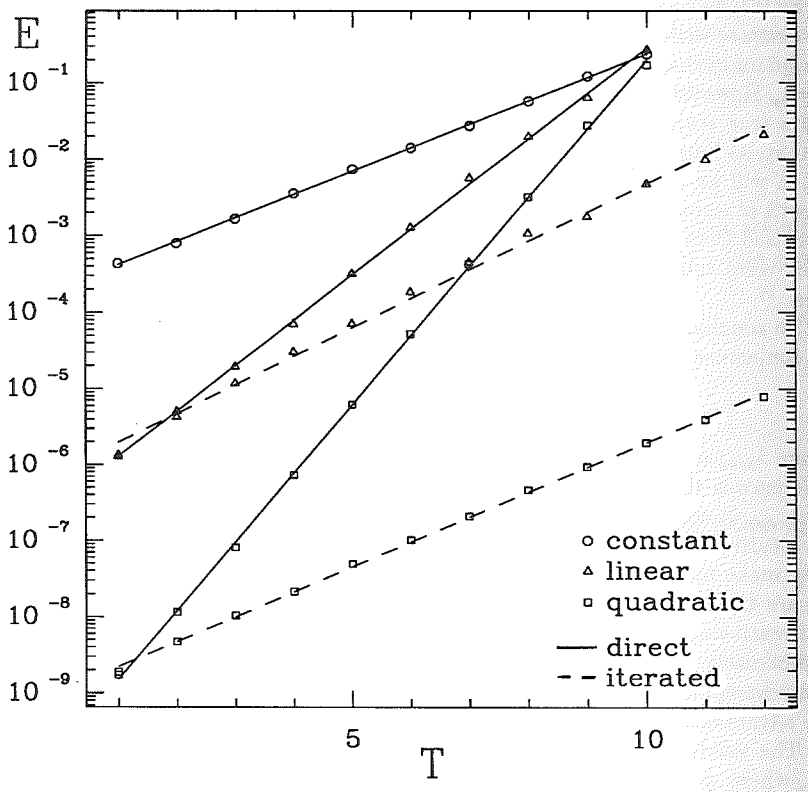


Figure 6: Forecasts made using local polynomial approximation, building the data base from a 5,000 point time series generated by Equation (18). We make 500 forecasts, and plot the average error as a function of the extrapolation time $T$. Letting $q = m + 1$, where $m$ is the order of the polynomial, as expected from Equations (34) and (45), the logarithm of the error for direct forecasts grows roughly according to $q\lambda$ whereas for iterative forecasts it grows according to $\lambda$, independent of the order of approximation. $\lambda$ is the Lyapunov exponent, which in this case is one bit per iteration.



will not perform as well. In fact, as discussed in Section 3.4, our numerical experiments indicate that iterated forecasts have long tails, but that direct forecasts produce tails that are even longer.

We wish to emphasize that the above results are for scaling in the limit as $\bar{E} \to 0$. In situations where the model is not good enough to achieve this limit, these scaling laws cannot be expected to hold.

### 3.2.4 Temporal scaling with noise

So far in this section we have assumed that the accuracy of predictions is limited by the quality of the approximation, which for a given data set is determined by the number of data points. In other cases the accuracy of prediction may be limited by noise. Even if we know the equations of motion exactly, noise limits prediction by making initial conditions uncertain, and by perturbing deterministic trajectories. The results of the previous section make it clear that the effect of noise is very much like the effect of approximation error. In fact, if we let $\delta$ in Equation (35) represent noise rather than approximation error, the results of that section go through essentially unchanged. Thus the scaling with noise is described by Equations (41) or (44), if we substitute $\langle |n(t)| \rangle$ for $\langle |\delta(x)| \rangle$.

## 3.3 Continuous time

So far our error estimates have been for discrete-time maps, where each iteration of the map causes a large change in the state. Two problems occur in continuous-time systems.

