IMM
INFORMATICS AND MATHEMATICAL MODELLING

Technical University of Denmark
DK-2800 Kongens Lyngby – Denmark

J. No. DACE2
1.8.2002
HBN/ms

ASPECTS OF THE MATLAB TOOLBOX DACE

Søren N. Lophaven
Hans Bruun Nielsen
Jacob Søndergaard

TECHNICAL REPORT
IMM-REP-2002-13

IMM

Aspects of the Matlab Toolbox DACE

Søren N. Lophaven
Hans Bruun Nielsen
Jacob Søndergaard

Contents

1. Introduction

This report discusses some numerical aspects of the DACE Toolbox for Matlab, [10], which is an implementation of a *Kriging model*, based on

- A set of *design points* $(s_1, y_1), \ldots, (s_m, y_m)$, with y_i denoting the response at site $s_i \in \mathbb{R}^n$.
- A *regression model* $\mathcal{F}$. This is a linear combination of basis functions $f_1, \ldots, f_p$, chosen by the user, and $\mathcal{F}(\beta, x) = f(x)^\top \beta$, where $f(x) = [f_1(x) \ \ldots \ f_p(x)]^\top$.
- A *correlation model* $\mathcal{R}$, so that $\mathcal{R}(\theta, x, s) \in [0, 1]$ is the correlation between the responses at x and s. The vector $\theta \in \mathbb{R}^q$ holds parameters of the model.

The toolbox has two major programs

- `dacefit`. This computes the elements of the Kriging model, especially the parameters θ have to be found by solving a nonlinear optimization problem, see Sections 2, 5, 6.
- `predictor`. Predict the response at an untried site and estimate its error, Section 3.

Sections 2 and 3 give a short review of the theory from [10]. Section 4 introduces tools for analyzing and regularizing the matrices involved. Section 5 discusses the type of correlation models that the toolbox is aimed at, and how to enhance computational efficiency by exploiting special properties. Also, in Section 5.4 a new class of correlation models is introduced. Finally, Section 6 presents our algorithm for finding the optimal θ and Section 7 presents some ideas for further development of the DACE toolbox.

2. Dacefit

The function `dacefit` allows multiple responses. For the sake of simplicity, however, we only discuss simple responses, as presented in Section 1.

Let $\overline{S} \in \mathbb{R}^{m\times n}$ and $\overline{Y} \in \mathbb{R}^{m\times 1}$ contain the design sites and associated responses, and define the *normalized* data S, Y with

$$\begin{aligned} S_{:,j} &= \left(\overline{S}_{:,j} - \mu(\overline{S}_{:,j})\right) / \sigma(\overline{S}_{:,j}), \quad j = 1,\ldots,n \\ Y &= \left(\overline{Y} - \mu(\overline{Y})\right) / \sigma(\overline{Y}) \quad , \end{aligned} \tag{2.1}$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ denote respectively the mean and the standard deviation. All computation is made with the normalized data, where the mean is zero and the variance is one in each coordinate direction.

The matrix $F \in \mathbb{R}^{m\times p}$ is defined by $F_{i,:} = f(s_i)^\top$, and for a given set θ of correlation parameters we define $R \in \mathbb{R}^{m\times m}$ by $R_{ij} = \mathcal{R}(\theta, s_i, s_j)$.

The regression problem

$$F\beta \;\simeq\; Y \tag{2.2}$$

has the *generalized least squares* solution

$$\beta^* \;=\; \left(F^\top R^{-1} F\right)^{-1} F^\top R^{-1} Y \ , \tag{2.3}$$

and the variance estimate

$$\sigma^2 \;=\; \frac{1}{m}(Y - F\beta^*)^\top R^{-1}(Y - F\beta^*) \ . \tag{2.4}$$

The matrix R and thereby β^* and σ^2 depend on θ. The optimal choice θ^* is defined as the *maximum likelihood estimator*, the maximizer of

$$-\tfrac{1}{2}(m \ln \sigma^2 + \ln |R|) \ ,$$

where $|R|$ is the determinant of R. This is equivalent with the definition in [12]: θ^* is a minimizer of

$$\psi(\theta) \;=\; |R(\theta)|^{\frac{1}{m}} \cdot \sigma(\theta)^2 \ . \tag{2.5}$$

