

Modeling and analyzing data from computer experiments.

Some general observations.

1. For simplicity, I assume that all factors (inputs) x_1, x_2, \dots, x_d are quantitative.

2. Because the code always produces the same output, $y(\mathbf{x})$, at a given value of \mathbf{x} , a model that produces estimates (predictions) of the output, which interpolate the data (pass exactly through the observed data) is advantageous.

3. Because the code is somewhat of a “black box” we don’t usually know the functional form of $y(\mathbf{x})$ a priori. Thus, it is advantageous to consider statistical models that are capable of approximating a wide variety of functional shapes.

4. Because the code runs slowly, we need models that can be fit with relatively few observations.

Some possible models

Regression models

Advantages:

- Easy to fit and interpret.
- By choosing a high enough degree polynomial, you can approximate any function.

Disadvantages:

- Does not necessarily interpolate the data.
- If there are many inputs (d is large), many observations are required to fit a polynomial of high degree.

Other choices

- splines
- neural nets
- nonparametric smoothers

Problem: Many of these other methods provide point predictors but not standard errors of prediction. In other words, they do not readily provide a confidence or prediction interval for predictions based on them.

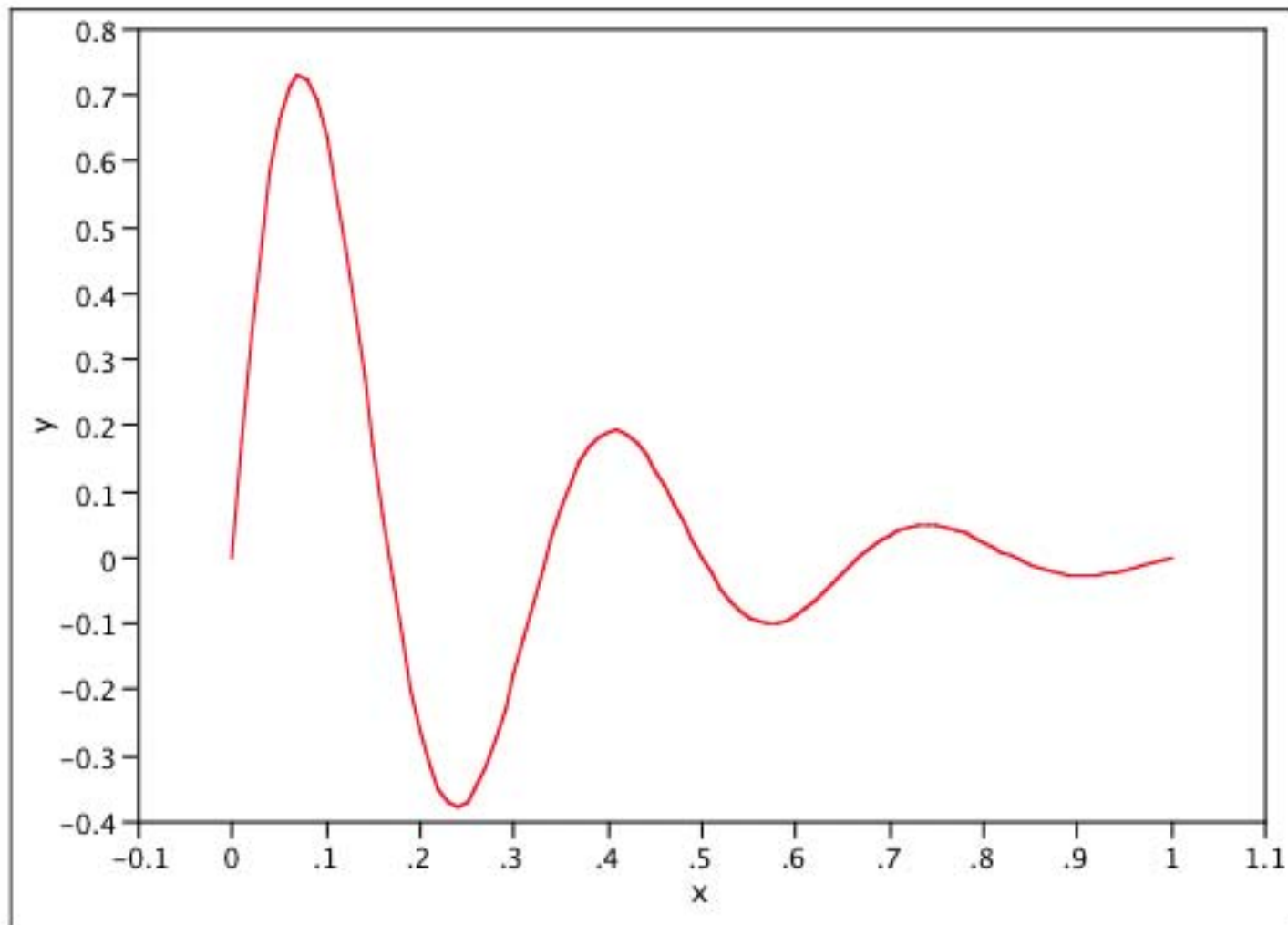
Some of these can be fit on JMP, but I will not discuss these further because they do not give standard errors of prediction.

An example

What happens if we use regression to fit a somewhat complicated function?

