

Model-Building with Interpolated Temporal Data

R I (Bob) McKay*, Hoang Tuan Hao*, Naoki Mori *†, Nguyen Xuan Hoai*,
and Daryl Essam*

*School of Information Technology and Electrical Engineering,
University of New South Wales @ Australian Defence Force Academy,
ACT 2600 Australia

†School of Computer Science, Osaka Prefecture University, Osaka, Japan

Corresponding Author:

R I McKay

Tel: +61 2 429 363 229

Fax: +61 2 6268 8581

160 Naylor Rd, Urila, NSW 2620 Australia (address for correspondence and
proofs)

b.mckay@adfa.edu.au and urilabob@hotmail.com

Abstract

Ecological data can be difficult to collect, and as a result, some important temporal ecological datasets contain irregularly sampled data. Since many temporal modelling techniques require regularly spaced data, one common approach is to linearly interpolate the data, and build a model from the interpolated data. However this process introduces an unquantified risk that the data is overfitted to the interpolated (and hence more typical) instances. Using one such irregularly-sampled dataset, the Lake Kasumigaura algal dataset, we compare models built on the original sample data, and on the interpolated data, to evaluate the risk of mis-fitting based on the interpolated data.

Keywords

Linear Interpolation, Modelling, Genetic Programming

Section 1 Introduction

Ecological modelling is crucial for gaining an understanding of ecosystems and permitting their effective management. Modelling techniques build best-fit models

based on available data, but in order to determine best-fit, must make assumptions about the data. For many ecological problems, data assembly is a difficult task, beset with practical issues which preclude perfect data. These issues arise particularly in time-series modelling, where it may be difficult to ensure that data are sampled on a regular basis (or even, that different attributes are sampled on equivalent time scales).

One common approach to this problem is to interpolate the data. The interpolation methods used range from linear interpolation, through smoothing approaches such as spline interpolation, to more sophisticated statistical model-building approaches. However this method is fraught with danger if the interpolated data is then used for model-building. In this circumstance, we are in effect building one model (the more sophisticated model built by the underlying modelling technique) on top of another (the usually simpler model used for interpolation). Since modelling methods are essentially looking for regularities in the data, there is a risk that the regularities detected may simply be those induced by the interpolation model, rather than real regularities in the data. The model used for interpolation may generate biases in the interpolated data sufficient to mislead the primary modelling technique.

In this paper, we will work with a well-studied phytoplankton dataset, describing water quality in Lake Kasumigaura in South-Eastern Japan (Recknagel et al, 1998). This dataset is well-known to be irregularly sampled and interpolated (Whigham, pers. comm.) We will use a modelling method close to those previously used on the dataset, namely Grammar-Guided Genetic Programming (GGGP – Whigham 1995) to generate a difference equation model. We first recover the original, irregularly sampled, dataset. We compare the predictive accuracy of models built using the original, irregularly sampled, dataset, and those built using interpolated data. This will allow us to assess the effect of interpolation on the quality of the models generated.

The rest of the paper is organised as follows. In section 2, we discuss the background, including the underlying domain (phytoplankton), the model building approach (GGGP for learning difference equations), and interpolation methods. Section 3 briefly presents a mathematical and experimental study of the effects of linear interpolation on variance (one of the important biases introduced by linear interpolation). We describe our overall method in section 4, and the detailed experiments in section 5. Section 6 presents the results and discussion, and we conclude with section 7.

Section 2 Background

2.1 Phytoplankton in Lake Kasumigaura

Phytoplankton are microscopic photosynthesising organisms, primarily from several groups of algae and bacteria. A number of species (e.g. *Microcystis*, *Oscillatoria*) can occasionally exhibit periods of superabundance (blooms) with consequent harmful ecological and economic effects (Reynolds 1984), so prediction of their abundance, and especially of blooms, is of considerable importance.

Phytoplankton population dynamics are affected by a wide range of endogenous variables, including physical factors such as light and temperature, chemical factors such as pH and the levels of nitrogen and phosphorus, and biological factors such as the level of grazing by zooplankton. While there has been considerable previous work on developing predictive models, there is still room for improvement in the quality and reliability of the models.

Lake Kasumigaura is a large shallow lake in South-Eastern Japan, about 70km NE of Tokyo. At the time of dataset collection, in the 1980s and 1990s, there was high nutrient runoff into the lake, and hence high nutrient loadings. Consequently, there was also a high phytoplankton abundance, with periodic blooms. There is considerable seasonal fluctuation in temperature and light loadings, resulting in a large seasonal component to the phytoplankton levels, measured by chlorophyll A readings (figure 1). Lake Kasumigaura is an important source of water for the urban areas of Tokyo and Tsukuba, as well as for agriculture and for a significant aquaculture industry, so that water quality is of critical economic, as well as ecological, importance.

The Lake Kasumigaura dataset contains an extensive range of ecological variables sampled over a ten-year period 1984-1993 (Recknagel et al, *ibid*). The data availability varies over the attributes, so in our work we have restricted our attention to eight variables, shown in Table 1.

The data, as distributed, is daily data. However analysis shows that it has in fact been sampled at a lower rate and then linearly interpolated. Only one variable, the light level, has been sampled on a daily basis, most of the other variables being sampled on approximately a monthly basis (29.96 ± 3.65 days), except for a short period toward the end where they were sampled fortnightly. A second variable, the

Copepoda density, has been sampled on a weekly basis in some periods, but the sampling dates include the dates on which the rest of the variables were sampled. Finally, the sampling dates for water temperature were also approximately monthly, but on clearly different days from those on which the rest of the data were sampled over about half the period of data collection. Fortunately, water temperature is one of the slowest and most regularly changing variables (see figure 2), so that the linear interpolation used is likely to provide realistic estimates of the actual values.