The first problem has to do with the meaning of $N$. In a continuous time system, the number of points can be increased by simply decreasing the sampling rate $\Delta t$. If $\Delta t$ is already small this may not give us any new information. In this case the amount of useful data depends more on the number of characteristic time scales rather than on the number of data points. Thus, for a continuous time system the estimates should be stated in terms of the number of characteristic times, rather than the number of data points.

Unfortunately, the notion of a characteristic time scale is ambiguous. It can be estimated in many ways, for example as the correlation time, as the inverse of the average frequency of the power spectrum, or as the average time between level crossings. Yet another way to define a characteristic time comes from trajectory segmenting; the average ratio of the number of trajectory segments to the number of data points inside local neighborhoods estimates the extent to which the data is oversampled. Note that although this ambiguity about the meaning of $N$ introduces some absolute uncertainty into the error estimates, it does not effect the scaling for any *fixed* $\Delta t$.

The second problem occurs in iterative forecasting. Although the conclusion that iterative forecasts are superior carries over to continuous time systems, it raises the question of how to pick the composition time. Clearly the effects of noise and finite

precision dictate that this time should not be zero. Thus, there is some optimal time for constructing an approximation, which should minimize the errors in iterative forecasts. This is one possible source of the discrepancies observed in the following section. A similar problem occurs for approximation of differentials (Equation (8)): There are many possible ways to approximate the derivative – in the presence of noise it is important to weight contributions from different times properly. These questions will be treated in more detail in a future paper.

## 3.4 Numerical results

In this section we present some numerical results, investigating the scaling properties for systems that are not simple maps. So far we have mainly studied the Mackey-Glass delay differential equation (19) and the chaotic convection data of Haucke and Ecke [43].

Figure (7) shows the results of making forecasts at several different extrapolation times for the Mackey-Glass delay differential equation with $\Gamma = 17$, where the dimension of the underlying attractor is roughly $D = 2.1$ [26]. We used a a four dimensional time delay embedding, with $\tau = 6$, using only 500 data points sampled at $\Delta t = 1$. We get the best results using approximation by quadratic polynomials, and choosing twice the minimum number of nearest neighbors, which in this case is $M = 30$. As seen in the figure, iterated forecasting is superior to direct forecasting. Using least squares to fit a line to the indicated portions of the curve we get good agreement with the known value of the largest Lyapunov exponent [26]. However, we do not understand why the initial rise is larger than this, although we have observed this behavior in several other continuous data sets.

In Figure (8) we show a similar plot, except that the number of points in the database is increased to 10,000. This improves the accuracy of the forecasts substantially, roughly as expected from Equation (16). However, when we attempt to compare the slope with the previous case, we observe something puzzling: Even in the straight portions of the curve, at the right of the figure, the slope is now larger, by a factor of roughly 1.3. This may be related to the same effect mentioned above, and needs further investigation.

Our numerical experiments indicate that while our forecasts with these methods are good on average, there are large fluctuations in accuracy, with occasional very bad predictions. The distribution of errors about the mean can be quite large, as demonstrated in the error histograms of Figure (9). As we see, the error distribution has long tails. (On this scale this is only apparent for long time forecasts.) As discussed in the previous section, this is expected for iterated forecasts. However, the tails for the direct forecasts are even longer than those of the iterated forecasts. We have found that this depends on several factors, such as the proper selection of neighborhoods. If $x$ is not enclosed by its neighborhood, for example, we get very bad predictions. The frequency of bad predictions goes up when we use fixed disjoint partitions instead of nearest neighborhoods. There may also be other factors that cause bad predictions, for example discontinuities, singular behavior, neighborhoods



