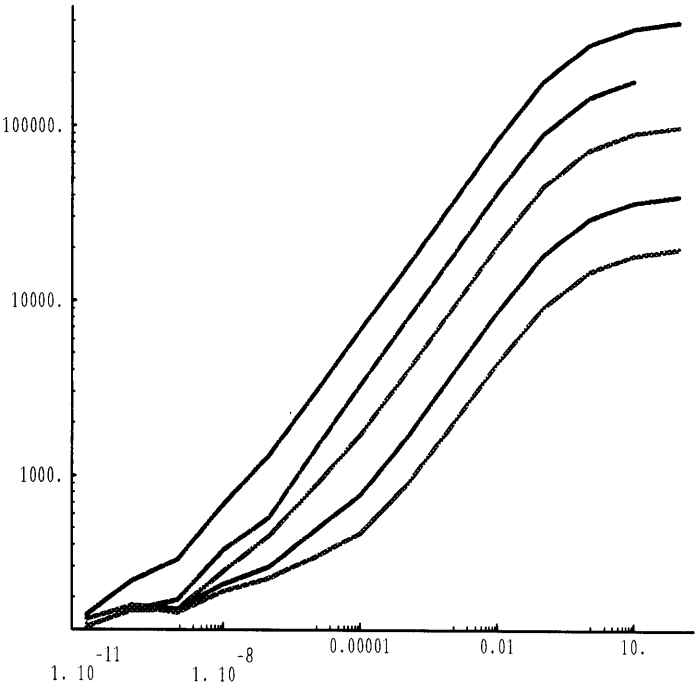


Figure 17: Plots of $\log(\delta F_{\lambda,\delta\lambda})$ vs. $\log||\delta\lambda||$ for the tent map, with $\lambda = 3$, where F_λ is escape time out of the unit interval, with a maximum number of iterations tolerated equal to 1000. The domain chosen is the unit interval. The plots are, from top to bottom, the data with a resolution of 200000, 100000, 50000, 30000, 10000 cells. The slope of the linear segments is about $d \approx 0.35$, i.e. $\gamma \approx 0.65$.

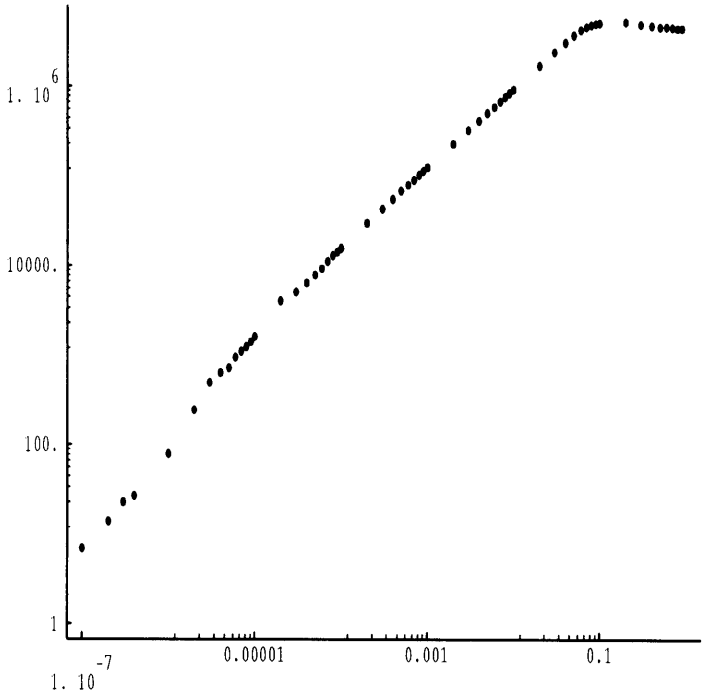


Figure 18: Plot of $\log(\delta F_{\lambda,\delta\lambda})$ vs. $\log||\delta\lambda||$ for the map 47, where F_λ is escape time out of a circle of radius $R = 150$ (maximum number of iterations tolerated is 200). The parameter value is $\lambda = 1.5$, a resolution of 256×256 cells is used, and the domain chosen is $D = [-1.3, -1.2] \times [0.1, 0.2]$.

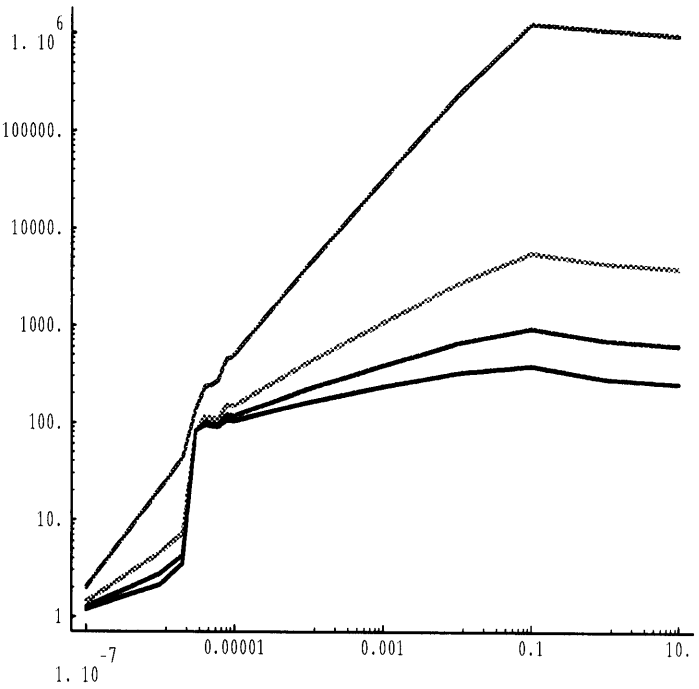


Figure 19: Plot of $\log(\delta F_{\lambda,\delta\lambda})$ vs. $\log||\delta\lambda||$ for the map 47, where F_λ is escape time out of a circle of radius $R = 150$ (maximum number of iterations tolerated is 200). The parameter value is $\lambda = 1.5$, a resolution of 256×256 cells is used, and the domain chosen is $D = [-1.3, -1.2] \times [0.1, 0.2]$. The plots are, from top to bottom, the data in L^1, L^2, L^3, L^4 norm.