The damped sine function

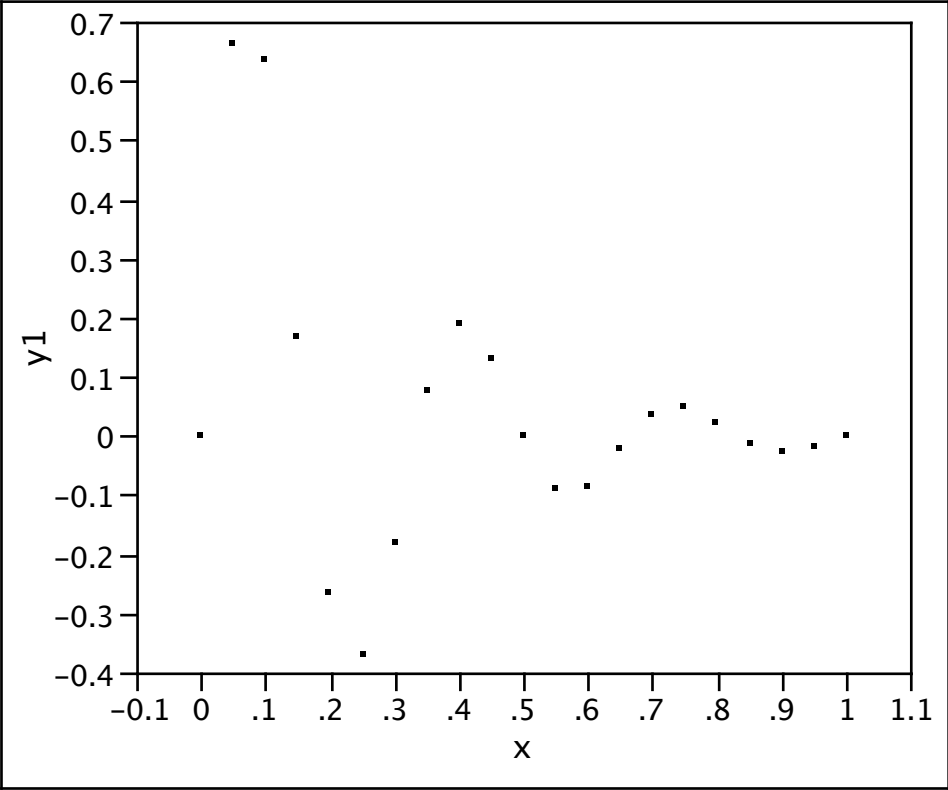
$$y(x) = e^{-4x} \sin(6\pi x)$$



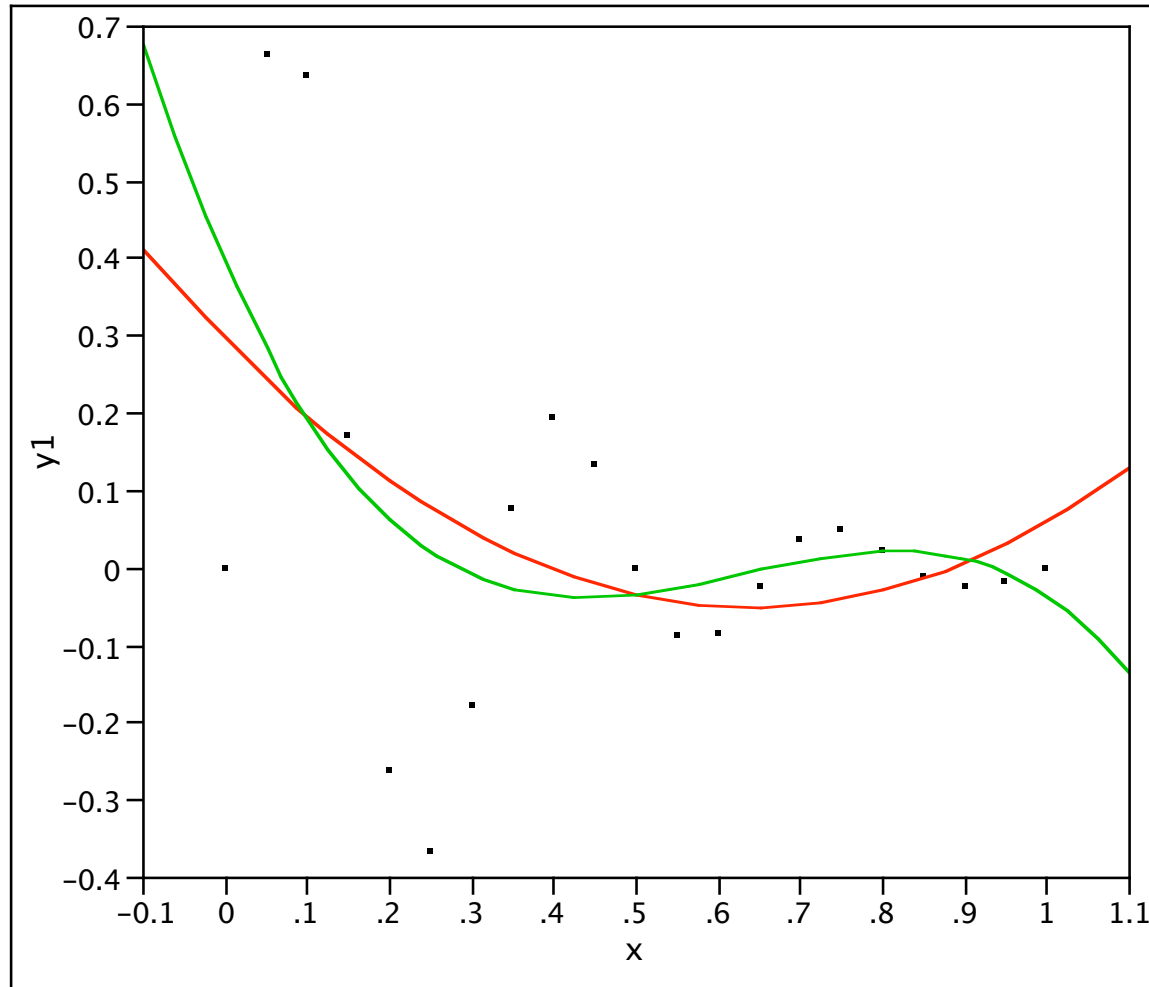
How well can we fit this with polynomial regression models using a 21 point LHS on the grid of points 0, 0.05, 0.10, 0.15, ..., 0.95, 1?

Note: A classic design, such as a central composite design, would only take observations at 0, 0.5, and 1 (and perhaps only one or two other interior points). Obviously, such a design would not provide enough data to see the oscillations.