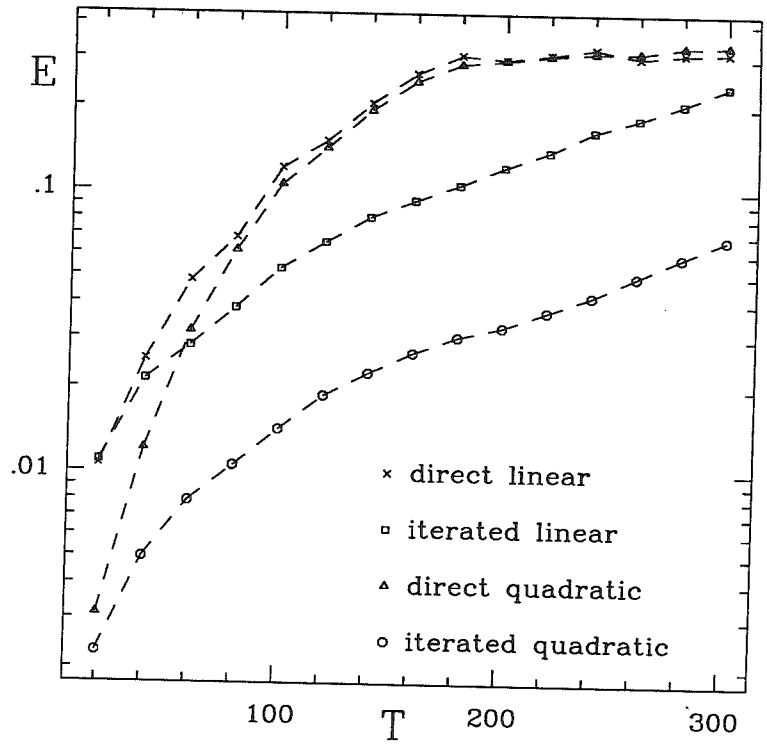


Figure 7: Prediction error for the Mackey-Glass delay differential equation as a function of extrapolation time, with $N = 500$ (roughly 10 characteristic times), $\Gamma = 17$, $\tau = 6$, $\Delta t = 1$, and $d = 4$, using nearest neighborhoods with twice the minimum number of points needed to fit the coefficients by least squares ($M = 30$ for quadratic, and $M = 10$ for linear). The iterated forecasts are made by iterating the model at the first extrapolation time shown. The fractal dimension of the underlying attractor is $D = 2.1$. To improve the statistical stability of our error estimates we have computed $\bar{E}$ by throwing out the 10% worst predictions.

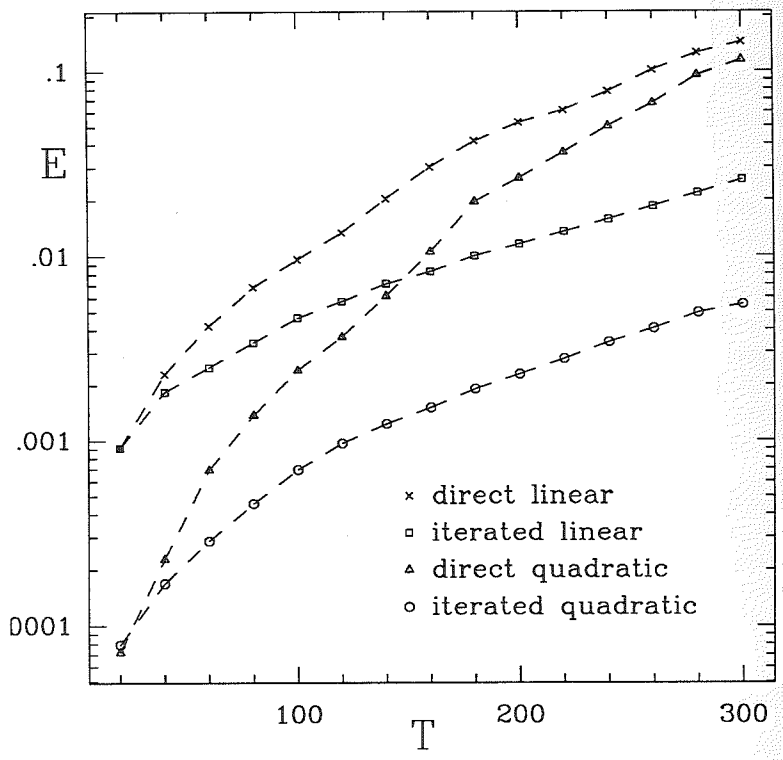


Figure 8: Same as Figure (7), except that $N = 10,000$.