always select an orbit that ends at (x_N, θ_N), by taking $x_{k-1} = \lambda^{-1}x_k - \lambda^{-1}\cos(2^{k-1}\theta_0)$. This orbit starts at

$$x_0 = \lambda^{-N}x_N - \sum_{i=0}^{N-1}(\frac{1}{\lambda})^{i+1}\cos(2^i\theta_0) \tag{50}$$

The boundary between the two basins $A_\lambda(\pm\infty)$ are those (x_0, θ_0) for which x_N remains finite as $N \to \infty$. So:

$$\partial A_\lambda(\pm\infty) = \{(x, \theta) : x = f_\lambda(\theta)\}$$

where

$$f_\lambda(\theta) = -\sum_{i=0}^{\infty}(\frac{1}{\lambda})^{i+1}\cos(2^i\theta) \tag{51}$$

Since $\lambda > 1$ the above sum converges absolutely and uniformly. In addition, the following sum

$$\frac{df_\lambda(\theta)}{d\theta} = \frac{1}{2}\sum_{i=0}^{\infty}(\frac{2}{\lambda})^{i+1}\sin(2^i\theta)$$

diverges, because $\lambda < 2$. Hence the curve $f_\lambda(\theta)$ is nondifferentiable. Moreover it has been proved (in [21]) $f_\lambda(\theta)$ has fractal dimension $d_c = 2 - \ln\lambda/\ln 2$.

Figure 18 is evidence that the parameter sensitivity power law holds locally, and figure 19 proves that the power law does not hold pointwise.

4 The Effect of Noise on Convergence

We notice that the information obtained from each orbit (number of iterations it takes to enter a neighborhood of an attractor) is very insensitive to observation noise, i.e. noise that enters in the measurement equation (for example equation 11 for the quadratic family)[11]. Moreover, observation noise is averaged away by taking the L^p norm of the corresponding images. Consequently, observation noise has no important effect on convergence properties of the global method.

It is much more interesting to see how the dynamic noise, i.e. noise that enters in the dynamic evolution equation, affects the convergence and accuracy of the global approach. It is very important to check that reasonably small dynamic noise does not destroy convergence properties of the global algorithms because in all systems some sort of small dynamical noise always exists. In particular, all numerical simulations introduce dynamic noise by rounding off.

Let us call the the log-log plot of the L^p distance of the image $F_{\lambda+\delta\lambda}$ generated for parameter value $\lambda + \delta\lambda$ from the noisy image $\tilde{F}_\lambda$ corresponding to the true value λ as

[11] We believe this is a remarkable property: using very coarse information from each orbit (coarse building blocks), we build a structure containing global information which is very delicate with respect to changes in parameters (fine overall structure).

a function of $||\delta\lambda||$ the *performance curve* of the global algorithm.

Our experiments have shown that the presence of dynamic noise has the effect of leveling off the performance curve of the global algorithm to around a value of the order of the difference $d_p(F_\lambda, \tilde{F}_\lambda)$.

It is very important to note that if the dynamical noise is below a certain value, then the local unimodality of the dissimilarity function is not destroyed. This is demonstrated by the experiments depicted in figures 20 and 21.

Intuitively, we would expect the presence of dynamic noise not to be disastrous, since the results of sections 2.3,2.5 show that on the average there are from 10 to 150 iterations per orbit, and the dynamic noise does not have the time to manifest itself if it is small enough. On the contrary, local algorithms attempt to follow a nominal trajectory for significantly longer. Hence, the presence of dynamic noise can have disastrouss effects on the convergence properties of local algorithms, especially in the case of chaotic dynamical systems.

5 Combining Estimation and Control

Let us onsider again the problem of estimating the parameter λ in the case of the quadratic system 7, but now instead of assuming that ξ is fixed, suppose ξ is a control parameter that we can tune. We want to address the question of what value of ξ gives optimal performance for the global method for estimating λ.

Consider the domain $D = [-1.445, -1.37] \times [0.05, 0.2]$, and observe how the performance plots of the global algorithm for the given phase space window change as we vary λ (figure 24). We notice that as we approach the boundary of the Mandelbrot set the global algorithm reaches better and better accuracy, increasing the parameter sensitivity exponent (p.s.e.) $\gamma = 1 - d$ for the domain. So our experiments suggest that we can maximize the p.s.e. γ and hence optimize performance of the global algorithm by choosing ξ so as to minimize the distance of $c = \lambda + \xi i$ from the boundary of Mandelbrot set M.

We remark that we should be careful about the above statement since the distance to the Mandelbrot set is not the only thing that controls γ: it matters a lot *which part of the Mandelbrot set we are close to.* Consider for example varying λ close to the point $c = -0.75 + 0i$ (the round top of the main cardioid of the Mandelbrot set). The point $c = -0.75$ is on the boundary of the Mandelbrot set, but as we move λ along the real axis we move inside the Mandelbrot set obtaining a parameter sensitivity exponent (p.s.e.) of $\gamma \approx 1 - 0.687 = 0.313$. A much bigger p.s.e. ($\gamma \approx 1 - 0.218 = 0.782$) is obtained when we are close to the boundary of M on the vertical line $\lambda = -1.0$.