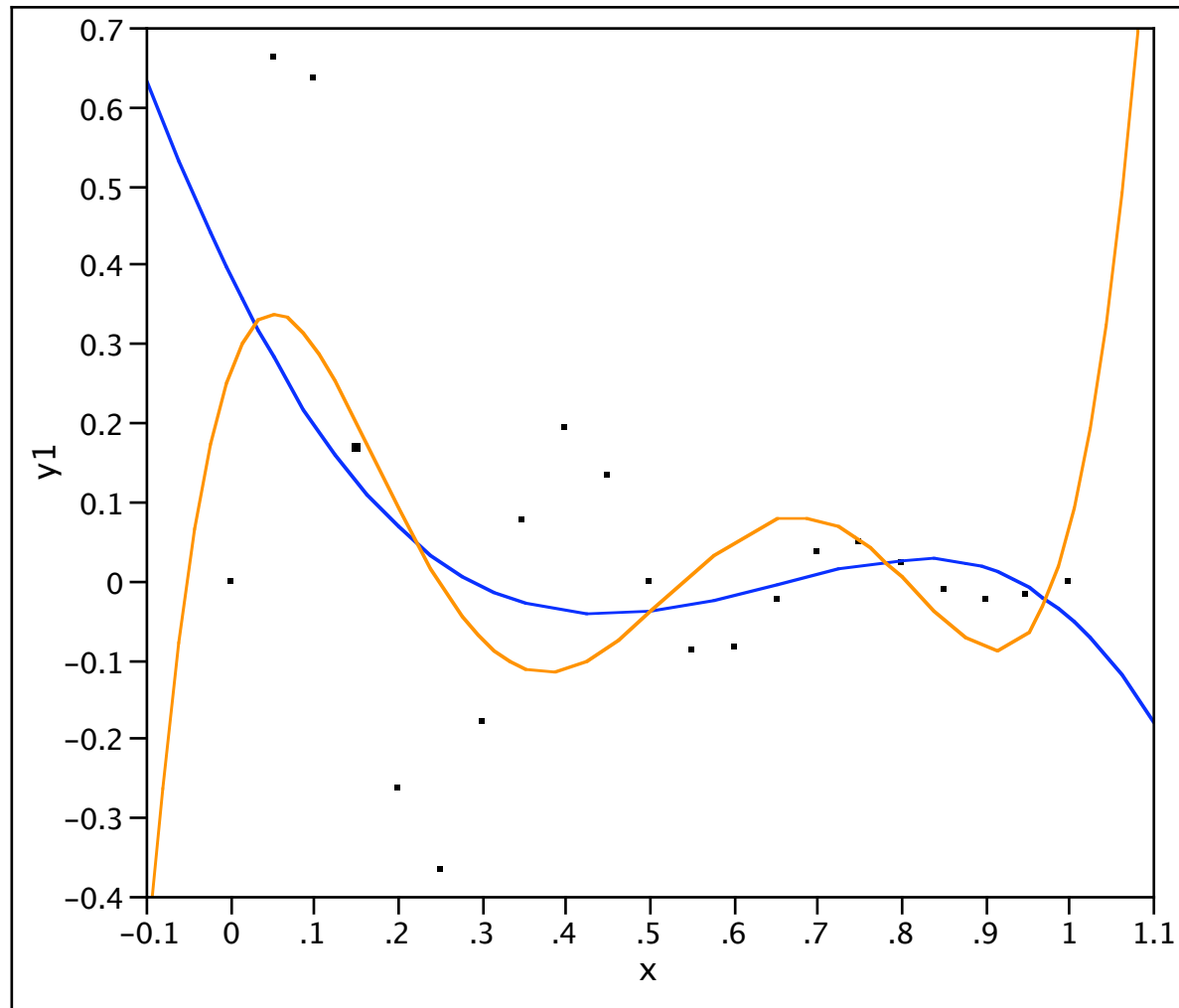
Below are a sequence of fits using JMP



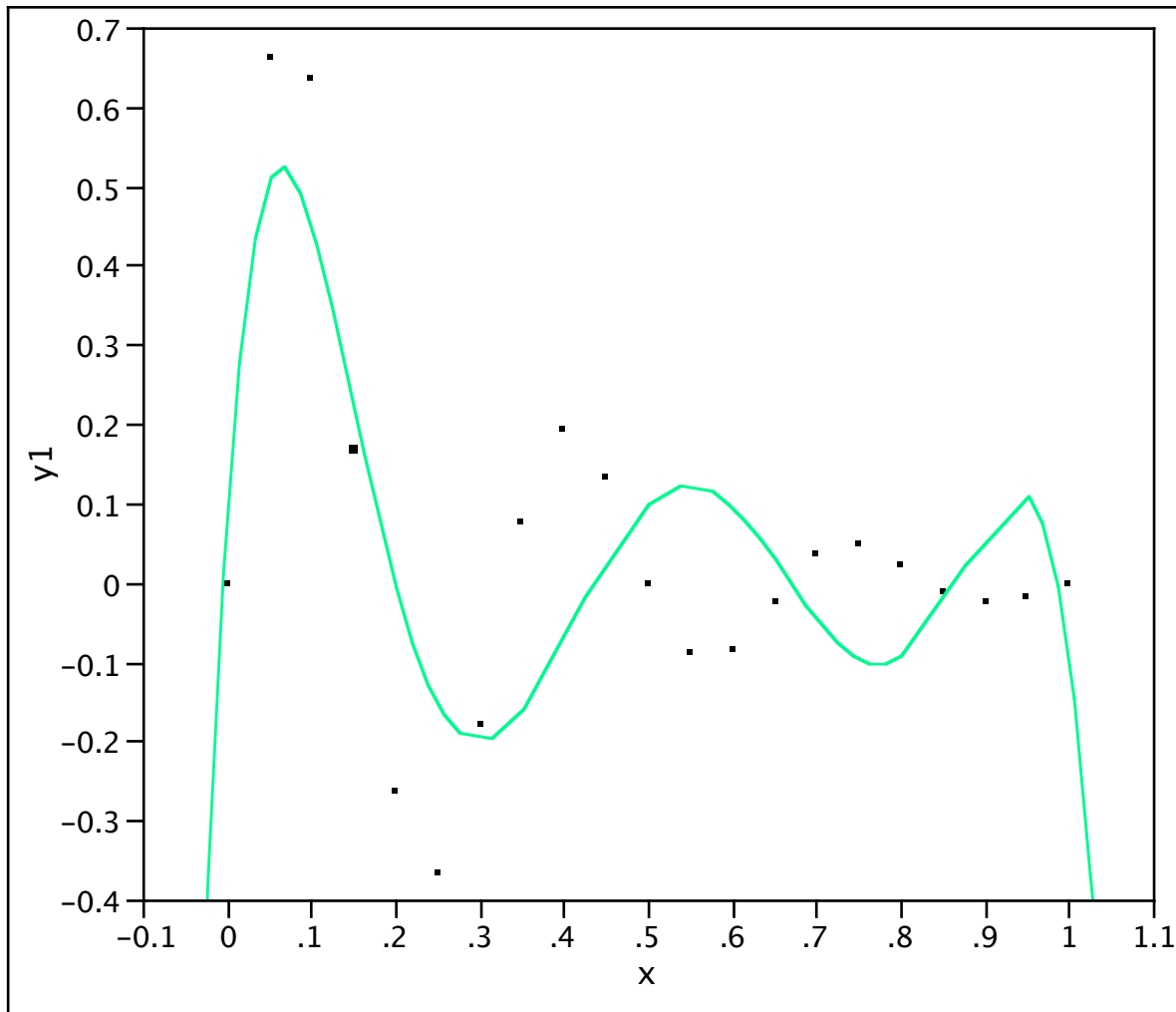
Second (red) and third (green) degree polynomial fit



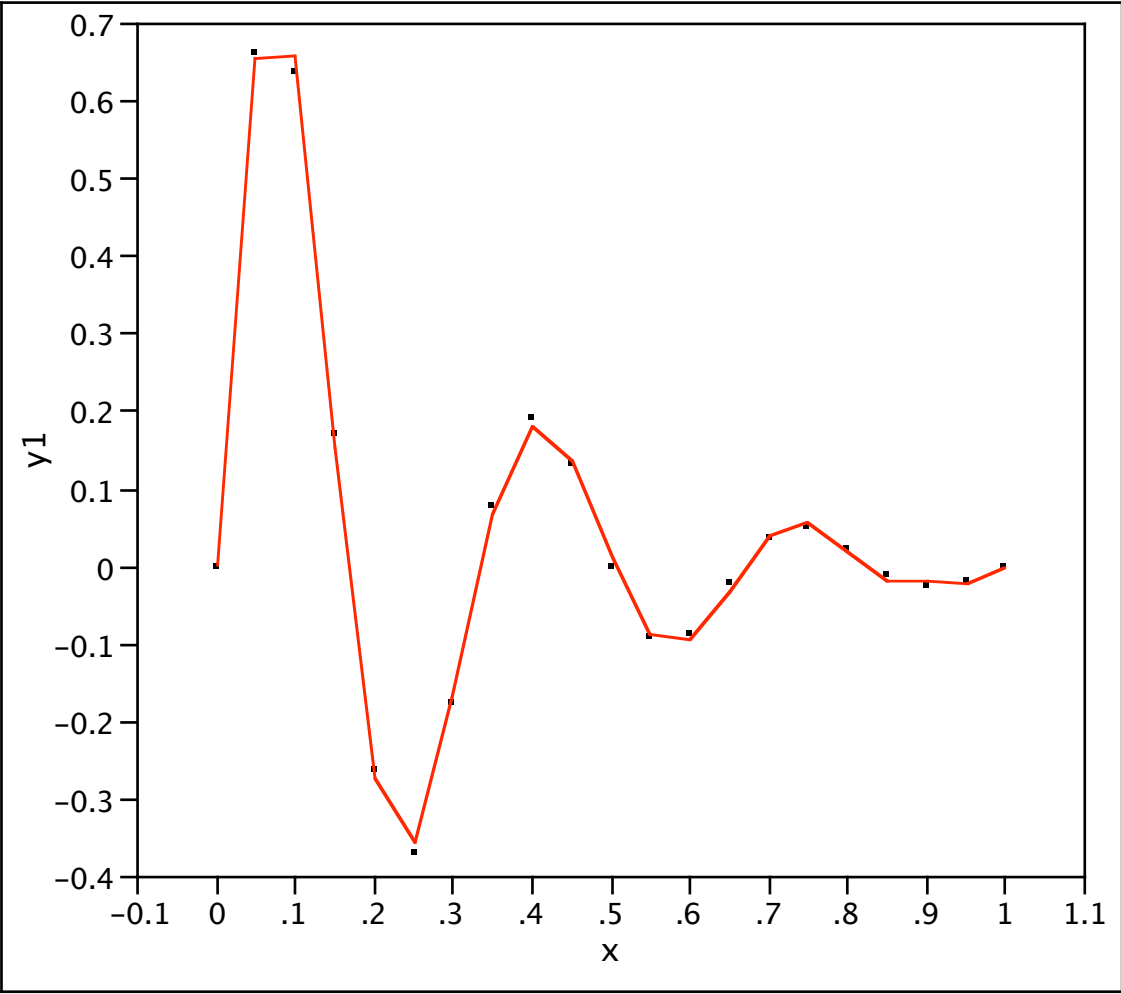
Fourth (blue) and fifth (orange) degree polynomial fit