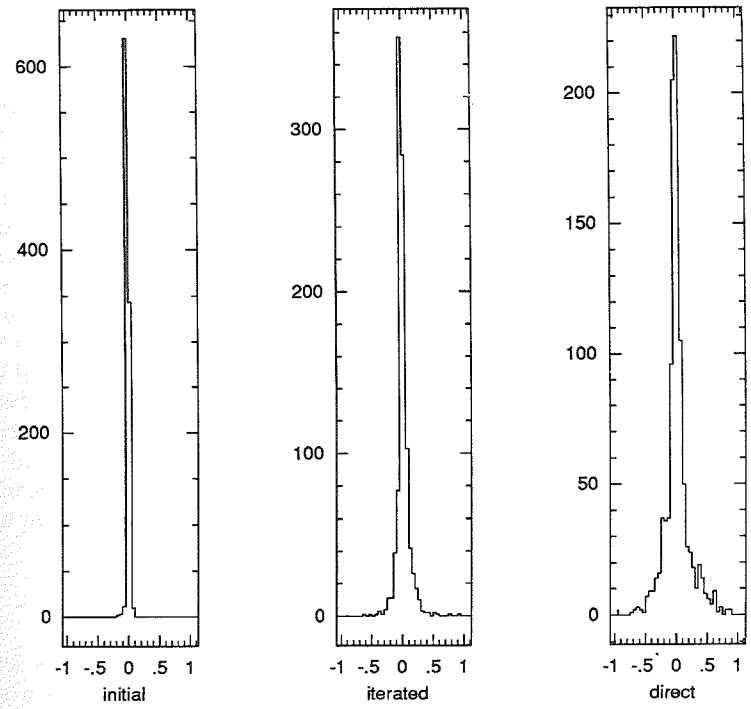


Figure 9: *Error histograms* for the data of Figure (7). The $y$ axis shows the number of counts with a given value of the prediction error $E = x(t+T) - \hat{x}(t,T)$. The histogram on the left corresponds to short-term forecasts with $T = 20$. The middle histogram corresponds to iterated forecasts at $T = 100$, using a composition time of 20. The histogram on the right corresponds to direct forecasts, also at $T = 100$.

that for some reason or another do not make a stable fit for the parameters of their chart, or dynamical factors such as nonuniformities of the derivatives. We intend to investigate this in more detail. It should be possible to reduce bad predictions by using appropriate diagnostics and tailoring the fitting strategy for each individual neighborhood. Also, it is obviously desirable to be able to make *a priori* confidence estimates for each prediction.

Occasional bad forecasts can cause large fluctuations in the mean prediction error, so that unless we average over a large number we do not get stable results. To reduce the data requirements, for Figures (7) and (8) we have computed the mean prediction error by rejecting the 10% of the predictions with the largest errors.

Note that even when we include the worst predictions, our results for the Mackey-Glass equation using a 500 point database are equivalent to or better than those of a feed-forward neural net on the same data set [49]. If we increase the data base to 10,000 points, we see an improvement of roughly $20^{\frac{3}{2.1}} \approx 80$ for short time forecasts with quadratic polynomial charts. Using a database this large for the neural net employed by Lapedes and Farber is currently computationally intractable, even using supercomputers. In contrast, to generate one point of Figure (8), which involves fitting 1000 local charts, consumes roughly five minutes on a SUN 3/60 microcomputer.