2.2 Modelling

Time-series models are used for three principal purposes

1. To predict future events before they occur (often, to permit remedial action)
2. To permit the investigation of possible scenarios, often as a result of management changes
3. To provide an explanation of the data

In this paper, we will primarily focus on the first two purposes, though they are closely related to the last and we believe our discussion is relevant to it as well. In both these cases, the aim is to use a model built from known data to predict the behaviour of the modelled system in unseen circumstances – ie to interpolate or extrapolate the data. The quality of the model determines the quality of the predicted value, but is itself determined by the quality of the underlying data from which the model was learnt. Hence it is clearly risky, even if often necessary, to use data interpolated from a simple model as the basis for learning a more complex model. Our aim is to investigate the extent of this risk for a specific set of modelling data.

A wide variety of error measures are available for assessing the quality of models (and hence, to guide the computerised search for good models). For scenario modelling, Root Mean Square Error (RMSE) is the most widely accepted, and that is what we use in this paper. There is some question whether RMSE is the most suitable error measure for the use of time series models for future prediction (notably, because it is time-symmetric whereas it is arguable that error measures for time-series prediction should not be: for example, a model which anticipates an algal bloom too early may be more valuable than one which predicts its occurrence too late). Nevertheless, RMSE is widely used in predictive use of time-series models, hence our discussion is relevant to this purpose at least; equally important, our analysis does not rely essentially on RMSE, so the results may well extend to other error measures.

In describing a model-inference system, we need to describe two components, the class of models explored, and the algorithm used to search amongst them. Here, we follow the lead of Whigham and Recknagel (1999) in using GGGP to generate difference equation models. The learning approach differs in some details, but the broad approach is similar.

2.2.1 Model Space

Whigham and Recknagel discuss a number of representations in the context of equal time differences, the most general being a simple first-order difference equation,

$$1) \quad y_{t+1} = f(\underline{x}_t, y_t)$$

However for irregularly-sampled data, it is essential to incorporate the time difference into the equation, the simplest approach being to incorporate it in differential form, i.e.

$$2) \quad \delta y / \delta t = f(\underline{x}_t, y_t)$$

which may be re-written as

$$3) \quad y_{t+1} = y_t + (\delta t * f(\underline{x}_t, y_t))$$

The function to be learnt is $f()$.

While the equation 3 form is crucial for our experiments, permitting a direct comparison between irregularly-sampled and interpolated data, it is important to note that it also has disadvantages. In particular, in this form, it is not possible to directly impose the learning constraint used by Whigham and Recknagel, namely that y_t must be positive (of course the training data still reflects this property, so that the learnt models may also incorporate it, but it cannot be readily imposed as a learning constraint).

2.2.2 Grammar-Guided Genetic Programming

Genetic Programming (GP – Cramer 1985, Koza 1992) searches a space of function representations using evolutionary methods; the grammar-guided variant relies on a context-free grammar to restrict the search space to a readily-defined subset of functions. It has previously been used in a number of ecological modelling problems (greater gliders – Whigham 2000; plankton – Whigham and Recknagel *ibid*, bandicoots – Shan et al 2002).

2.3 Interpolation

It is frequently the case that ecological time-series data must be interpolated before modelling. This commonly arises because the different attributes have been sampled on different dates, but can also be a consequence of the use of irregularly-sampled data in combination with a modelling method which requires regular time-interval data. The interpolation methods used fall into three broad classes: linear interpolation, spline interpolation and statistical interpolation.

2.3.1 Linear Interpolation

Linear interpolation is the simplest of the commonly-used methods. Its primary advantage is this simplicity. Its primary disadvantage is the biases that it imports, most importantly in reducing the variance of the data. We discuss this issue in section 3. Since linear interpolation is widely used, it is the primary focus of this study.

2.3.2 Spline Interpolation

Spline interpolation fits a more complex curve to the data, smoothing the curvature at the data points. The simplest approach, and probably the most widely used, fits piecewise cubic functions to the data (cubic splines). The exact biases vary, depending both on the function-set used for fitting, and the error minimisation procedure, but since the primary aim of spline interpolation is to smooth the data near the data points, there is some argument that spline interpolation methods will lead to over-emphasis on the regions near the data points in subsequent modelling.

2.3.3 More Sophisticated Interpolation Techniques

This study concentrates on simple interpolation methods (i.e. linear- and cubic-spline- interpolation) because they are widely used, and because their biases are understandable. As a result, their interactions with subsequent learning methods are relatively comprehensible. The literature contains an enormous variety of more sophisticated approaches, shading imperceptibly from interpolation into more complex modelling. Two principal directions are detectable: geostatistical methods deriving from kriging (Krige 1951); and moving average methods stemming from Box's (1976) Auto-Regressive Moving Average (ARMA) approach. As Rice (2004) points out, there is generally a close correspondence between stochastic modelling and spline-based smoothing approaches. However the key issue in using these more

complex interpolation methods to generate data for learning techniques is the complexity of building one modelling technique on the data supplied by another. There is a risk that the learning system may simply be learning the biases of the interpolation mechanism, rather than the regularities in the raw data; the more complex the underlying model of the interpolation method, the more difficult it will be to detect this effect.

Section 3 Linear Interpolation and Variance

As an example of the sort of bias which may be introduced by interpolation, we consider the effect of linear interpolation on variance. We mentioned previously that linear interpolation induces a bias in the interpolated data set, and that one important – and undesirable – aspect of this bias is a reduction in the variance below that of the original dataset; that is, the interpolated data will (probabilistically) have a lower variance than that of the original dataset. We are able to prove:

Theorem 1: Given a sequence value distribution $\mathbf{D}_y(n): n = 1, \dots, \mathbf{N}$, and a time interval distribution $\mathbf{D}_\delta(n): n=1, \dots, \mathbf{N}$, and under the reasonable assumption that \mathbf{D}_y and \mathbf{D}_δ are independent, if we linearly interpolate y to obtain a new sequence value distribution $\mathbf{D}'_y(m): m=1, \dots, \mathbf{M}$, the expected standard deviation decreases, ie $\mathbf{E}(\sigma(\mathbf{D}'_y)) < \mathbf{E}(\sigma(\mathbf{D}_y))$.