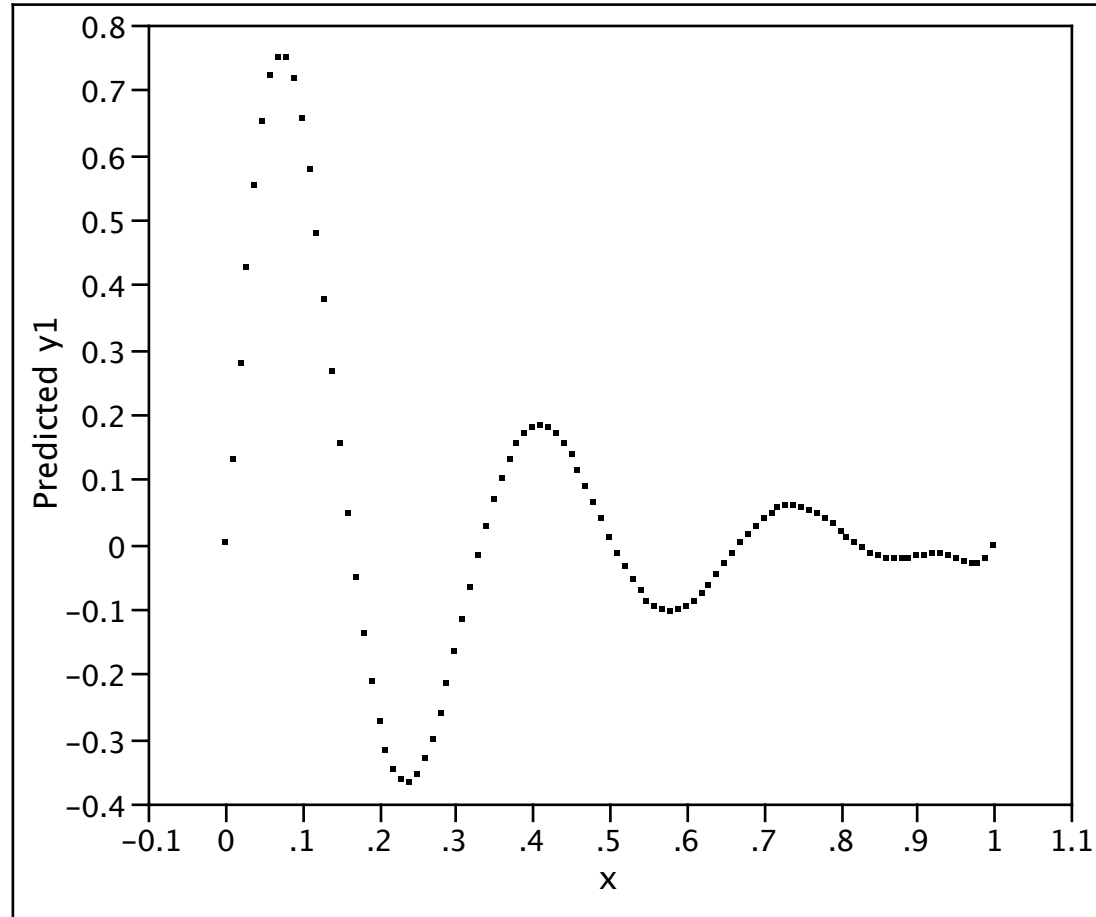
Sixth degree polynomial fit



A tenth degree polynomial fit.



The predicted values over the grid 0 to 1 in steps of 0.01.



The lesson is that regression models (of sufficiently high degree) can fit complicated looking functions. However, if you need a very high degree polynomial, this will not be satisfactory in high dimensions (many inputs) because a very large number of observations will be needed to fit the models.

Imagine a 10th degree polynomial in d inputs. The model will have a very large number of parameters and hence require a very large number of observations just to fit the model.

For reasons such as the above, another class of models that generalizes the above (actually includes regression, splines, and even certain neural nets as special cases) has become popular in practice.

These models are called Gaussian Stochastic Process (GASP) models. Let me now describe them.

I apologize in advance because this part of the talk is more technical than anything I have discussed so far.

Gaussian Process Models (GASP models)
(popular in spatial statistics and sometimes referred to as kriging models)

View $y(\mathbf{x})$ as a realization of the random function

$$Y(\mathbf{x}) = \beta_0 + \beta_1 f_1(\mathbf{x}) + \dots + \beta_p f_p(\mathbf{x}) + Z(\mathbf{x})$$

where $Z(\mathbf{x})$ is a mean zero, second-order stationary Gaussian process, and

$$\text{Cov}(Y(\mathbf{x}_1), Y(\mathbf{x}_2)) = \sigma_Z^2 R(\mathbf{x}_1 - \mathbf{x}_2)$$

Here the β_i are unknown constants (regression parameters) and the f_i are known regression functions.

Notice that if $Z(\mathbf{x})$ was replaced by independent random errors, this would be the standard general linear model. However, we have allowed observations to be correlated.

What does second-order stationary Gaussian process mean?

Second-order means that the mean and variance of $Y(\mathbf{x})$ are constant, i.e., do not depend on \mathbf{x} .

Stationary means that the covariance (correlation) between $Y(\mathbf{x}_1)$ and $Y(\mathbf{x}_2)$ is a function only of the difference $\mathbf{x}_1 - \mathbf{x}_2$.

Gaussian process means that for any $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$, the joint distribution of $Y(\mathbf{x}_1), Y(\mathbf{x}_2), \dots, Y(\mathbf{x}_n)$ is multivariate normal.

R here is the so-called correlation function.

1. As presented here, the correlation function tells you how correlated two observations $Y(\mathbf{x}_1)$ and $Y(\mathbf{x}_2)$ are as a function of the difference $\mathbf{x}_1 - \mathbf{x}_2$ between the two inputs.

In practice, people typically use correlation functions that only depend on some measure of distance between \mathbf{x}_1 and \mathbf{x}_2 .

2. Not any old function will do for R . R must satisfy several conditions. For example,

a. $R(0) = 1$.

b. For any n values $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$, the $n \times n$ matrix whose i, j -th entry is $R(\mathbf{x}_i - \mathbf{x}_j)$ must be a valid correlation matrix (e.g., nonnegative definite).

3. The spectral density theorem provides one method for constructing correlation functions. In particular, for any probability density function φ on \mathbb{R}^d

$$R(\mathbf{x}) = \int_{\mathbb{R}^d} \cos(\mathbf{x}^\top \mathbf{w}) \phi(\mathbf{w}) d\mathbf{w}$$

is a valid correlation function.

4. Some of the most popular correlation functions in practice are

a. The Gaussian correlation function

$$R(\mathbf{x}) = \prod_{i=1}^d \exp(-\theta_i x_i^2)$$

which is generated by the standard multivariate normal density.

b. The power exponential correlation function

$$R(\mathbf{x}) = \prod_{i=1}^d \exp(-\theta_i x_i^{p_i})$$

Here $0 \leq p_i \leq 2$. Note when all $p_i = 2$, this is the Gaussian correlation function.

c. The Matern correlation family

$$R(\mathbf{x}) = \prod_{i=1}^d \frac{1}{\Gamma(\nu_i) 2^{\nu_i-1}} \left(\frac{2\sqrt{\nu_i} |x_i|}{\theta_i} \right)^{\nu_i} K_{\nu_i} \left(\frac{2\sqrt{\nu_i} |x_i|}{\theta_i} \right)$$

which is generated by the density corresponding to that of d independent t random variables with ν_i degrees of freedom. Here K_ν is a modified Bessel function of order ν and Γ is the gamma function.