We have seen that the boundary of the Mandelbrot set is a *bifurcation set* for the quadratic family, since the topology of the Julia set changes as we cross it. We conjecture that for more general systems we can get optimal performance out of a global estimation algorithm by tuning control parameters so as to drive the system near a bifurcation set in parameter space. Intuitively we expect

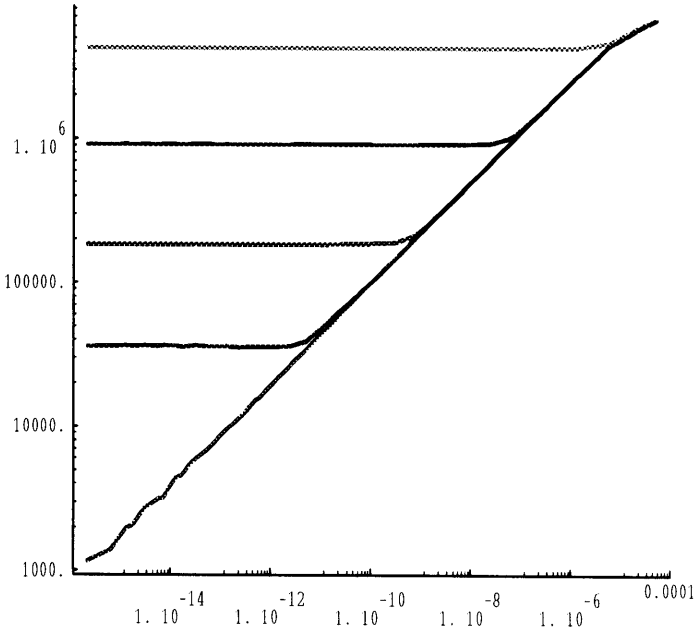


Figure 20: Quadratic Family: Plots of $\log(\delta F_{\lambda,\delta\lambda})$ as a function of $\log||\delta\lambda||$, $\delta\lambda > 0$, for $\xi = 0.3$, for varius strengths of dynamic noise, and fixed observation noise with $\sigma_o = 10^{-3}$. The curves (from top to bottom) correspond to dynamic noise $\sigma_d = 10^{-6}, 10^{-8}, 10^{-10}, 10^{-12}$. The maximum number of iterations is 100, and the domain is $D = \{(x, y) : x \in [-0.079555, -0.079525], y \in [0.265320, 0.265350]\}$. A 512×512 cell resolution is used.

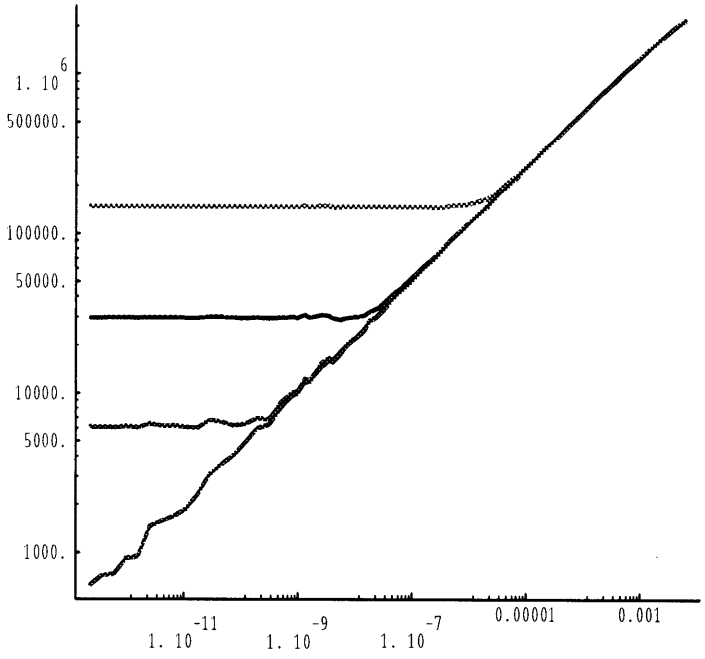


Figure 21: Quadratic Family: Plots of $\log(\delta F_{\lambda,\delta\lambda})$ as a function of $\log||\delta\lambda||$, $\delta\lambda > 0$, for $\xi = 0.3$, for varius strengths of dynamic noise, and fixed observation noise with $\sigma_o = 10^{-3}$. The curves (from top to bottom) correspond to dynamic noise $\sigma_d = 10^{-6}, 10^{-8}, 10^{-10}$. The maximum number of iterations is 100, and the domain is $D = \{(x, y) : x \in [-1.445, -1.37], y \in [0.05, 0.2]\}$. A 512×512 cell resolution is used.

to have much more sensitivity to changes in parameters (high p.s.e.) as we get close to bifurcations.

We believe it would be very interesting to develop a general theory of optimal global geometric estimation through control and in general analyze the dependece of the the parameter sensitivity exponent on the position in parameter space.

5.1 A Global Geometric Controller for the Quadratic Family

In this section we will show how to use a result of J. Milnor and W. Thurston [25, 27] for bounding the distance to the Mandelbrot set M, to obtain a global geometric controller for the quadratic family. Our method for determining the control ξ value gives an astonishingly good performance, improving the initial estimate by more than 13 orders of magnitude (figure 26), and improving the best estimate that we had for $\xi = 0.3$ by more than 7 orders of magnitude for the given domain.

5.1.1 Bounding the distance to M

J. Milnor and W. Thurston proved a bound for the distance $d(c, M)$ of a point c in the parameter plane from the Mandelbrot set M, on which some algorithms for producing pictures of the Mandelbrot set are based. In particular, they have shown the following result (see [27]):

Theorem 5.1 *If c is a point outside of M, then*

$$\frac{\sinh G(c)}{2\exp G(c)||G'(c)||} < d(c, M) < \frac{2\sinh G(c)}{||G'(c)||} \tag{52}$$

Similar inequalities can be obtained for points inside the Mandelbrot set, as well as for the distance of points in the dynamical plane from connected Julia sets.