The algorithm for finding an optimizer of (2.5) is discussed in Section 6. It is an iterative process, and for large values of m the determination β^* for each new value of θ dominates the computational effort. In [10] we showed that instead of brute force evaluation of (2.3) – involving literal inversion of R – we can proceed as follows: Let

$$R \;=\; CC^T \tag{2.6}$$

denote the *Cholesky factorization* of the correlation matrix R, which is *symmetric* and *positive definite* (SPD), and introduce the *"decorrelation transformation"*

$$\tilde{Y} - \tilde{F}\beta \;\equiv\; \left(C^{-1}Y\right) - \left(C^{-1}F\right)\beta \ . \tag{2.7}$$

Then we can reformulate (2.3) to

$$\beta^* \;=\; \left(\tilde{F}^\top \tilde{F}\right)^{-1} \tilde{F}^\top \tilde{Y} \ ,$$

which we recognize as the solution to the normal equations for the overdetermined system of equations

$$\tilde{F}\beta \;\simeq\; \tilde{Y} \ . \tag{2.8}$$

Experience shows – cf. Sections 4 and 5 – that R may be very *ill conditioned*. This will be transferred to $\tilde{F}$ (which may also inherit a poor condition of F). In order to reduce effects of rounding errors we recommend to find β^* via *orthogonal transformation* of (2.8): Compute the "economy size" (or "thin") QR factorization [6, Section 5.2.6]

$$\tilde{F} \;=\; QG^\top \ , \tag{2.9}$$

where $Q \in \mathbb{R}^{m\times p}$ has orthonormal columns and $G^\top \in \mathbb{R}^{p\times p}$ is upper triangular. Then the least squares solution to (2.8) is found by back substitution in the upper triangular system

$$G^\top \beta^* \;=\; Q^\top \tilde{Y} \ . \tag{2.10}$$

The associated variance estimate is

$$\sigma^2 \;=\; \frac{1}{m}\|\tilde{Y} - \tilde{F}\beta^*\|^2 \ . \tag{2.11}$$

3. Predictor

The *Kriging estimator* at site x is given by

$$\hat{y}(x) \;=\; f(x)^\top \beta^* + r(x)^\top \gamma^* \;, \tag{3.1}$$

where the vector $r(x)$ has components $r_i = \mathcal{R}(\theta, x, s_i)$, and

$$\gamma^* \;=\; R^{-1}(Y - F\beta^*) \;=\; C^{-\top}(\tilde{Y} - \tilde{F}\beta^*) \;. \tag{3.2}$$

The estimated *mean squared error* (MSE) is

$$\varphi(x) \;=\; \sigma^2 \left(1 + \|v\|^2 - \|\tilde{r}\|^2\right) \;, \tag{3.3}$$

where

$$\begin{aligned} \tilde{r} &= C^{-1} r(x) \;, \\ v &= G^{-1}(\tilde{F}\tilde{r} - f(x)) \;. \end{aligned}$$

Thus, for each new site x we just have to compute the vectors $f(x)$ and $r(x)$ and add two dot products to get the predictor (3.1). The MSE involves the solution of two triangular systems with matrices computed during the fitting of the model, (2.6) and (2.9).

The *gradients* (with respect to x) of the predictor and the MSE are also of interest. The first one is

$$\hat{y}'(x) \;=\; J_f(x)^\top \beta^* + J_r(x)^\top \gamma^* \;, \tag{3.4}$$

where J_f and J_r is the Jacobian of f and r, respectively,

$$(J_f)_{ij} \;=\; \frac{\partial f_i}{\partial x_j}, \qquad (J_r)_{ij} \;=\; \frac{\partial \mathcal{R}}{\partial x_j}\,(\theta, x, s_i) \;. \tag{3.5}$$

From (3.3) it follows that the gradient of the MSE can be expressed as

$$\begin{aligned} \varphi'(x) &= 2\sigma^2 \left(J_v^\top v - J_{\tilde{r}}\tilde{r}\right) \\ &= 2\sigma^2 \left((J_{\tilde{r}}^\top \tilde{F} - J_f^\top) G^{-\top} v - J_{\tilde{r}}^\top \tilde{r}\right) \\ &= 2\sigma^2 \left(J_r^\top C^{-\top}(\tilde{F}w - \tilde{r}) - J_f^\top w\right) \;, \end{aligned} \tag{3.6}$$

where $w = G^{-\top} v$.

4. Interlude: Sensitivity

In Sections 2 and 3 there is a number of expressions like $\tilde{Y} = C^{-1}Y$ and $w = G^{-\top}v$. They are shorthand for "Solve the linear systems of equations $C\hat{Y} = Y$ and $G^\top w = v$."