5. Some properties of these correlation functions are:
- a. The Gaussian correlation function is the limit of the Matern correlation function as the ν_i tend to ∞ . This is an immediate consequence of the fact that the t distribution tends to the normal as the degrees of freedom go to ∞ .

b. The Gaussian correlation function produces $Y(x)$ that are infinitely differentiable as a function of x .

c. The Matern correlation function produces $Y(x)$ that have $\nu-1$ derivatives as a function of x . Thus ν controls the “smoothness” of $Y(x)$.

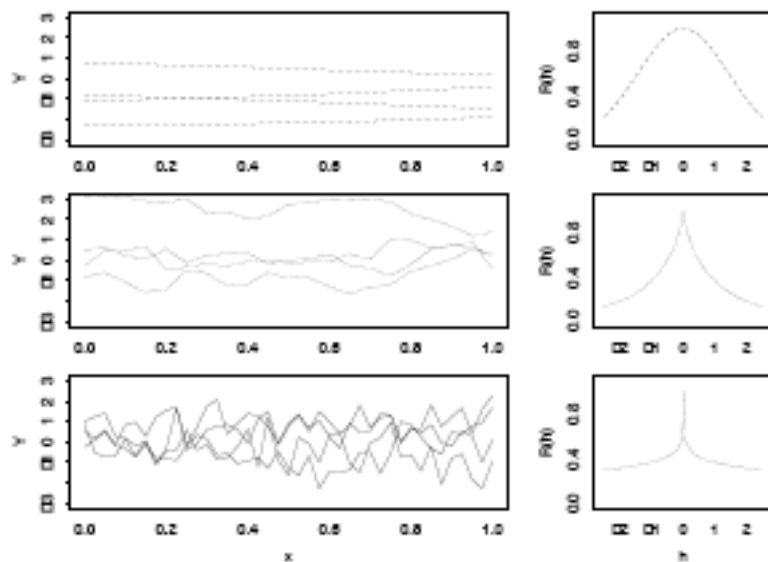
d. The power exponential correlation function produces $Y(x)$ that are infinitely differentiable as a function of x when $p = 2$ but are only continuous but not differentiable when $0 \leq p < 2$.

6. In practice, the most common correlation function is the power exponential correlation function. To get a sense of how the parameters θ_i and p_i affect behavior of $Y(x)$, consider the following figures.

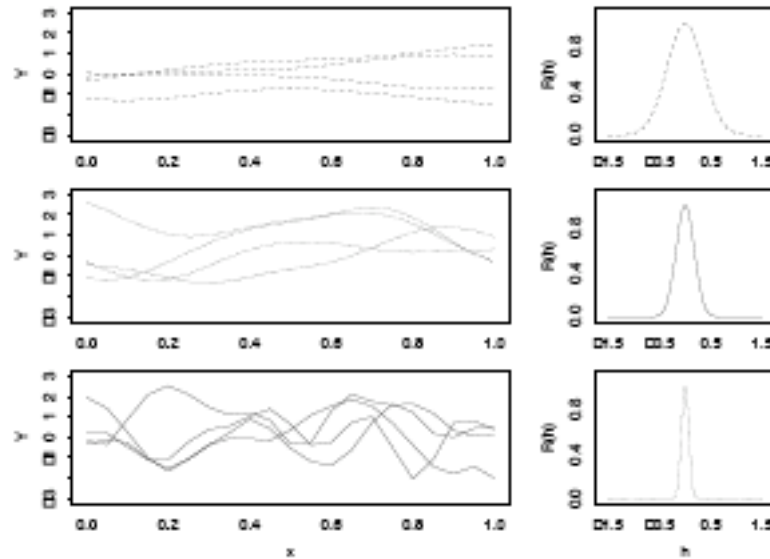
In these figures, we have taken the regression part of the model to be just the intercept, i.e.

$$Y(\mathbf{x}) = \beta_0 + Z(\mathbf{x})$$

The effect of varying p on $Y(x)$. Here $\theta = 1$ and p is 2, 0.75, and 0.20. As p decreases, $Y(x)$ gets rougher and rougher.



The effect of varying θ on $Y(x)$. Here $p = 2$ and θ is 0.5, 0.25, and 0.1. As θ decreases, $Y(x)$ varies more rapidly.



Prediction with GASP models

If we observe $y(\mathbf{x})$ at $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$, and wish to predict the value of $y(\mathbf{x})$ at the new input \mathbf{x}_0 , we will use the so-called empirical best linear unbiased predictor, or EBLUP.

$$\hat{y}(\mathbf{x}_0) = \mathbf{f}_0^\top \hat{\boldsymbol{\beta}} + \mathbf{r}_0^\top \hat{\mathbf{R}}^{-1} (\mathbf{Y}^n - \mathbf{F} \hat{\boldsymbol{\beta}})$$

$$\hat{y}(\mathbf{x}_0) = \mathbf{f}_0^\top \hat{\boldsymbol{\beta}} + \mathbf{r}_0^\top \hat{\mathbf{R}}^{-1} (\mathbf{Y}^n - \mathbf{F} \hat{\boldsymbol{\beta}})$$

where

$$\mathbf{f}_0 = (1, f_1(\mathbf{x}_0), \dots, f_p(\mathbf{x}_0))^\top$$

\mathbf{F} is the $n \times (p+1)$ (design) matrix whose i, j -th entry is $f_{j-1}(\mathbf{x}_i)$, for $j > 1$ and 1 if $j = 1$.

$$\mathbf{r}_0 = (R(\mathbf{x}_0 - \mathbf{x}_1), R(\mathbf{x}_0 - \mathbf{x}_2), \dots, R(\mathbf{x}_0 - \mathbf{x}_n))^\top$$

\mathbf{R} is the $n \times n$ matrix whose i, j -th entry is $R(\mathbf{x}_i - \mathbf{x}_j)$

$\hat{\mathbf{R}}$ is the maximum likelihood estimate of \mathbf{R} .

$$\hat{y}(\mathbf{x}_0) = \mathbf{f}_0^\top \hat{\boldsymbol{\beta}} + \mathbf{r}_0^\top \hat{\mathbf{R}}^{-1} (\mathbf{Y}^n - \mathbf{F} \hat{\boldsymbol{\beta}})$$

$$\mathbf{Y}_n = (y(\mathbf{x}_1), y(\mathbf{x}_2), \dots, y(\mathbf{x}_n))^\top$$

and

$$\hat{\boldsymbol{\beta}} = (\mathbf{F}^\top \hat{\mathbf{R}}^{-1} \mathbf{F})^{-1} \mathbf{F}^\top \hat{\mathbf{R}}^{-1} \mathbf{Y}^n$$

is the generalized least squares estimate of $\boldsymbol{\beta}$.

Some facts.

1. If \mathbf{x}_0 is one of the inputs $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$, say $\mathbf{x}_0 = \mathbf{x}_i$, then

$$\hat{y}(\mathbf{x}_i) = y(\mathbf{x}_i)$$

i.e., the EBLUP interpolates the data.

2. The EBLUP is a complicated nonlinear function of the data because it involves the inverse of the maximum likelihood estimate of R .

3. If we use the simple intercept model as the regression part of our model, i.e., we use the model

$$Y(\mathbf{x}) = \beta_0 + Z(\mathbf{x})$$

the EBLUP is still an interpolator and does a surprisingly good job of fitting observed data.

In practice, people often just fit this simple form of the GASP model.

4. The GASP model and the EBLUP should look somewhat familiar. The GASP model looks like a linear model with correlated observations and the EBLUP is related to generalized least squares. In fact, this leads to another way of looking at the GASP model.