We now return to the convection data analyzed in Figure (1). Based on our previous results, and the scaling results of Section 3.2.3, we originally assumed that we would be able to improve these results by using iterated rather than direct forecasts. We were quite surprised to discover that iterated forecasts are actually worse than direct forecasts in this case, as shown in Figure (10). We do not understand at this time why iterated forecasting is inferior in this case, or whether we could change this by altering the parameters of our model. So far, this is the only case where we have found direct forecasting is superior.

## 3.5 Is there an optimal approach?

These estimates make it clear that the primary limitation on short term forecasts comes from the dimension. If the dimension is too high, then the nearest neighbors of a given point may be so distant that the dynamics is poorly approximated even with a data base consisting of thousands of points. Is this an intrinsic limit, or is it possible to do better?

In fact, there is an optimal method, namely, to guess the right answer. With prior information, or a sufficiently exhaustive search, we may be able to find a model that is superior to those we would be led to by blind analysis with the techniques that we have described so far. For example, suppose a chaotic time series is produced by a differential delay equation, of the form

$$\frac{dx}{dt} = f(x(t), x(t-\Gamma)) \tag{46}$$

As demonstrated in reference [24], when $\Gamma$ is large equations of this form can have attractors of very large dimension. However, suppose we know in advance that the



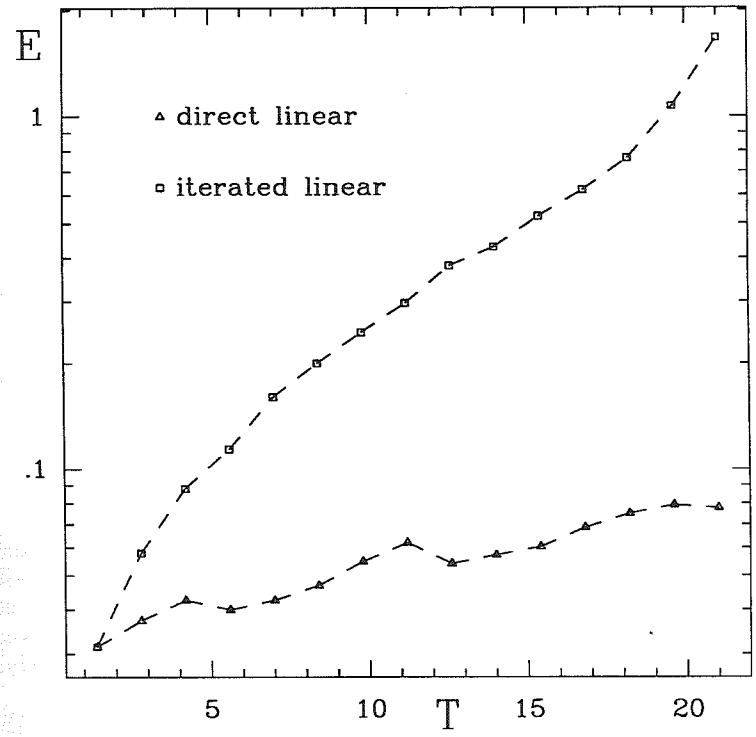


Figure 10: A comparison of iterated and direct forecasts using iterated and direct forecasting, using the same data shown in Figure (1). For reasons that we do not understand, direct forecasting is superior in this case.

dynamics is this form. If we formulate our model *as a delay equation*, the problem becomes two dimensional, irrespective of the dimension of the attractor. This obviously makes it much simpler. The error estimates we have given here apply, but $D$ is two, irrespective of the attractor dimension. The key to this is algorithmic simplicity; if we can find a way to formulate a good model in algorithmically simple terms, even apparently complex behavior may be quite tractable. When many aspects of the model are known *a priori*, this becomes a standard problem in parameter estimation [22].

# 4 Experimental Data Analysis

With an accurate model of the dynamics, experimental data can be examined with all the tools that are currently available only to the numerical analyist. We can use the model to compute statistical quantities such the Lyapunov exponents or the fractal dimension, or even to do more exotic things such as computing unstable cycles or estimating bifurcation points. Furthermore, with a model we can make use of smoothness to probe the structure on a scale smaller than the typical separation between points. Thus, with a given number of points it is possible to achieve more accuracy than with straightforward techniques.