In this connection it is important to realize that small changes in the matrix and/or right hand side may lead to large changes in the solution. If this is the case, the matrix is said to be *ill conditioned.* Also, on a computer every arithmetic operation suffers a *rounding error*,

$$\mathsf{fl}(a \diamond b) \;=\; (a \diamond b)(1 + \varepsilon) \quad \text{with} \;\; |\varepsilon| \le \varepsilon_{\mathsf{M}} \;,$$

where ε_{M} is the socalled *machine accuracy* (or *unit round-off*). With a reliable equation solver the computed solution $\overline{x}$ to the linear system $Ax = b$ can be shown [6, Section 3.5.1] to satisfy

$$\frac{\|\overline{x} - x\|}{\|x\|} \;\approx\; \kappa(A)\varepsilon_{\mathsf{M}} \;, \tag{4.1}$$

where $\kappa(A)$ is the (spectral) *condition number* of A. In words: we can expect to "loose" $\log_{10}(\kappa)$ digits because of rounding errors.

The spectral condition number for $A \in \mathbb{R}^{m\times n}$ is defined via the *Singular Value Decomposition* (SVD), see e.g. [6, Section 2.7.2],

$$A \;=\; U\Sigma V^\top \;, \tag{4.2}$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal and $\Sigma \in \mathbb{R}^{m\times n}$ is "diagonal" with elements

$$\Sigma_{11} \ge \Sigma_{22} \ge \cdots \ge \Sigma_{pp} \ge 0, \quad p = \min\{m, n\} \;. \tag{4.3}$$

Equation (4.2) is equivalent with $AV = U\Sigma$, or

$$AV_{:,j} \;=\; \Sigma_{jj} U_{:,j}, \quad j{=}1, \ldots, p \;, \tag{4.4}$$

This can be used to show that the spectral *norm* of the matrix,

$$\|A\| \;\equiv\; \max_{x\ne 0}\{\|Ax\|/\|x\|\} \;=\; \Sigma_{11} \;, \tag{4.5}$$

and the condition number is

$$\kappa(A) \;=\; \Sigma_{11}/\Sigma_{pp} \; . \tag{4.6}$$

Next, consider the eigensolutions of a symmetric matrix $R \in \mathbb{R}^{m\times m}$,

$$R v_j \;=\; \lambda_j v_j, \quad j = 1, \ldots, m \; , \tag{4.7}$$

with orthonormal eigenvectors $v_j \in \mathbb{R}^m$ and real eigenvalues $\{\lambda_j\}$. Without loss of generality we can assume that they are ordered so that

$$|\lambda_1| \ge |\lambda_2| \ge \cdots \ge |\lambda_m| \; . \tag{4.8}$$

In the SVD of R we get $U = V$ with $V_{:,j} = v_j$, and (4.4) is equivalent with $Rv_j = |\lambda_j| v_j$, i.e. $\Sigma_{jj} = |\lambda_j|$.

If R is SPD, then all its eigenvalues are positive, and they are equal to the singular values. Thus, for a symmetric matrix we can express the condition numbers in terms of the eigenvalues,

$$\begin{aligned} \kappa(R) \;&=\; (\max\ |\lambda_j|) \,/\, (\min\ |\lambda_j|) \; , \\ R \text{ is SPD:} \quad \kappa(R) \;&=\; \lambda_1/\lambda_m \; . \end{aligned} \tag{4.9}$$

Now consider the matrix $H = A^\top A$, where $A \in \mathbb{R}^{m\times n}$ with $m \ge n$. From (4.2), the orthogonality of U, and the diagonality of Σ it follows that

$$H \;=\; V\Sigma^\top U^\top U \Sigma V^\top \;=\; V\,\mathrm{diag}(\Sigma_{11}^2, \ldots, \Sigma_{nn}^2)\,V^\top \; . \tag{4.10}$$

This shows that H is symmetric and positive *semidefinite* (some of the singular values may be zero). Further, it follows that

$$\kappa(A^\top A) \;=\; (\kappa(A))^2 \; . \tag{4.11}$$

Combined with (4.1) this shows that if we use the normal equations to find the least squares solution to $Ax \simeq y$ with an ill conditioned A, then we may get only few (if any) correct digits in the computed solution.