Another take on the GASP model.

The GASP model,

$$Y(\mathbf{x}) = \beta_0 + \beta_1 f_1(\mathbf{x}) + \dots + \beta_p f_p(\mathbf{x}) + Z(\mathbf{x})$$

can be viewed as a mixed model in the framework of the general linear model, where all observations are taken on the same subject, hence correlated.

$Z(\mathbf{x})$ represents the within subject effect.

This point of view allows us to fit GASP models using
PROC MIXED in SAS!

Fitting GASP models and calculating EBLUPs

We use PROC MIXED in SAS. As of version 9.1, SAS allows you to fit the GASP model with the power exponential correlation function. Here is the code for fitting the damped sine wave data.

The data entry part of the code.

```
data dampedisine;
input case x y;
lines;
1 0.0 0.0
2 0.05 0.66236671
3 0.1 0.63751258
4 0.15 0.16959337
5 0.2 -0.2641078
6 0.25 -0.3678794
7 0.3 -0.1770387
8 0.35 0.07620135
9 0.4 0.1920146
10 0.45 0.13373031
11 0.5 0.00000108
12 0.55 -0.0896411
13 0.6 -0.0862782
14 0.65 -0.0229525
15 0.7 0.03574271
16 0.75 0.04978707
17 0.8 0.02395984
18 0.85 -0.0103125
19 0.9 -0.0259863
20 0.95 -0.0180985
21 1.0 0.0
22 0.00 .
23 0.01 .
24 0.02 .
25 0.03 .
.
.
.
115 0.93 .
116 0.94 .
117 0.95 .
118 0.96 .
119 0.97 .
120 0.98 .
121 0.99 .
122 1 .
;
```

Notice the lines ending with dots specify that the value of y is missing. These lines will force SAS to compute the EBLUP predictor of y for the values of x given on these lines – here all x values between 0 and 1 in steps of 0.01.

You can also read data in from other files (such as spreadsheets) with the import command in the SAS menu.

The basic PROC mixed commands

```
PROC mixed method=ml;  
model y = /outp = predictions;  
repeated / type=sp(expa) (x)  
          subject=intercept;  
parms  
      /lowerb= .,0,. upperb= .,2,. ;  
estimate 'intercept' intercept 1;  
run;
```

The first line. I have specified that the method of estimation used be maximum likelihood. You can also specify that the method be restricted maximum likelihood by replacing `method = ml` with `method = reml`. The default method is `reml` if no method is specified

```
PROC mixed method=ml;
```

Reml is usually advocated as the preferred method because it supposedly avoids some problems (with potentially ill-posed parameterizations) that can occur with maximum likelihood. In practice, we have found the performance to be similar. I will use maximum likelihood in the example because most of you are probably more familiar with maximum likelihood. I have tended to use maximum likelihood, but have occasionally tried reml if there have been convergence problems to see if reml avoids these problems. It usually hasn't.

The model statement. I have used the intercept only model by writing

```
model y = /
```

You can include a regression model here, for example, the statement

```
model y = x /
```

```
model y = /outp = predictions;
```

outp = predictions (after the /) specifies the name of the output file that predicted values will be stored in. You can then do additional analysis on the data in this output file (graphical displays, for example) using SAS.

The repeated option is crucial.

The first line after the /, i.e., the

type =

indicates the type of correlation function used.

```
repeated / type=sp (expa) (x)  
          subject=intercept;
```

The SAS manual lists the various correlation structures allowed. The

`type = sp(expa)(list of the input variables)`

asks for the power exponential.

I think “sp” indicates one is using a spatial correlation function and “expa” the anisotropic exponential correlation function (what I have called the power exponential). After `sp(expa)` you list the names of the input variables.

The second line after the /, namely

```
subject = intercept
```

tells SAS that it is to treat the data as coming all from a single subject. This forces SAS to treat all observations as correlated. This line is crucial for fitting GASP models in SAS.

Of course, one ends any command in SAS with a ;

The parameters command.

For the power exponential (type = sp(expa)), if there are d input variables, SAS assumes there are $2d + 1$ parameters. The first d are the d θ_i (scale parameters), the next d are the d p_i (powers), and the last parameter is the variance σ^2 .

In the damped sine wave, there is a single input, hence 3 parameters.

parms

/lowerb= ., 0, . upperb= ., 2, .;

I have given the simplest form of the parms command that I suggest you use.

lowerb =

specifies lower bounds for all the $2d+1$ parameters and constrains SAS's search for the maximum likelihood estimates of the parameters. lowerb = must be followed by a list of $2d+1$ values separated by commas. A "." indicates no lower bound is specified.

parms

/lowerb= ., 0, . upperb= ., 2, .;

Similar comments hold for

upperb =

This specifies upper bounds for all the $2d+1$ parameters and constrains SAS's search for the maximum likelihood estimates of the parameters.

parms

/lowerb= ., 0, . upperb= ., 2, .;

You can include just `lowerb =` , `upperb =` , or both. I noticed that SAS does not constrain the powers of the power exponential to be between 0 and 2, so I suggest you use the `lowerb =` and `upperb =` command to constrain the powers to be between 0 and 2. The power exponential is not a legitimate correlation function for powers below 0 or above 2.

```
parms
```

```
  /lowerb= ., 0, .   upperb= ., 2, . ;
```

There are some additional options for the parms command. These include the following.

1. parms followed by $2d+1$ values, each in parentheses, specifies starting values for each of the parameters. SAS begins its search for maximum likelihood estimates with these starting values. Good guesses at the starting values can improve the efficiency of the search.

In the damped sine wave example, the command

```
parms (1) (2) (0.5)  
      /lowerb= .,0,. upperb= .,2,. ;
```

would tell SAS to begin its search with $\theta = 1$, $p = 2$, and $\sigma^2 = 0.5$, while constraining the power to be between 0 and 2.

2. parms followed by $2d+1$ values, each in parentheses, followed by

`/hold=`

followed by a list of integers between 1 and $2d+1$ does the following. As before, the list in parentheses are the starting values for the parameters as SAS begins its search for the maximum likelihood estimates (mle's). `/hold` tells SAS which (if any) of these parameters to keep fixed. In other words, SAS is to assume these are the estimates for the parameters.

In our damped sine wave example,

```
parms (1) (2) (0.5) /hold = 2;
```

would tell SAS to hold the second parameter (the value of the power p) at value 2. In the power exponential, this is a useful option. Holding all the powers = 2 forces SAS to fit the Gaussian correlation function. If you believe the function you are fitting is smooth, use this device (remember the power exponential with powers other than 2 is continuous but not differentiable)

I could have used the command

```
parms (1) (2) (0.5) /hold = 2  
                lowerb= .,0,. upperb= .,2,. ;
```

but this would be sort of redundant. If I am forcing the power to be fixed at 2, asking SAS to constrain the search for the value of the power to the range 0 to 2 is unnecessary.

3. Other options are possible. Consult the SAS manual for specifics. One example is the following. You can specify a grid of starting values for the search for the maximum likelihood estimate of any parameter by the command (in the context of our damped sine wave example)

parms (0 to 1 by 0.1) (1 to 2 by 0.2) (2)

This tells SAS to begin the search for the mle for θ over the grid 0, 0.1, 0.2, ..., 1, to begin the search for the mle for p over the grid 1, 1.2, 1.4, ..., 2, and to begin the search for σ^2 at 0.5.

If you suspect that the parameters lie in a particular range of values, this may improve the quality of SAS's search for the mle.