Let us approximate, somewhat arbitrarily, the distance of a point c outside of M, by the estimate of the upper bound in inequality 52, i.e. let

$$d(c, M) \approx \frac{2\sinh G(c)}{||G'(c)||} \tag{53}$$

For c near M, we may approximate $\sinh G(c)$ by $G(c)$. A further approximation to $G(c)$ gives:

$$d(c, M) \approx 2\frac{||z_n||}{||z'_n||}\log||z_n|| \tag{54}$$

where

$$z'_n = \frac{dz_n}{dc} \tag{55}$$

5.1.2 The Controller

Using the approximation 54 it is easy to build a program that minimizes the distance to M. First of all in order to obtain an estimate for the distance of c from M, iterate:

$$z_{k+1} = f_c(z_k) = z_k^2 + c, z_0 = 0, k = 0, 1, 2, \ldots$$

until either $||z_{k+1}|| \geq R$ (where R is large) or $k = N_{max}$, where N_{max} is the maximum number of iterations that we allow (we take $N_{max} = 1000$). If we stopped at $k = N_{max}$ we let $d(c, M) = 0$. If we have stopped at

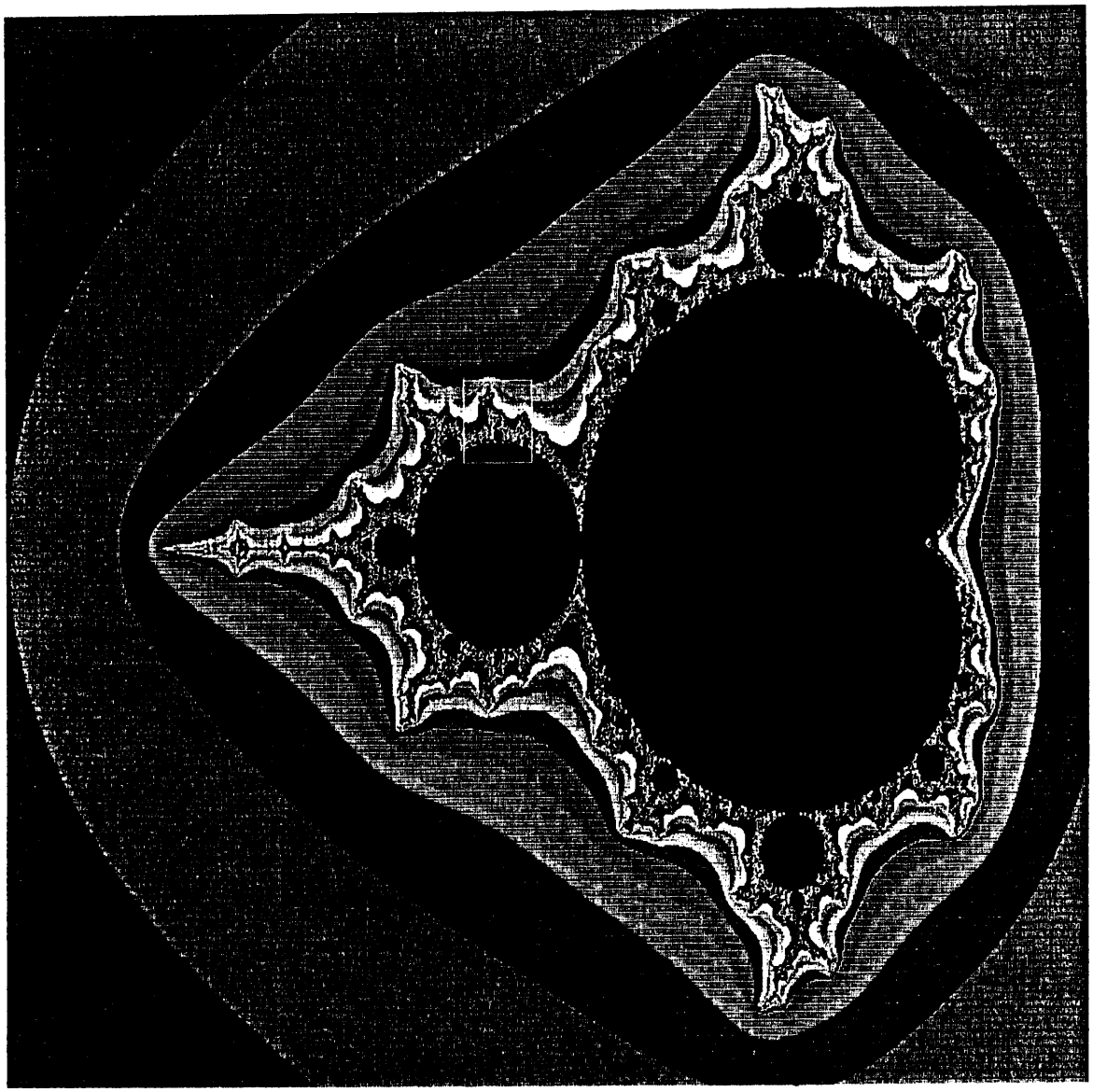


Figure 22: The Mandelbrot Set: $D = [-2.45, 0.75] \times [-1.3, 1.3]$.