The Cholesky factorization (2.6) can be computed only if R is SPD – and this is faster than computing the eigenvalues to check their positivity. An analysis similar to the derivation of (4.11) shows that

$$\kappa(C) \;=\; \sqrt{\kappa(R)} \; . \tag{4.12}$$

Note, however, that rounding errors imply that instead of the correct Cholesky factor we find a perturbed matrix $\overline{C}$, which according to [6, Section 4.2.7] satisfies

$$\overline{C}\,\overline{C}^\top \;=\; R + \Delta \quad \text{with} \quad \|\Delta\| \approx \varepsilon_{\mathsf{M}} \|R\| \; , \tag{4.13}$$

and if $\kappa(R) \gtrsim \varepsilon_{\mathsf{M}}{}^{-1}$, then the matrix $R+\Delta$ may be indefinite, so that the Cholesky factorization does not exist.

4.1. Regularization

Let $R \in \mathbb{R}^{m\times m}$ be a symmetric, ill conditioned matrix and consider the "regularized" matrix (to use the notation of [7])

$$\widehat{R} \;=\; R + \mu I \quad \text{with} \quad \mu > 0 \; . \tag{4.14}$$

From (4.7) it is easy to see that

$$\widehat{R} v_j \;=\; (\lambda_j + \mu) v_j, \quad j{=}1, \ldots, m \; , \tag{4.15}$$

with $\lambda_j = \lambda_j(R)$. This shows that $\widehat{R}$ has the same eigenvectors as R, and each eigenvalue is increased by μ. Thus, for sufficiently large μ all $\lambda_j{+}\mu > 0$, i.e., $\widehat{R}$ is SPD. Further, if R itself is SPD, then it follows from (4.9) and (4.15) that

$$\kappa(\widehat{R}) \;=\; \frac{\lambda_1 + \mu}{\lambda_m + \mu} \; , \tag{4.16}$$

and it is easy to show that $\kappa(\widehat{R}) < \kappa(R)$ for $\mu > 0$. If R is SPD and we use $\mu = K\varepsilon_{\mathsf{M}}\|R\|$, then the larger eigenvalues suffer insignificant changes if K is small, but $\widehat{R}+\Delta$ in (4.13) is SPD for sufficiently large K. We return to this in Section 5.1.

Now, consider the two linear systems of equations,

$$R\,x \;=\; b, \qquad \widehat{R}\,\widehat{x} \;=\; b\ .$$

In the basis formed by the orthonormal eigenvectors we find

$$b \;=\; \sum_{j=1}^{m} \alpha_j v_j \quad \text{with} \quad \alpha_j = v_j^\top b\ ,$$

and it follows from (4.7) and (4.15) that the solutions are

$$x \;=\; \sum_{j=1}^{m} \frac{\alpha_j}{\lambda_j}\, v_j\ , \qquad \widehat{x} \;=\; \sum_{j=1}^{m} \frac{\alpha_j}{\lambda_j+\mu}\, v_j\ .$$

Thus, the components in b corresponding to small eigenvalues are most enhanced in the solution. If R is SPD, then all components are damped in $\widehat{x}$ relative to x, and the components corresponding to the smallest eigenvalues suffer the largest change.

All the elements in a correlation matrix R are nonnegative, and for such a matrix it often holds, see [7], that the number of sign changes in v_j grows with j, i.e., the contributions from eigenvectors corresponding to the smallest eigenvalues exhibit a fast oscillating behaviour. This is damped when we regularize.

Finally, in the objective function (2.5) we use the determinant of R. It satisfies the relation

$$|R| \;=\; \prod_{j=1}^{m} \lambda_j\ . \tag{4.17}$$

Consider two extreme cases, cf. Section 5.1:

$\underline{R=E}$, the matrix of all ones, has the eigenvalues $\lambda_1 = m$, $\lambda_2 = \cdots = \lambda_m = 0$, and $|R|^{1/m} = 0$, while $|E+\mu I|^{1/m} = \mu^{(m-1)/m}\,\sqrt[m]{m} \to \mu$ for $m \to \infty$.

$\underline{R=I}$, the unit matrix, has all $\lambda_j = 1$, $|R|^{1/m} = 1$, $|R+\mu I|^{1/m} = 1+\mu$.

It should be mentioned that the determinant is not computed by means of (4.17). Instead we use (2.6),

$$|R|^{1/m} \;=\; |CC^\top|^{1/m} \;=\; \Big(\prod C_{jj}\Big)^{2/m} \;=\; \prod \Big(C_{jj}^{2/m}\Big)\ . \tag{4.18}$$

The last formulation is used to avoid the serious risk of underflow.

The idea of replacing an ill conditioned matrix R by $\widehat{R}$ defined by (4.14) is not new. In Kriging circles it is known as *"increasing the nugget effect"*; in general statistics it is *"ridge regression"*; in inverse problems it is *"Tikhonov regularization"* and in optimization it is *"damped Newton"* with *Levenberg-Marquardt's method* as a special case.