Corollary 1: in the limit, if y and δ are bounded as $\mathbf{N} \rightarrow \infty$, then this is true of any sample as well (that is, for large enough \mathbf{N} , and for any sample of data points, linear interpolation will reduce the standard deviation).

To evaluate the practical consequences, we carried out a number of preliminary experiments, described below.

3.1 Experimental Analysis

We conducted sampling from reasonable distributions as in Theorem 1 above, to evaluate how often the variance would fail to decrease on linear interpolation. We used five sample sizes for \mathbf{N} , $\mathbf{N} \in \{10, 100, 200, 500, 1000\}$. We used two different distributions for \mathbf{D}_y , a Gaussian with $\mu=0.5$ and $\sigma=0.5$, and a uniform distribution over the range 0..1. For \mathbf{D}_δ , we used uniform distribution with four different ranges: $\{1..1, 1..5, 1..25, 1..125\}$. Thus there were 40 treatments in all. Each treatment was run 1000 times, and we recorded the number of instances where the standard deviation

$E(\sigma(\mathbf{D}'_y))$ was not less than $E(\sigma(\mathbf{D}_y))$, as recorded in Table 2. From the table, it is clear that, even for quite small samples (ie $N=10$), the variance almost always decreases on linear interpolation, and for larger sample sizes, it effectively always decreases (the case where the number of insertions is exactly 1 is a special case; here it is provable that there will be a decrease in variance unless the variance is already zero).

Section 4 Method

The results of section 3 confirm that we should be concerned about the biases induced by interpolated data. Since the use of interpolated data is often unavoidable, it is important to investigate the scale of the effect on data modelling. In particular, it could be argued that the costs of sampling imply that ecological datasets are frequently relatively small, so that stochastic noise in the data might swamp the bias effects resulting from interpolation. In these respects, the Lake Kasumigaura data is fairly typical of the scale of a class of human-sampled datasets which arise relatively commonly in ecological modelling, hence it would be useful to investigate the importance of the bias induced by interpolation.

4.1 Data Preparation

4.1.1 Data Cleaning

As previously mentioned, we first extracted the original underlying data from the dataset (assuming linear interpolation). It was clear that the data had been sampled on an approximately monthly basis, but that there were additional observations for one variable (light level) throughout the data, and additional observations for all variables in the more recent period. We discarded these additional observations to obtain a dataset with irregular but approximately equal sampling (29.96 ± 3.65 days).

One variable, water temperature, causes a more difficult (but perhaps more typical) problem: it has been sampled on roughly the same time scale as the other variables, but (in some parts of the dataset) on different dates. Our solution was straightforward: we used the linearly interpolated values for this variable. Water temperature changes relatively slowly (Fig. 1) so the interpolated value is arguably a good surrogate for the real value in this case. However it is certainly not perfect: the mean and standard deviation changed from $16.20 \pm 8.14^\circ\text{C}$ (original data) to

16.50±7.82°C (interpolated data). To this limited extent, the interpolation process may have reduced the credibility of our modelling of Lake Kasumigaura. Recalling that the primary purpose in this paper is not to model Lake Kasumigaura, but to investigate the effects of interpolation on the modelling process, it is perhaps more useful to view the new dataset as a different dataset problem, still quite typical of ecological modelling problems, if no longer perfectly reflecting the water-temperature behaviour of Lake Kasumigaura.

4.1.2 Data Interpolation

The dataset constructed to this point makes a good baseline for comparison with modelling from interpolated data. It will be referred to in the rest of this paper as the “original data”. From this dataset, with 120 data points, we constructed two preliminary datasets, interpolating back to the apparent original sampling rate of the Kasumigaura dataset, ie daily, using linear and cubic spline interpolation. We then constructed a further three datasets, typical of interpolated data that might be used for modelling, by sampling from these daily datasets. One was sampled from the linearly-interpolated daily data at 7-day intervals (labelled “weekly”), and another at the mean sampling rate of the original data, i.e. 29.96 days (rounded to the nearest day – labelled “monthly”). The third was sampled at 7-day intervals from the spline dataset (labelled “spline”). Table 3 shows the mean and standard deviation of each of the variables in each dataset.

4.1.3 Data Categorisation

Each dataset was further divided into two equal portions, for training and testing the models. The first portion, covering the years 1984-1988, was used for training, while the second portion, covering the years 1989-1993, was used to test the generalisation ability of the evolved models.

4.2 Experimental System

As is usual in GGGP, the acceptable form for f in our equation 3:

$$3) \quad \delta y / \delta t = f(\underline{x}_t, y_t)$$

is defined by a context-free grammar (Table 4). The variables $p, n, s, t, l, o, co, chla$ are the corresponding attributes from Table 1, while $r1$ and $r2$ are random ephemeral constants with ranges [0.0...1.0] and [-50.0...50.0] as in Whigham and

Recknagel (ibid). In this example, the independent variables p, n, s, t, l, o, co form the vector \underline{x}_t , while y_t is the dependent variable $chla$. The function set consists of the arithmetic operators (+, -, *, /) together with the exponential function (pow), permitting the learning of very general forms for the models.

Section 5 Experiments

The primary purpose of these experiments was to compare models generated by training on the original, uninterpolated, data with models generated by the different interpolated datasets. Hence there were four primary treatments, corresponding to the four training sets: original, monthly linear interpolated, weekly linear interpolated and weekly spline interpolated, each covering the period 1984-1988. Each treatment was replicated in 30 independent runs. The evolutionary settings are detailed in Table 5.

The fitness function used was the Root Mean Square Error (RMSE) over the training cases (naturally, the evolutionary objective was to minimise the RMSE).

An important complication with the function search space used in these experiments is the potential to generate invalid numeric values, either out-of-range or undefined. Out-of-range values can be generated by arithmetic operations such as division by zero or exponentiation; fortunately, the C compiler used treats out-of-range values logically correctly (ie infinity is larger than any other number), so that these infinite values cause no problems to the evolutionary algorithm. Undefined values cause more problems. With this function set, they arise primarily from division of 0 by 0, resulting in a value of NaN (Not a Number). NaN is readily detected, since it is the only value satisfying $!(x==x)$. In these experiments, we handled NaN fitness values by resetting them to a very large (i.e. unfit) value, namely $\exp(700)$. This is potentially disruptive to the evolutionary process, if too many NaN values occur. As a precaution, we recorded the number of NaN substitutions in each run; typical values were around 5,000, ie around 1% of the total number of fitness evaluations, suggesting that NaN substitutions did not substantially affect the evolutionary process.