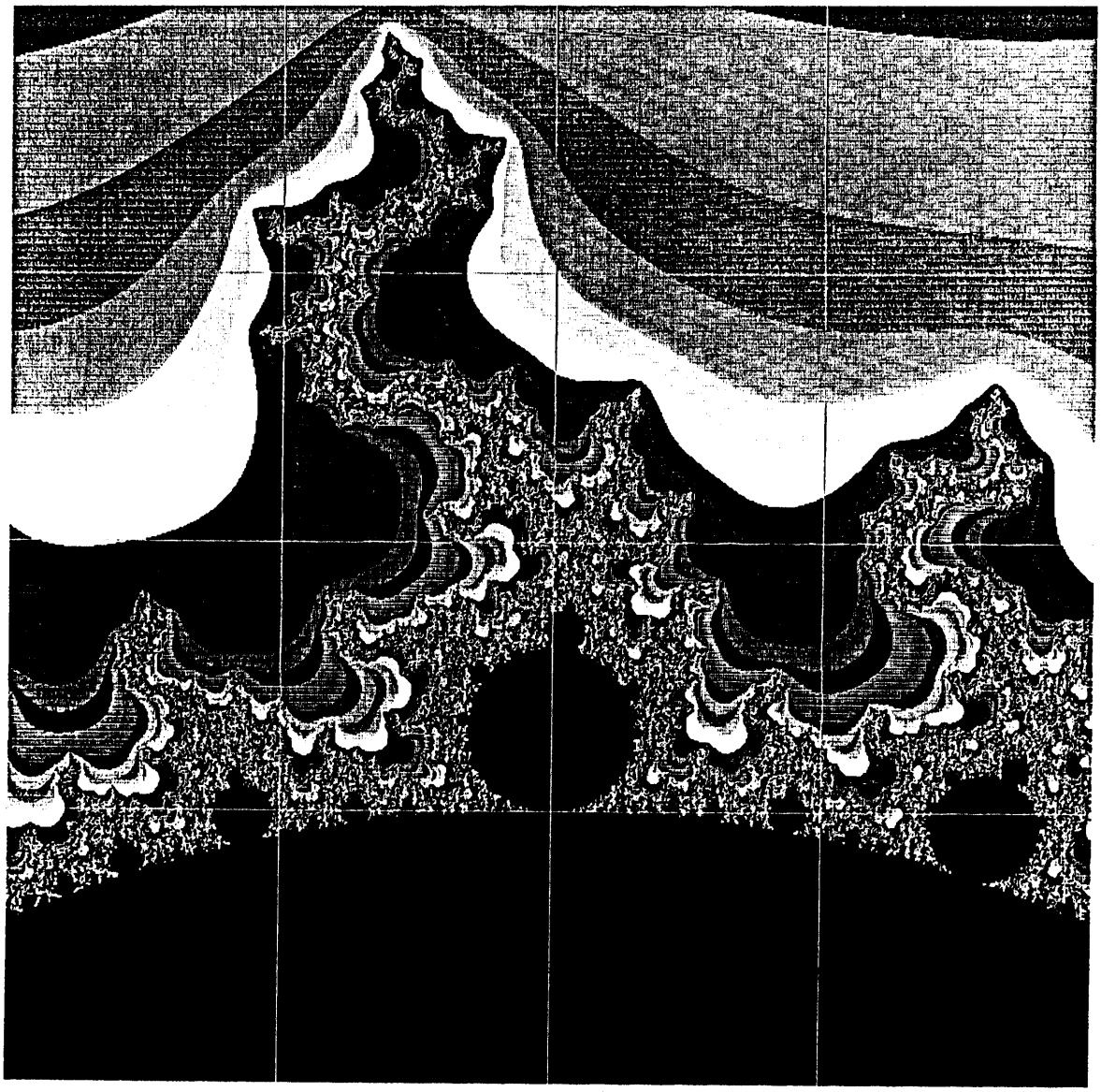


Figure 23: The Mandelbrot Set: $D = [-1.1, -0.9] \times [0.2, 0.4]$ (magnification of the window shown in figure 22).

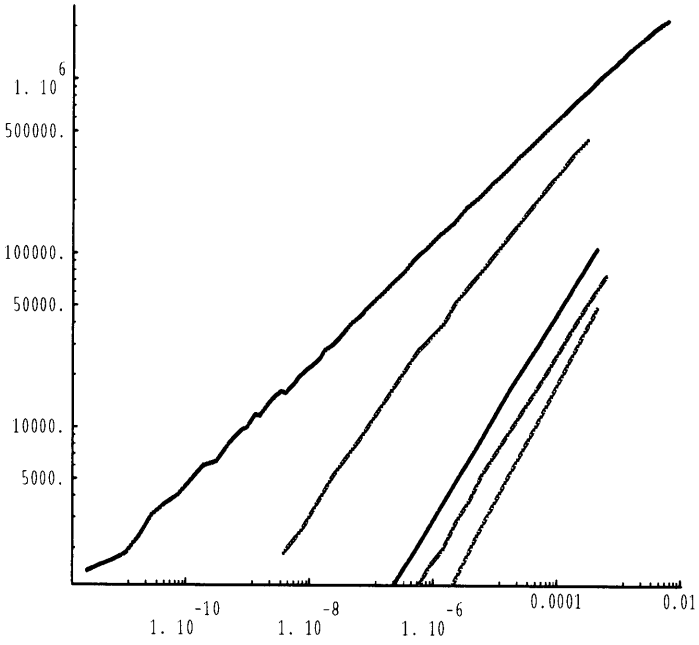


Figure 24: Plot of $\log(\delta F_{\lambda,\delta\lambda})$ vs. $\log\|\delta\lambda\|$ for the quadratic map with $c = \lambda + \xi i$, where F_λ is escape time out of a circle of radius $R = 100$, and the cell resolution is 512×512. The real global function is at $\lambda = -1$, and is measured with Gaussian observation noise with variance $\sigma_o = 10^{-3}$ and no dynamical noise ($\sigma_d = 0$). From top to bottom, the lines correspond to $\xi = 0.3$, $\xi = 0.25$, $\xi = 0.35$, $\xi = 0.2$, $\xi = 0.4$.