You can supplement the command to use a grid of values for the starting search with the noiter option. This requests that no Newton-Raphson iterations be performed and that PROC mixed use the best value from the grid search to perform inferences. By default, iterations begin at the best value from the PARMs grid search. The code would be

```
parms (0 to 1 by 0.1) (1 to 2 by 0.2) (2)/ noiter
```

The last line in our code asks SAS to print out the estimate (generalized least squares estimate) of the intercept in our model. SAS does not print this out unless requested.

```
estimate 'intercept' intercept 1;
```

Our PROC mixed code again is

```
PROC mixed method=ml;  
model y = /outp = predictions;  
repeated / type=sp(expa) (x)  
          subject=intercept;  
parms  
      /lowerb= .,0,. upperb= .,2,. ;  
estimate 'intercept' intercept 1;  
run;
```

Of course, the

```
run;
```

command tells SAS to run PROC mixed.

The last few lines of the code were

```
Proc Print data = predictions;  
run;
```

which tell SAS to run PROC mixed and then to print out the predicted values in the file named “predictions” where we stored them.

Here is the output from this code.

The Mixed Procedure

Model Information

Data Set	WORK.SPACE
Dependent Variable	y
Covariance Structure	Spatial Anisotropic Exponential
Subject Effect	Intercept
Estimation Method	ML
Residual Variance Method	Profile
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Between-Within

Dimensions

Covariance Parameters	3
Columns in X	1
Columns in Z	0
Subjects	1
Max Obs Per Subject	21

Number of Observations

Number of Observations Read	122
Number of Observations Used	21
Number of Observations Not Used	101

Iteration History

Iteration	Evaluations	-2 Log Like	Criterion
0	1	-1.36501512	
1	3	-4.97132486	0.12207411
2	3	-112.02510169	.
3	0	-112.02510169	1.626528E24
4	0	-112.02510169	1.626528E24
5	0	-112.02510169	1.626528E24

WARNING: Did not converge.

One of the problems with fitting GASP models by maximum likelihood is that the likelihood can be flat over a large region and it can be difficult to find the maximum numerically. SAS limits the number of iterations in its search to 50 and the number of maximum likelihood evaluations to 150, and if the search hasn't converged you can get an error message.

You can increase the maximum number of iterations and the maximum number of likelihood evaluations with the command

PROC mixed maxiter = *number* maxfunc = *number*

I needed to refine my search for the maximum likelihood estimates. To this end, I fixed the power to be 2 (Gaussian correlation function) because I know the output should be smooth.

From the initial run, it appeared that the likelihood was becoming essentially flat around $\theta = 13$ or 14. To avoid convergence problems because of a flat likelihood, I restricted the range of θ to be 15 to 150 using the `lowerb=` and `upperb=` commands.

Here is the code I used.

```
PROC mixed method=ml;  
model y = /outp = predictions;  
repeated / type=sp(expa) (x)  
          subject=intercept;  
parms (100) (2) (1) / hold = 2  
      lowerb= 15,2,. upperb=150,2,.;  
estimate 'intercept' intercept 1;  
run;  
Proc Print data = predictions;  
run;
```

The Mixed Procedure

Model Information

Data Set	WORK.SPACE
Dependent Variable	Y
Covariance Structure	Spatial Anisotropic Exponential
Subject Effect	Intercept
Estimation Method	ML
Residual Variance Method	Profile
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Between-Within

Dimensions

Covariance Parameters	3
Columns in X	1
Columns in Z	0
Subjects	1
Max Obs Per Subject	21

Number of Observations

Number of Observations Read	122
Number of Observations Used	21
Number of Observations Not Used	101

Parameter Search

CovP1	CovP2	CovP3	Variance	Log Like	-2 Log Like
100.00	2.0000	1.0000	0.07624	15.9719	-31.9437

Iteration History

Iteration	Evaluations	-2 Log Like	Criterion
1	2	-32.43695268	0.00074016
2	3	-110.26247039	0.00000000

Convergence criteria met.

The Mixed Procedure

Covariance Parameter Estimates

Cov Parm	Subject	Estimate
SP(EXPA) x	Intercept	15.0000
Power x	Intercept	2.0000
Residual		26.6700

Fit Statistics

-2 Log Likelihood	-110.3
AIC (smaller is better)	-106.3
AICC (smaller is better)	-105.6
BIC (smaller is better)	-104.2

PARMS Model Likelihood Ratio Test

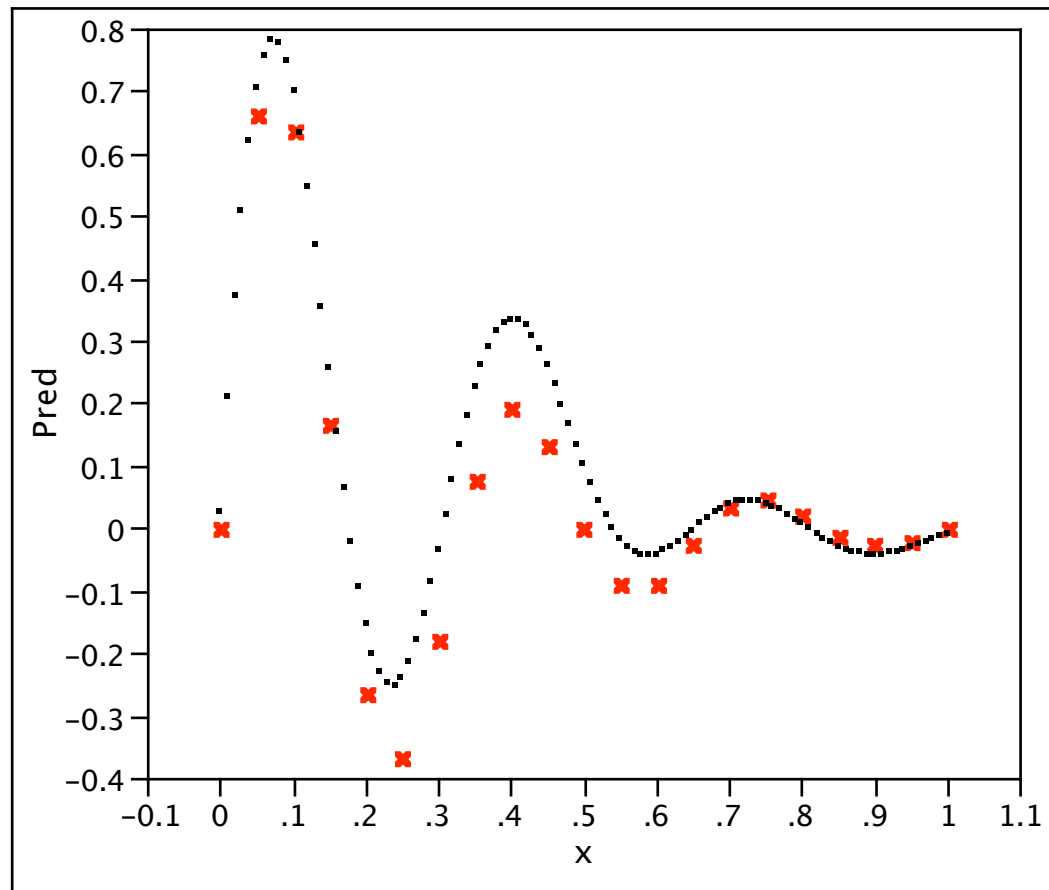
DF	Chi-Square	Pr > ChiSq
0	78.32	1.0000

Estimates

Label	Estimate	Standard Error	DF	t Value	Pr > t
intercept	1.2867	2.5141	0	0.51	.