We begin the discussion with a brief review of previous work on computing fractal dimension, and then discuss new approaches in the following section.

## 4.1 Computing fractal dimension: A review

Efficient methods for computing fractal dimension have been very important in the search for chaotic behavior. The most popular current method is due to Grassberger and Procaccia [36,38]. Their method is one in a class of techniques that we will refer to as *ball scaling* methods [57].[15] The basic idea is to estimate a quantity such as the information, at different scales of resolution. Usually this is done by counting the number of points inside balls of different radii $\epsilon$. The scaling exponent yields a dimension. For a recent review see reference [74].

A common misconception about the effectiveness of the Grassberger and Procaccia technique for computing dimension is that it depends on the correlation dimension. Ball scaling can equally well be applied to compute information dimension or fractal dimension [57,74]. The underlying reason for the superiority of ball scaling is the fact that it gives better estimates of probability distributions and their moments, as explained by Omohundro[16] [58].

[15] As far as we know the use of ball scaling to compute dimension is originally due to Pettis *et al.* (1979). Ball scaling methods independently arose from the work of several different groups, including Takens [72,73], Grassberger and Procaccia [36,38], Guckenheimer [40], and Farmer and Jen [9]. Since Grassberger and Procaccia were the first to develop and demonstrate the effectiveness of these procedures, the use of ball scaling is usually attributed to them.

[16] See page 301.



Numerical dimension computations based on ball scaling are subject to misinterpretation and must be used with care. Misapplication of the ball scaling technique has led to many false statements about the presence of chaos.

Several promising new methods to compute dimension may help solve this problem. For example, if the Broomhead and King embedding procedure is applied *locally*, to the points in a ball of radius $\epsilon$, the embedding dimension computed from the singular values gives a good upper bound on the fractal dimension [10]. If the ball is small enough, curvature due to global structure is negligible, since inside the ball the dynamics is locally linear. The principal value decomposition automatically yields the *local* embedding dimension, which is an upper bound on the fractal dimension. Examination of the scaling with the size of balls provides a self-consistency check on the results.

Another method to compute the local embedding dimension has recently been suggested independently by Cremers and Hübler [16] and by Cenys and Pyragas [13]. This approach is similar to a method previously suggested by Packard *et al.* [59], but they examine the *scaling* of the width of conditional probability distributions with ball size and embedding dimension. If the embedding dimension is large enough the width of the distribution narrows sharply. Furthermore, if the embedding dimension is sufficiently large the width of the distribution scales linearly with $\epsilon$, providing a self-consistency check. Yet another approach has been suggested by Bayly *et al.* [3], who propose fitting rational polynomials of different dimension, to find the dimension with minimum variance. Their method of computing dimension is analogous to that of Froehling *et al.* [31], except that they use global polynomials instead of local linear fits, which seems to produce much better results. We need further work comparing these methods to determine if they are more reliable than ball-scaling methods. Although all of these methods compute the embedding dimension rather than the fractal dimension, in practice this is often just as good.

## 4.2 More accurate data analysis with higher order approximation

Computations based on counting points cannot probe the dynamics to scales of resolution that are less than the typical separation distance between points. This necessarily limits the accuracy of these methods, particularly for small numbers of points in high dimensions. The accuracy of an estimate of the fractal dimension $D$ based on $N$ data points scales roughly as $N^{-\frac{1}{D}}$ [31].[17] Thus, in the language we have developed here, conventional ball scaling methods have the scaling properties of first order approximation.

With higher order approximation schemes it is possible to probe the dynamics to scales that are smaller than the typical separation between points. If our model is suf-

[17] Although this estimate was originally stated for box counting methods, it also holds for ball scaling. The difference between them comes from the constant in front, plus the difference in the demand they place on computer resources.