The models learnt were evaluated against all four training sets and all four testing sets, though the primary interest is in the performance of the models learnt from each training dataset, on the original testing dataset.

For comparison, we present in table 5 the RMSE values on these datasets for the naïve predictor, ie (ie $y_t = y_{t-1}$). We note from these results that:

- The testing set data (ie the later years) are much better predicted by the naïve predictor than the earlier years (presumably because of the smaller overall excursions)
- The smoothing effects of interpolation mean that the naïve predictor does far better on the weekly- and spline- interpolated data than it does on the original data

Section 6 Results & Discussion

6.1 Results

In these experiments, there were four different treatments (i.e. training on each of the four training sets). In each generation of the each run, we located the best individual (in terms of minimum RMSE against its own training data), and calculated its error against the four training datasets and the original testing dataset. We averaged this value over all 30 runs with the same training set. These results are presented in figures 3 to 7. In figure 8, we show the equivalent plot for the average performance of all individuals in a generation against the original testing data.

Figures 3 to 6 show the performance against the training sets. Unsurprisingly, for each training set, the best performance comes from the individuals trained against that dataset. The plots show the expected gradual improvement of performance through the course of the run, and the continuing improvement suggests that premature convergence has not been a problem in these experiments, with the possible exception of training on the spline data. We also note that the dramatic differences in figure 5 between the weekly-trained model and the other models suggest that this model is fitting regularities in the weekly data that are not present in the other datasets; similar remarks apply to the spline training in figure 6.

The more interesting results come in figures 7, showing the performance of the different treatments against the original test data. The models trained against monthly-interpolated data perform best against the testing data, outperforming even those trained against the original data, however the differences are small. The models trained against weekly-interpolated data also perform acceptably on the testing dataset, suggesting that fitting to the weekly interpolation has not caused a problem in

this instance. However the models trained from the spline-interpolated data perform poorly on the testing data, comparably with the naïve predictor.

Table 6 displays the best individual found (minimum training error), over all runs, for the treatments f_1, f_2, f_3 : original, monthly and weekly data respectively, and shows their testing error. The complex individual (273 symbols) found for the spline data, with very poor testing data performance and clearly overfitted to the training data, is omitted.

6.2 Discussion

It appears that interpolation per se does not prevent effective learning from a dataset. Rather, the bias imposed by the interpolation method appears to be the important issue. In the case of the Lake Kasumigaura dataset, linear interpolation appears not to affect the modelling results appreciably, whereas spline interpolation results in poorer modelling.

In fact, the models trained from the monthly interpolated data actually generalise marginally better to the original test data than do those trained on the original training data – perhaps the irregular intervals in the original training data generate chance regularities to which the trained models are over-fitted, while the monthly-interpolated data generalise better because this source of confusion is absent.

In passing, we note that these results are not directly comparable to those of Whigham and Recknagel (ibid): in order to obtain a suitable cleaned dataset for these experiments, we chose a slightly different subset of the available attributes, and used a different separation into training and testing periods. However the RMSE values we obtained for the most directly comparable dataset (ie the “weekly” data) are in the same ballpark as theirs, suggesting that the results are compatible.

Section 7 Conclusions and Further Work

Generalisation from a single modelling problem is fraught with risk, and it is clear that these experiments need to be repeated on a number of ecological time-series modelling problems before general conclusions can be drawn. However, the results are sufficiently clear to indicate a need for caution in building models from interpolated data. The bias of the interpolation method may impact on the quality of the models built from it.

In this experiment, at least, there is an indication that sampling interpolated data at the average sampling frequency of the underlying data may be relatively safe (presumably because the modelling process receives too small a signal from the interpolation process to be confused by it), and a hint that the interpolation process may even be beneficial for generalisation.

Future work will broaden these results in two directions:

- Repeating the experiments with other interpolation methods, to assess their effects.
- Repeating the experiments on a range of ecological time-series modelling problems, to confirm their breadth of generalisation

Section 8 Acknowledgements

We would like to thank Prof. F. Recknagel of Adelaide University for his assistance in providing access to the data. We would also like to acknowledge his help, along with Dr P. Whigham of the University of Otago, in gaining an understanding of the structure of the dataset. Finally, we would like to acknowledge the assistance of Mr Y. Shan, of the University of New South Wales, through his insight into modelling issues arising from the modelling problem.

References

Box, G E P and Jenkins, G M, 1976. Time Series Analysis: Forecasting and Control, Holden-Day, 1976

Cramer, N L 1985. A representation for the Adaptive Generation of Simple Sequential Programs. Proceedings of the International Conference on Genetic Algorithms and Applications, pp. 183-187

Koza, J R 1992. Genetic Programming: on the Programming of Computers by means of Natural Selection. MIT Press

Krige, D G, 1951. A Statistical Approach to some Mine Valuation and Allied Problems at the Witwatersrand. Masters thesis, University of Witwatersrand, South Africa

Recknagel, F, Fukushima, T, Hanazato, T, Takamura, N and Wilson, H, 1998, Modelling and Prediction of Phyto- and Zooplankton Dynamics in Lake Kasumigaura by Artificial Neural Networks. Lakes and Reservoirs: Research and Management 3: 123-133

Reynolds, C 1984. The Ecology of Freshwater Plankton. Cambridge University Press.