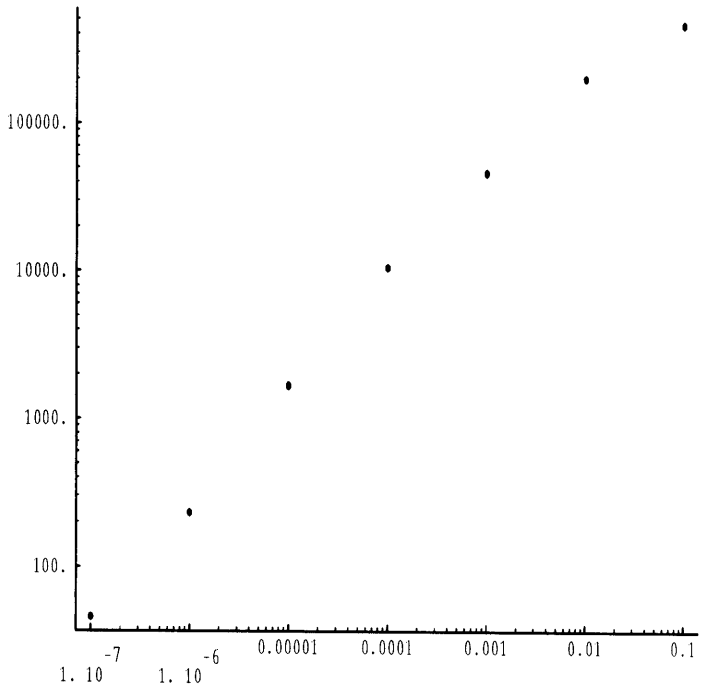


Figure 25: Plot of $\log(\delta F_{\lambda,\delta\lambda})$ vs. $\log\|\delta\lambda\|$ for the quadratic map with $c = \lambda + \xi i$, $\xi = 0$, where F_λ is escape time out of a circle of radius $R = 100$, and the cell resolution is 256×256. The real global function is at $\lambda = -0.75$, and the slope is approximately 0.687.

$n < N_{max}$, then having saved the orbit $\{z_0, \ldots, z_n\}$ we compute:

$$z'_{k+1} = 2z_k z'_k + 1, z'_0 = 0, k = 0, 1, \ldots, n-1$$

If we get an overflow in the course of this iteration, then this means that c is very close to M and we return $d(c, M) = -1$. Otherwise, we return

$$d(c, M) = 2\frac{\|z_n\|}{\|z'_n\|}\log\|z_n\|$$

A simple program implementing this is shown in figure 27.

The optimal control value ξ is the value for which the distance estimate $d(\lambda + \xi i, M)$ is minimized. Figure 28 shows a listing of the very simple program that achieves the value of `q` which minimizes the estimate

```
(MSetDist (make-rectangular p q) maxiter R
          overflow)
```

for given values of `maxiter, R, overflow`.

We first fix a value of ξ and obtain an estimate for λ to feed the function `optimal-control` (figure 5.1.2) by running a local algorithm (for example an extended Kalman filter). The local algorithm (implemented by Elmer Hung [17]) that we have used gave an accuracy of $10^{-5} - 10^{-6}$. The function `optimal-control` will return a value `q` for ξ whose distance from the trully optimal value will be of the order of $10^{-5} - 10^{-6}$. We can then drive ξ to the estimate `q` and run the global

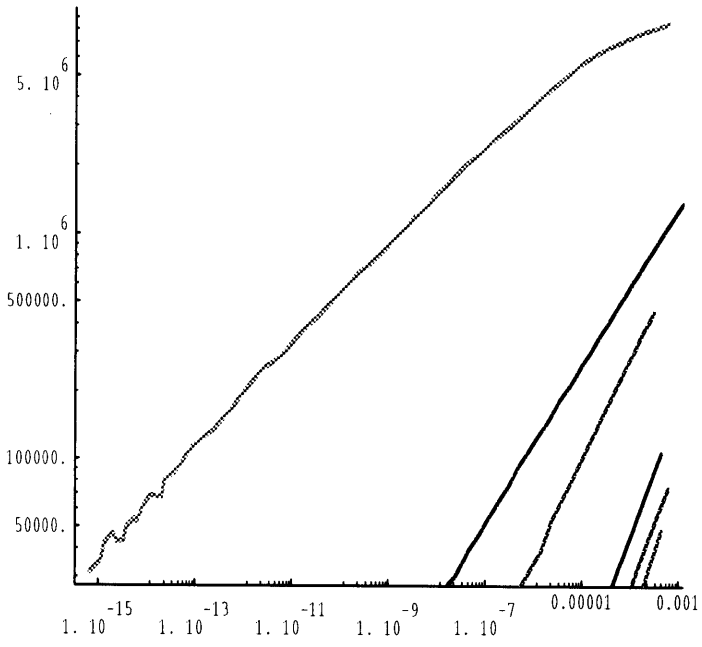


Figure 26: Plot of $\log(\delta F_{\lambda,\delta\lambda})$ vs. $\log\|\delta\lambda\|$ for the quadratic map with $c = \lambda + \xi i$, where F_λ is escape time out of a circle of radius $R = 100$, and the cell resolution is 512×512. The real global function is at $\lambda = -1$, and is measured with Gaussian observation noise with variance $\sigma_o = 10^{-3}$ and no dynamical noise ($\sigma_d = 0$). From top to bottom, the lines correspond to $\xi = 0.28651237$ ($d \approx 0.218$) $\xi = 0.3$ ($d \approx 0.35$), $\xi = 0.25$ ($d \approx 0.48$), $\xi = 0.35$ ($d \approx 0.59$), $\xi = 0.2$ ($d \approx 0.58$), $\xi = 0.4$ ($d \approx 0.68$).

```
(define (MSetDist c maxiter R overflow)
  (let ((zorbit (make-vector (+ max-
iter 1) 0.0)))
    (define (orbit-loop i z)
      (cond ((> i maxiter)
             0.0)
            ((> (magnitude z) R)
             (der-loop 1 i 0.0+0.0i))
            (else
             (let ((newz (+ (* z z) c)))
               (vector-set! zorbit i newz)
               (orbit-loop (+ i 1) newz)))))
    (define (der-loop i iter zder)
      (let ((z (vector-ref zorbit (- i 1))))
        (cond ((> (magnitude zder) overflow)
               -1)
              ((= i iter)
               (let ((m (magnitude z)))
                 (/ (* 2 (* m (log m)))
                    (magnitude zder))))
              (else
               (der-
loop (+ i 1) iter (+ (* 2 (* z zder)) 1))))))
    (orbit-loop 0 0.0+0.0i)))
```

Figure 27: A program estimating the distance of a point c in the parameter plane from M.

algorithm. After having obtained a better estimate for p ($= \lambda$) from the global algorithm, we may feed that estimate into `optimal-control` to get a better estimate for the optimal control value and repeat the whole process, until we get satisfactory accuracy.