Obs	case	x	y	Pred	StdErr	DF	Alpha	Lower	Upper	Resid
					Pred					
1	1	0.00	0.00000	1.28666	2.51407	20	0.05	-3.95761	6.53092	-1.28666
2	2	0.05	0.66237	1.28666	2.51407	20	0.05	-3.95761	6.53092	-0.62429
3	3	0.10	0.63751	1.28666	2.51407	20	0.05	-3.95761	6.53092	-0.64914
4	4	0.15	0.16959	1.28666	2.51407	20	0.05	-3.95761	6.53092	-1.11706
5	5	0.20	-0.26411	1.28666	2.51407	20	0.05	-3.95761	6.53092	-1.55076
6	6	0.25	-0.36788	1.28666	2.51407	20	0.05	-3.95761	6.53092	-1.65454
7	7	0.30	-0.17704	1.28666	2.51407	20	0.05	-3.95761	6.53092	-1.46369
8	8	0.35	0.07620	1.28666	2.51407	20	0.05	-3.95761	6.53092	-1.21045
9	9	0.40	0.19201	1.28666	2.51407	20	0.05	-3.95761	6.53092	-1.09464
10	10	0.45	0.13373	1.28666	2.51407	20	0.05	-3.95761	6.53092	-1.15293
11	11	0.50	0.00000	1.28666	2.51407	20	0.05	-3.95761	6.53092	-1.28666
.										
.										
.										
.										
.										
111	111	0.89	.	-0.040563	0.19708	20	0.05	-0.45167	0.37054	.
112	112	0.90	.	-0.041066	0.20544	20	0.05	-0.46962	0.38748	.
113	113	0.91	.	-0.040505	0.20933	20	0.05	-0.47716	0.39615	.
114	114	0.92	.	-0.038966	0.20907	20	0.05	-0.47507	0.39714	.
115	115	0.93	.	-0.036563	0.20530	20	0.05	-0.46480	0.39167	.
116	116	0.94	.	-0.033432	0.19886	20	0.05	-0.44824	0.38137	.
117	117	0.95	.	-0.029720	0.19081	20	0.05	-0.42774	0.36830	.
118	118	0.96	.	-0.025584	0.18186	20	0.05	-0.40493	0.35377	.
119	119	0.97	.	-0.021187	0.17174	20	0.05	-0.37943	0.33706	.
120	120	0.98	.	-0.016687	0.16094	20	0.05	-0.35240	0.31903	.
121	121	0.99	.	-0.012242	0.14980	20	0.05	-0.32472	0.30024	.
122	122	1.00	.	-0.007999	0.13850	20	0.05	-0.29692	0.28092	.

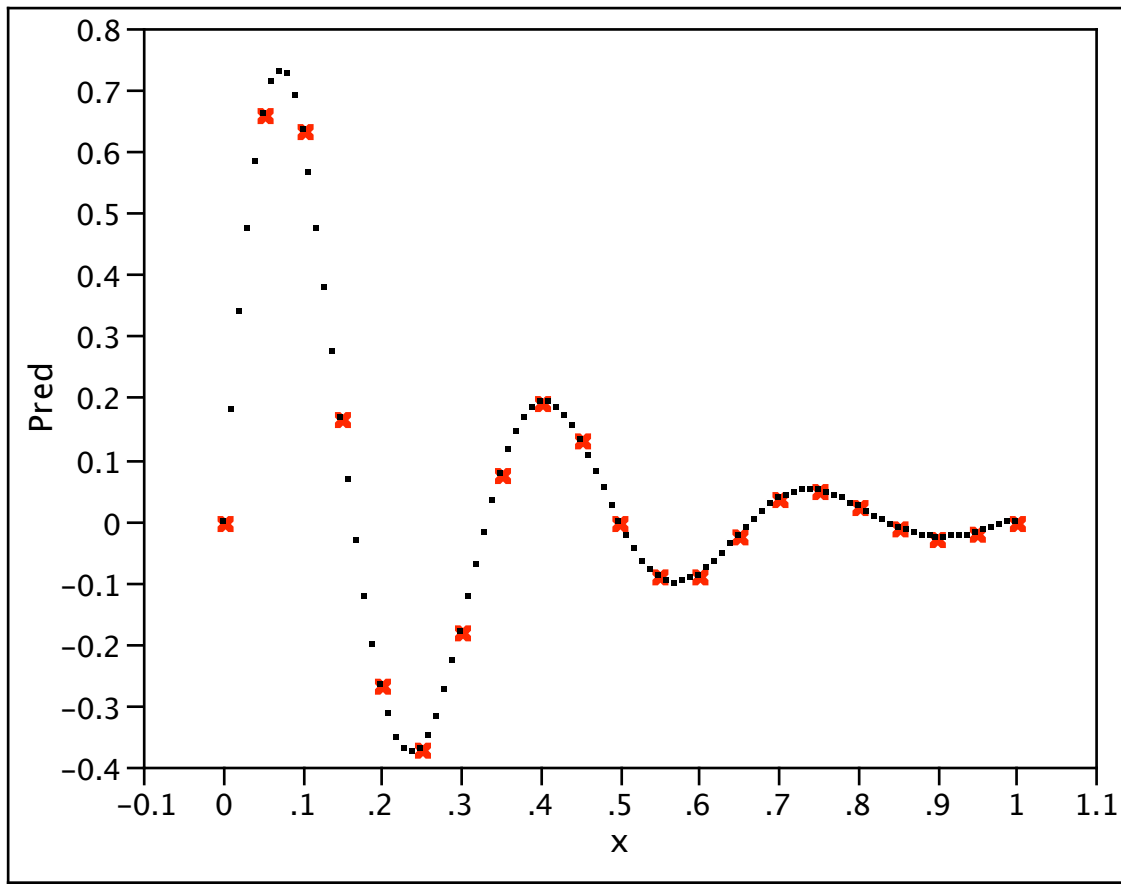
Here are the predicted values over the grid 0 to 1 in steps of 0.01. The red x's are the values produced by the code.



Note: The EBLUP is an interpolator, so the standard error of prediction at input values at which we ran the code should be 0. They are not here. This happens when the estimated R is close to singular and suggests that the fit is not trustworthy.

When I constrained θ to be no less than 50, this problem did not occur and the predictions were excellent.

Here are the predicted values over the grid 0 to 1 in steps of 0.01. The red x's are the values produced by the code.



Note how good the fit is *even though we didn't technically find the maximum likelihood estimate!*

It is this ability to fit well that has made these models popular.

It is good practice to check the quality of your fit via, say, cross validation. If the fit doesn't change much for various "near" maximum likelihood estimates of the parameters, you can be confident that your model is not sensitive to the particular choice you make. Otherwise, select the parameter values with best fit.

Comparing standard regression and GASP models

1. Standard regression models

Advantages: Easy to fit, easy to understand, easy to generate a variety of analyses (hypothesis tests, estimation, prediction, diagnostics)

Disadvantages: Will not give good fit if the output is not well approximated by a simple response surface (in other words, the output is a somewhat complicated surface).

2. GASP models

Advantages: Able to accommodate a wider variety of functional forms for the output. It interpolates the data. Includes standard regression as a special case.

Disadvantages: Fitting can be “touchy.” The likelihood may not converge and one has to adjust bounds over which the search for the maximum of the likelihood takes place. Not available in most commercial software.

Advice on modeling in computer experiments

1. If you have access to Unix, download our PErK software and fit the Matern model. Check the value of the smoothness parameters ν_i . If the values are tending to be large, this is evidence that the fitted surface is reasonably smooth. If not, use the Matern model for analysis.

2. If the Matern model suggests that the output is a smooth function of the inputs, or if you know on scientific grounds that the output should be smooth, fit both the power exponential model in SAS (constraining the powers p_i to be 2, i.e., using the Gaussian correlation function) and a classic response surface using regression (possibly after transformations). Compare the quality of the fits (perhaps via cross validation).

3. If the fits are comparable, the standard response surface is adequate and subsequent analysis can be done using the response surface model.

Otherwise, use the GASP model for subsequent analysis.