Rice, J A 2004. Functional and Longitudinal Data Analysis: Perspectives on Smoothing. Statistica Sinica 14: 613-629

Shan, Y, McKay, R I and Paull D 2002. Building Ecological Models Using Genetic Programming. Proceedings of the fourth Asia-Pacific Conference on Simulated Evolution and Learning, IEEE, Singapore, 320 - 325

Whigham, P A 1995. Grammatically-biased Genetic Programming. In , J Rosca (ed.) Proceedings of the 1995 Workshop on Genetic Programming, Morgan-Kaufmann, 33-41

Whigham, P A and Recknagel, F (1999). Predictive Modelling of Plankton Dynamics in Freshwater Lakes using Genetic Programming. In Oxel, L and Scrimgeour, F (eds) Proceedings of the International Congress on Modelling and Simulation, 679-684

Whigham, P A 2000. Induction of a Marsupial Density Model using Genetic Programming and Spatial Relationships. Ecological Modelling 131: 299-317

Variable	Mean±Standard Deviation	Units
Ortho Phosphate (p)	15.46±32.11	mg/l
Nitrate (n)	517.17±525.10	µg/l
Secchi Depth (s)	84.72±47.15	cm
Water Temperature (t)	16.50±7.82	°C
Light (l)	1199.16±695.55	MJ/m ²
Dissolved Oxygen (o)	11.13±2.41	-
Copepoda (co)	160.36±96.73	Inds/l
Chlorophyll-A (chla)	74.35±46.60	µg/l

	Interpolations $\mathbf{D}_{\delta t}$	1	1-5	1-25	1-125
Distribution	Samples \mathbf{N}				
Gaussian	10	0/1000	1/1000	9/1000	10/1000
	100	0/1000	0/1000	0/1000	0/1000
	200	0/1000	0/1000	0/1000	0/1000
	500	0/1000	0/1000	0/1000	0/1000
	1000	0/1000	0/1000	0/1000	0/1000
Uniform	10	0/1000	4/1000	4/1000	10/1000
	100	0/1000	0/1000	0/1000	0/1000
	200	0/1000	0/1000	0/1000	0/1000
	500	0/1000	0/1000	0/1000	0/1000
	1000	0/1000	0/1000	0/1000	0/1000

	Original	Monthly	Weekly	Spline
Ortho Phosphate (p)	15.35±32.00	16.54±32.87	15.56±29.27	15.58±31.86
Nitrate (n)	519.88±523.72	499.97±494.66	512.73±496.91	515.59±516.96
Secchi Depth (s)	85.26±47.32	85.80±48.31	85.28±44.67	85.13±46.49
Water Temperature (t)	16.40±7.87	16.30±7.45	16.44±7.56	16.36±7.75
Light (l)	1197.75±692.79	1186.65±682.61	1200.00±614.71	1194.63±688.03
Dissolved Oxygen (o)	11.12±2.40	10.92±2.41	11.09±2.17	11.09±2.34
Copepoda (co)	160.54±96.34	157.98±98.82	159.24±81.56	158.45±89.91
Chlorophyll-A (chla)	73.92±46.65	73.06±44.61	74.22±42.14	74.00±45.01

$S \rightarrow T$

$T \rightarrow T \text{ OP } T$

$T \rightarrow T \text{ pow } r1$

$T \rightarrow p|n|s|t||o|co|chla|r2$

$\text{OP} \rightarrow + | - | * | /$

Runs per treatment	30
Population size	1000
Generations per run	51
Probability of crossover	0.9
Probability of mutation	0.1
Selection Tournament Size	3
Max depth (initial generation)	6
Max depth (later generations)	10

	Training	Test
Original	60.15	37.58
Monthly	55.43	36.78
Weekly	31.56	20.04
Spline	15.18	9.60

$$f_1 \quad [s+p*(do/chla)^{0.049878} - chla] / [s+(chla^{0.069428}/do)]$$

or approximately

$$f_1 \quad [s+p - chla] / [s+(1/do)]$$

RMSE on testing data 35.44

$$f_2 \quad [-(chla+1)/48.928529]+do^{0.286526}$$

RMSE on testing data 36.21

$$f_3 \quad t^{0.695560} / \{$$

$$[t^{0.024622} * n^{0.975378} * (chla+t+1)]$$

$$/ [(chla+t)^{0.024622} * c^{0.044035} * do^{0.936691} * (t^{0.476382} - n)]]$$

$$+ [c + (c+t)^{0.543833} + (p/o)^{0.051561}]^{0.051561}$$

$$+ ((n+24.147064) * (t+c+(t^{0.695560}/p)))^{0.733995}]$$

$$/s\}$$

or approximately

$$f_3 \quad (t^{0.695560} * s) /$$

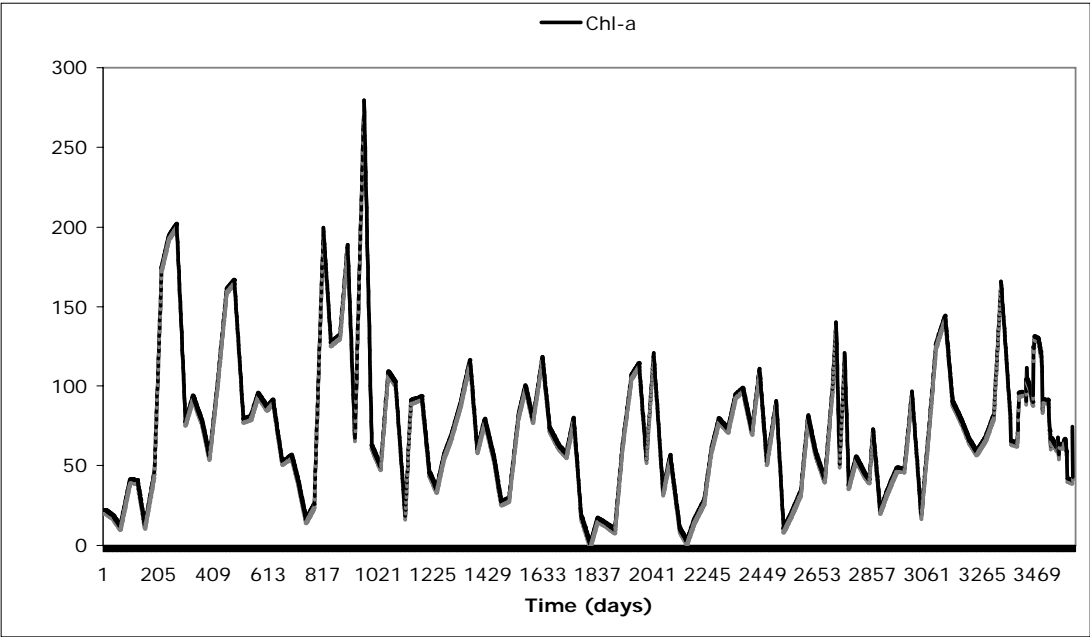
$$(s * n * (c+t+1) / o * (t^{0.476382} - n))$$