6 Cooperation between local and global approaches

We know that the dissimilarity function is locally unimodal but we have to have a way of entering the region of unimodality in order for the global algorithm to work.

One approach is to look at the domain D at different cell resolutions. In other words increase the number of cells used to represent the global function. As we get to coarser and coarser resolutions, the region of unimodality starts at bigger and bigger values. A more interesting solution to this problem is to use a combination of local and global approaches. In particular, we can *use the local algorithms to obtain an estimate of the parameter that places us in a region of unimodality of the dissimilarity function, and then use a global approach to zero in to the correct parameter value.* Numerical experiments seem to indicate that local methods we have tried, reach rather quickly an good estimate of the parameter but then get stuck and do not seem to improve further (probably because of numerical problems, invalid linearization assumptions, or bad finite approximation assumptions). The global algorithm, given the estimate obtained by the local method, increases it by many orders of magnitude. It seems that a cooperation between local and global approaches is the right way to attack real estimation problems.

```
(define (optimal-control p tol q1 q2)
  (define (loop q1 q2 d1 d2
    (let* ((q (/ (+ q1 q2) 2))
           (c (make-rectangular p q))
           (d (msetdist c 1000 1000 1e100)))
      (cond ((< d1 tol)
             q1)
            ((= d 0.0)
             (loop q1 q
                   d1 d))
            (else
             (loop q q2
                   d d2)))))
  (loop q1 q2
        (msetdist (make-
rectangular p q1) 1000 1000 1e100)
        (msetdist (make-
rectangular p q2) 1000 1000 1e100)))
```

Figure 28: A program returning an estimate of an optimal control value ξ, given an estimate p for λ and two points q1 not in M, and q2 inside M.

7 Final Comments

In this paper, we propose a new approach to parameter estimation based on exploiting the global geometrical complexity of nonlinear dynamical systems, rather than trying to do local approximations as the classical algorithms do.

We demonstrate the power of a global approach in the context of complex analytic dynamics. Under very reasonable magnification and noise assumptions, and with a careful combination of global estimation and control, we reach an improvement as big as 13 orders of magnitude over the initial estimate.

The approach which we follow in the case of complex analytic dynamics can be extended to much more general settings.

We remark that the choice for global functions that we have made is just one of the many possibilities. We have just given one **implementation of the much more general idea of global parameter estimation**. In different settings, we are forced to chose different global functions or minimization algorithms. For example, in the case of Hamiltonian dynamical systems no attractors exist and completely different global functions must be devised (for example functions that reflect the shape of chaotic layers).

The global approach is computationally much more demanding than the local approaches but can be much more accurate and can have more robust convergence properties. With the use of increasingly faster computers the disadvantage mentioned above becomes less and less important. Moreover, the global approach is **ideally suited for parallel computation**, opening the door to tremendous icrease in performance.

There are many new directions to follow: first of all the power law that the dissimilarity function locally obeys needs to be understood thoroughly and analyzed theoretically. Many interesting open questions concerning the nature and behavior of the parameter sensitivity exponent γ also arise. To our knowledge these results are completely new.

Moreover, we believe that it would be interesting to develop a general theory of optimal global geometric parameter estimation through control, and investigate how p.s.e. changes with the position of the parameter in parameter space.

The ultimate goal is to use chaos and instability and combine the local and global approaches to parameter estimation in order to obtain breakthrough extraordinarily precise measurements of quantities that are very difficult to measure, such as the Universal Constant of Gravity.

Acknowledgements

I would like to thank first of all Hal Abelson and Gerry Sussman for their help, support, and encouragement. I would like to point out that the idea for a global approach to parameter estimation was born through conversations with Prof. Sussman, who was the first person that motivated me to work in this direction.

I would also like to thank John Tsitsiklis, Steve Strogatz, Elmer Hung, and George Verghese for many very useful discussions, as well as all the members of Project MAC at the AI lab for tolerating my using all the machines in the group for the necessary computations.

Appendix

A The Golden Section Method

The Golden Section Method uses the fact that we can bracket the location of the minimum of a unimodal function by evaluating the function at two distinct points in the region L of unimodality.

To describe how it works, we first assume that the function φ_λ is unimodal in the interval $L_1 = [l_1, r_1]$. Suppose we evaluate the function at two points x_1, x_2 in L such that $x_1 < x_2$ and find that $f(x_1) < f(x_2)$. It follows from the definition of unimodality that $\lambda \in [l_1, x_2]$. Similarly, if $f(x_1) > f(x_2)$, then we must have $\lambda \in [x_1, r_1]$. If the function values at x_1, x_2 are equal then $\lambda \in [x_1, x_2]$, but for simplicity we may again consider that it belongs to any one of the above bigger intervals. In any case, after the first two function evaluations, a portion of L_1 to the right of x_2 or to the left of x_1 can be eliminated from further search. If $L_2 = [l_2, r_2]$ is the remaining interval, we can obtain two more function evaluations and further reduce the length of the interval containing λ. By using this procedure we can keep reducing the unimodality interval, obtaining an increasingly tighter bracketing of the minimum value.

An improved version of the above naive algorithm is the *Fibonacci method*, which gets its name from using the Fibonacci sequence

(56)
$$\mathcal{F}_0 = 0, \quad \mathcal{F}_1 = 1, \quad \mathcal{F}_k = \mathcal{F}_{k-1} + \mathcal{F}_{k-2}, \quad k = 2, 3, \ldots$$

in picking the points at which the function is evaluated. The method works as follows: Let N be the total number of points at which the function will be evaluated. Suppose that at iteration k, the interval containing λ (the

local minimum) is $[l_k, r_k]$. For $k = 1, 2, \ldots, N-1$, the function values are computed at the two points

$$x_1^k = l_k + \frac{\mathcal{F}_{N-k}}{\mathcal{F}_{N+2-k}}(r_k - l_k) \tag{57}$$

$$x_2^k = l_k + \frac{\mathcal{F}_{N+1-k}}{\mathcal{F}_{N+2-k}}(r_k - l_k) \tag{58}$$

We notice that (by the definition of the Fibonacci sequence) one of the points x_1^k, x_2^k is the same as one of the points at a previous iteration. Hence, only one new function evaluation is required at each point. This is extremely important in our case where function evaluations are computationally very expensive.

One of the disadvantages of the Fibonacci method is that the number of function evaluations N must be known in advance. Getting rid of this requirement leads to a method known as the *Golden section method*, which is a good approximation to the Fibonacci search. It can be shown that

$$\lim_{N\to\infty} \frac{\mathcal{F}_{N-1}}{\mathcal{F}_N} = \frac{1}{\tau} = \frac{\sqrt{5}-1}{2} \tag{59}$$

The golden section method then places the points at which the function is to be evaluated at

$$x_1^k = l_k + \frac{\tau - 1}{\tau}(r_k - l_k) \tag{60}$$

$$x_2^k = l_k + \frac{1}{\tau}(r_k - l_k) \tag{61}$$

Again, only one function evaluation is required.

References

[1] H. Abelson, G. J. Sussman, with J. Sussman, *"Structure and Interpretation of Computer Programs"*, The MIT Electrical Engineering and Computer Science Series, MIT Press, McGraw-Hill, 1985.

[2] L. V. Ahlfors, *Complex Analysis*, Third Edition, McGraw-Hill, 1979.

[3] K. T. Alligood, J. A. Yorke, "Fractal Basin Boundaries and Chaotic Attractors", Proceedings of Symposia in Applied Mathematics, Vol. 39, 1989.

[4] M. Aoki, *"Optimization of Stochastic Systems"*, Academic Press, 1967.

[5] M. Avriel, *"Nonlinear Programming: Analysis and Methods"*, Prentice-Hall Series in Automatic Computation, 1976.

[6] P. Blanchard, "Complex Analytic Dynamics on the Riemann Sphere", *Bulletin of the American Mathematical Society*, Vol. 11, Number 1, 1984.

[7] M. L. Blank, "Stochastic Attractors and their Small Perturbations", *Mathematical Problems of Statistical Mechanics and Dynamics*, pp. 161-197, Reidel Pub. Co., 1986.

[8] B. Branner, "The Mandelbrot Set", Proceedings of Symposia in Applied Math., Volume 39, 1989.

[9] H. Brolin, "Invariant Sets under Iteration of Rational Functions", *Ark. Mat.*, 6, pp. 103-144, 1965.

[10] A. Cayley, "The Newton-Fourier Imaginary Problem", Amer. J. Math., 2, p. 97, 1879.

[11] A. Douady, J. H. Hubbard, "Iteration des polynomes quadratiques complexes", CRAS Paris 294, pp. 123-126, 1982.

[12] R. O. Duda, P. E. Hart, *"Pattern Classification and Scene Analysis"*, J. Wiley and Sons.

[13] C. Grebogi, E. Ott, J. A. Yorke, "Crises, sudden changes in Chaotic Attractors, and Transient Chaos", *Physica 7D*, p. 181-200, 1983.

[14] C. Grebogi, E. Ott, J. A. Yorke, "Fractal basin boundaries, long-lived chaotic and unstable-unstable pair bifurcations", *Phys. Rev. Letters*, **50**, p.935-938, 1983.

[15] K. F. Gauss, *"Theory of Motion of the Heavenly Bodies"*, New York, Dover Publ., 1963.

[16] J. Guckenheimer, P. Holmes, *"Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields"*, Applied Mathematical Sciences, Vol. 42, Springer-Verlag, 1983.

[17] E. S. Hung, *"Dynamics Enhanced Parameter Estimation in Chaotic Systems"*, Massachusetts Institute of Technology, Bachelor of Science Thesis, 1991.

[18] E. Atlee Jackson, *Perspective of Nonlinear Dynamics*, Vol. 2, Cambridge University Press, 1989.

[19] A. H. Jazwinski, *"Stochastic Processes and Filtering Theory"*, Academic Press, 1970.

[20] R. E. Kalman, "A new approach to linear filtering and prediction problems", *J. Basic Eng.*, vol. 82D, pp.35-45, Mar. 1960.

[21] J. L. Kaplan, J. Mallet-Paret, J. A. Yorke, "The Lyapounov dimension of a nowhere differentiable attracting torus", *Ergodic Theory and Dynamical Systems*, **4**, p. 261-281, 1984.

[22] L. Keen, "Julia Sets", Proceedings of Symposia in Applied Math., Volume 39, 1989.

[23] J. Kennedy, J. A. Yorke, "Basins of Wada", *Physica D*, 51, p. 213-225 (1991).

[24] A. J. Lichtenberg, M. A. Lieberman, *"Regular and Stochastic Motion"*, Applied Mathematical Sciences, Vol. 38, Springer-Verlag, 1983.

[25] J. Milnor, "Self-similarity and hairiness in the Mandelbrot set", *Institute for Advanced Study*, preprint.

[26] H. O. Peitgen, P. H. Richter, *The Beauty of Fractals*, Springer-Verlag, 1986.

[27] H. O. Peitgen, D. Saupe eds., *The Science of Fractal Images*, Springer-Verlag, 1988.

[28] H. L. Royden, *"Real Analysis"*, Second Edition, Macmillan Press, 1968.

[29] D. Ruelle, *"Elements of Differentiable Dynamics and Bifurcation Theory"*, Academic Press, 1989.

[30] C. L. Siegel, "Iteration of Analytic Functions", *Ann. of Math.*, (2) 43, pp. 607-612, 1942.

[31] H. W. Sorenson, *"Kalman Filtering: Theory and Application"*, IEEE Press, 1985.

[32] H. W. Sorenson, "Least-squares estimation: from Gauss to Kalman", *IEEE Spectrum*, vol. 7, pp. 63-68, 1970.

[33] D. Sullivan, "Quasiconformal homeomorphisms and dynamics I", *Ann. Math.* 122, pp. 401-418, 1985.