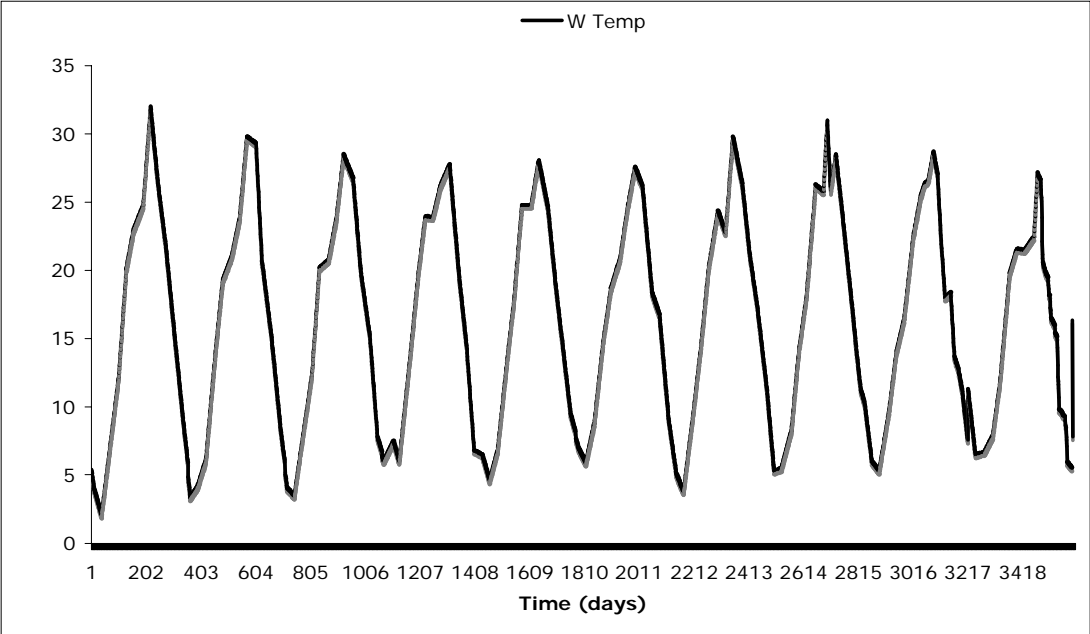
$$+ 1 + ((n+24.147064) * (t+c+(t^{0.695560}/p)))^{0.733995})$$

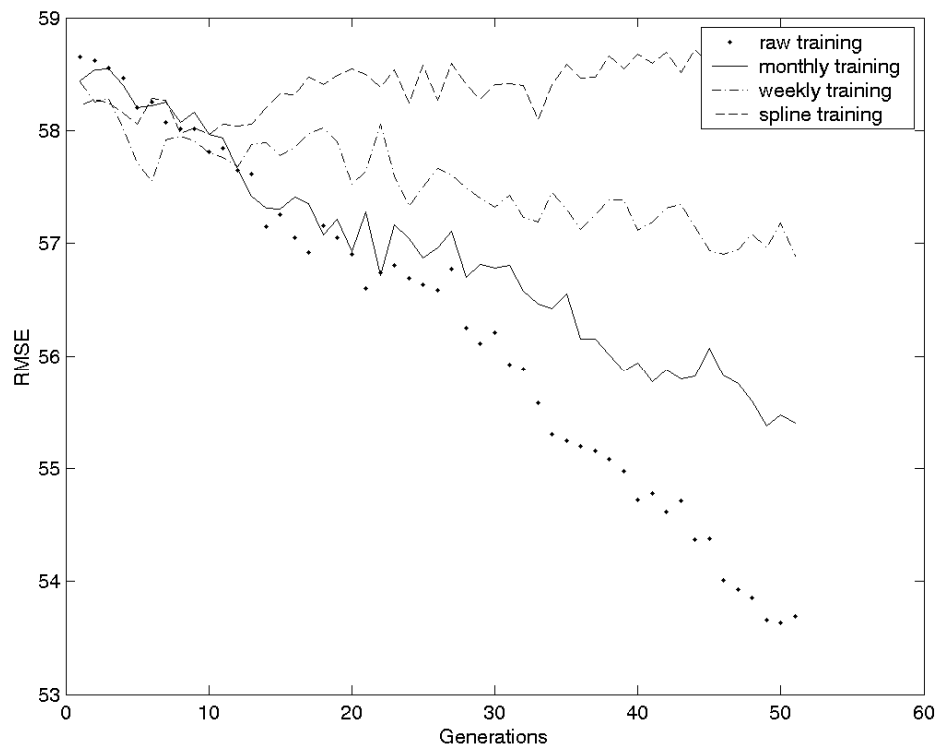
RMSE on testing data 37.53185242

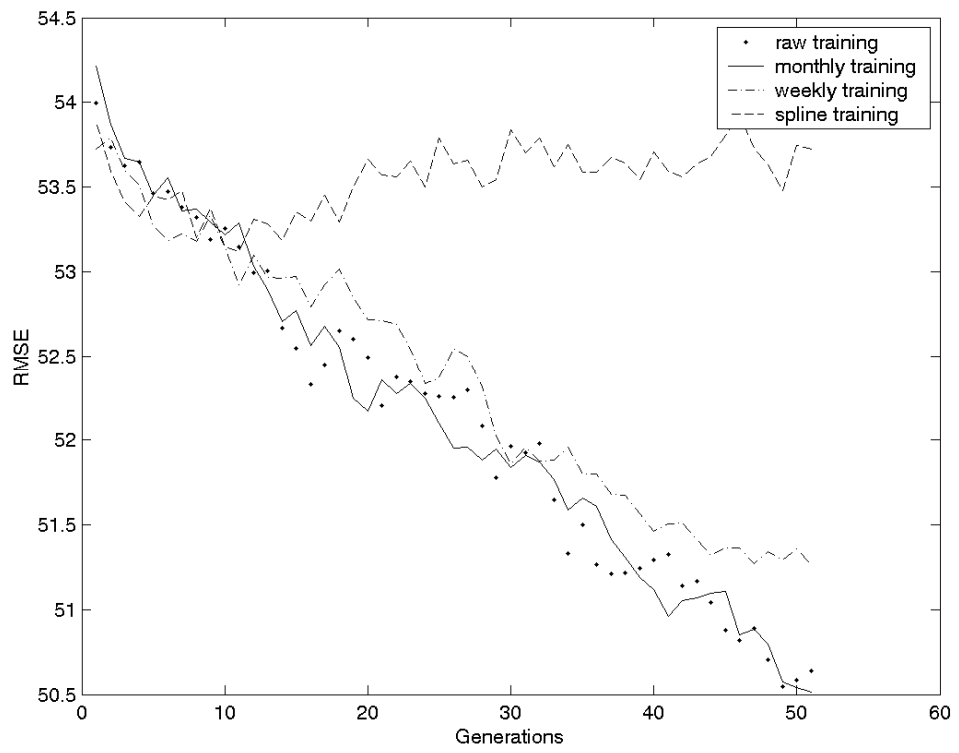
$$f_4 \quad 273 \text{ symbols}$$

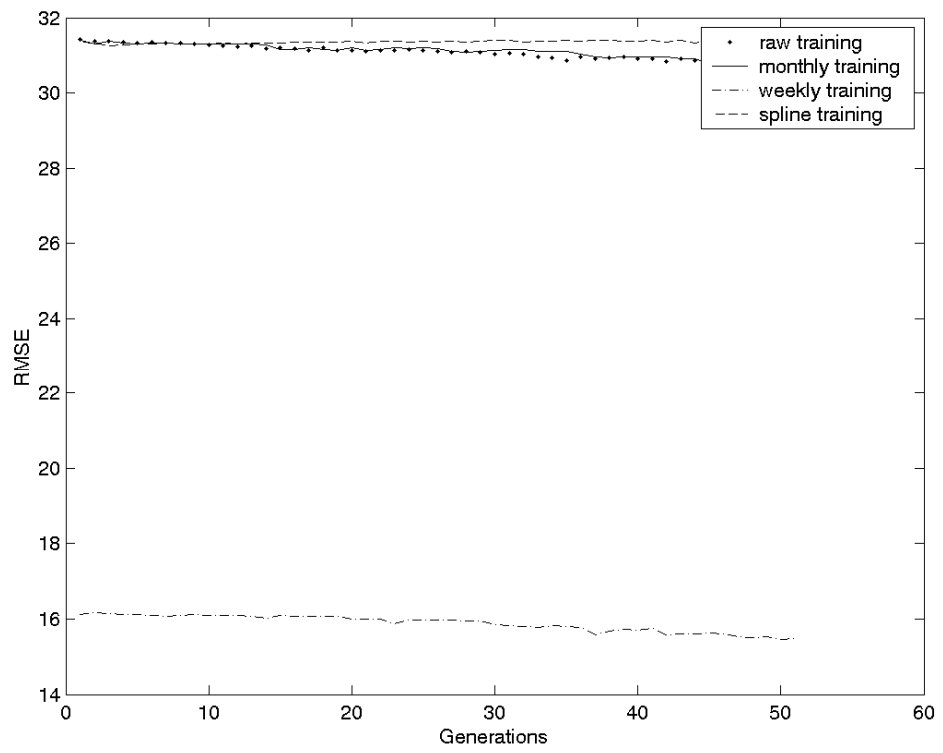
RMSE on testing data 144.5473054

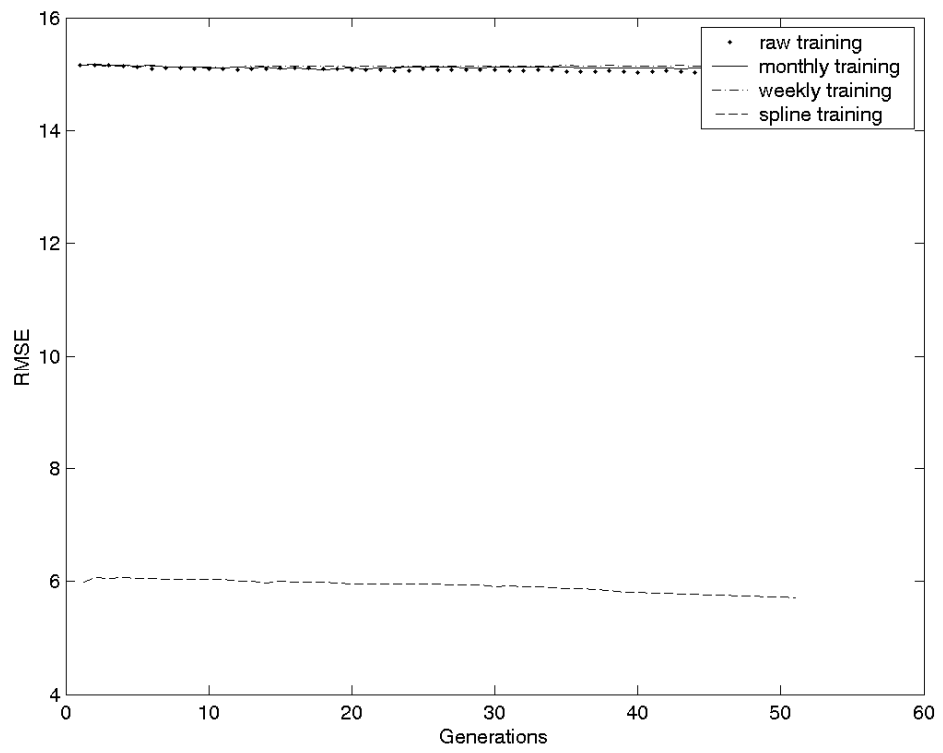


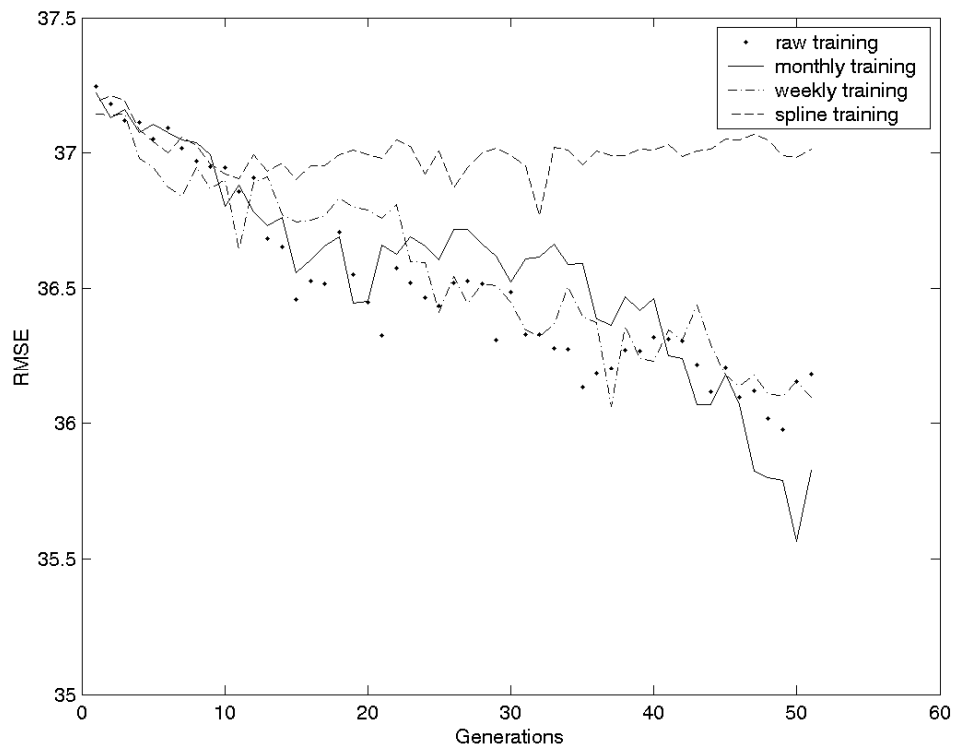












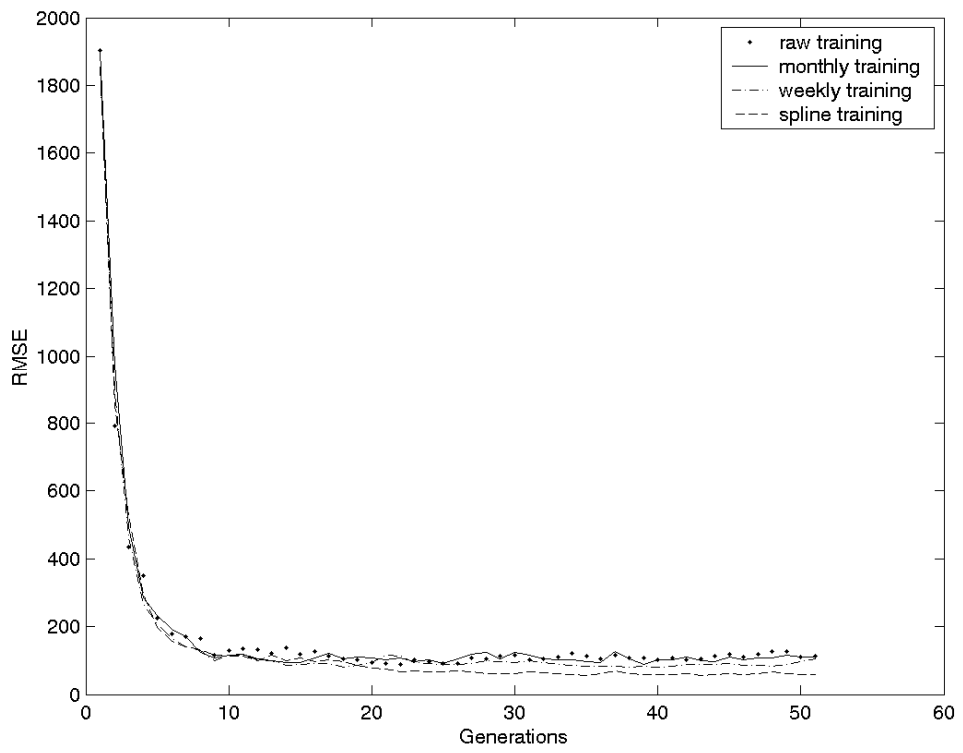


Table 1: Data Variables

Table 2: Proportion of Runs (N=1000), in which Variance does not Decrease after Linear Interpolation

Table 3: Mean and Standard Deviation of Attributes in Original and Interpolated Data Sets

Table 3: Context Free Grammar for Difference Equations

Table 4: Genetic Programming Parameters

Table 5: RMSE of Naïve Model on Training and Test Sets

Table 6: Best of the Best Individuals for each Training Regime

Figure 1: Chlorophyll A Time Series Data

Figure 2: Water Temperature Time Series Data

Figure 3: Best Error (RMSE) vs Generation on Original Training Data

Figure 4: Best Error (RMSE) vs Generation on Monthly Training Data

Figure 5: Best Error (RMSE) vs Generation on Weekly Training Data

Figure 6: Best Error (RMSE) vs Generation on Spline Training Data

Figure 7: Best Error (RMSE) vs Generation on Original Testing Data

Figure 8: Mean Error (RMSE) vs Generation on Original Testing Data