5. Correlation Models

We only consider stationary models, i.e., $\mathcal{R}(\theta,x,s)$ depends only on θ and the difference $d = x-s$. Further, like [12] we focus on models that have product form

$$\mathcal{R}(\theta,x,s) \;=\; \prod_{j=1}^{n} \mathcal{R}_j(\theta,(x-s)_j)\ . \tag{5.1}$$

This structure is, however, not explicitly exploited in `dacefit`.

Basic examples of such models are

$$\begin{array}{lrcl} \text{EXP} & \mathcal{R}_j(\theta,d) & = & \exp(-\theta_j|d_j|) \\ \text{GAUSS} & \mathcal{R}_j(\theta,d) & = & \exp(-\theta_j d_j^2) \end{array} \tag{5.2}$$

for $\theta_j > 0$. They are illustrated in Figure 5.1 below. Note that in both cases the correlation decreases with $|d_j|$ and a larger value for θ_j leads to a faster decrease. The normalization (2.1) of the data implies that

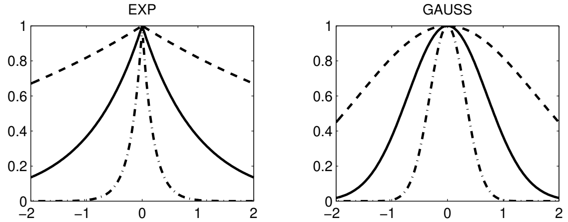

Figure 5.1. *Correlation functions for* $-2 \le d_j \le 2$.
Dashed, full and dash-dotted line: $\theta_j = 0.2,\ 1,\ 5$.

$|s_{ij}| \lesssim 1$ and therefore we are interested in cases where $|d_j| \lesssim 2$, as illustrated in the figure.

A major scope for the toolbox is to use the Kriging model as a surrogate for a continuously differentiable function, and from (3.4) it follows that J_r must be continuous across $d_j = 0$ in order to get a continuous gradient of the Kriging model. This is the case with GAUSS but not with EXP.

We start by taking a closer look at some properties of the matrices generated by (5.1). Numerical results are obtained from two classes of problems, defined by

$$\begin{array}{lrcl} \text{Design sites:} & \multicolumn{3}{l}{q \times q \text{ equidistant mesh over } [0,5]\times[0,10]} \\ \text{Problem 1:} & \Upsilon_1(x) & = & \sin \tfrac{1}{2}x_1 \cdot \sin \tfrac{1}{2}x_2 \\ \text{Problem 2:} & \Upsilon_2(x) & = & \sin 2x_1 \cdot \sin 2x_2 \end{array} \tag{5.3}$$

In this section we use the regression model $\mathcal{F}(x) = 1$ and only look at *isotropic* correlation models, i.e., all $\theta_j = \theta$. Since Υ_2 oscillates faster than Υ_1, we expect that $\theta^{(2)} > \theta^{(1)}$.

5.1. Conditioning

It is well known, see e.g. [3], that the correlation matrix may be very ill conditioned. In Section 4.1 we discussed two extreme cases: If all $\theta_j \to 0$, then $R \to E$, the matrix with all elements equal to one, while

$R \to I$, the unit matrix, when all $\theta_j \to \infty$. The matrix E has one eigenvalue equal to m, and all the other eigenvalues equal to zero. Therefore, for small θ we can expect R to be ill conditioned, while a large θ gives a well conditioned, significantly positive definite R. This is illustrated in Figure 5.2.

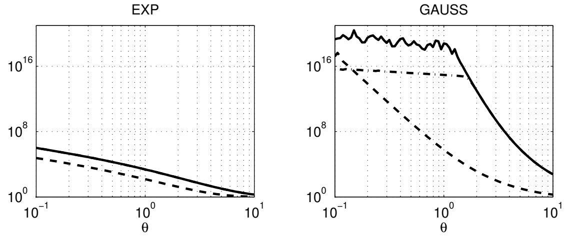

Figure 5.2. *Condition numbers for* R *given by* (5.2) *and* (5.3).
Dashed line: q=7. *Full line:* q=14.
Dash-dotted line: q=14, *regularized by* (5.4)

We see that EXP gives relatively well conditioned correlation matrices in this θ-range, while $R_{(\text{GAUSS})}$ is severely ill conditioned even for quite large θ-values, and the condition number grows with m, the number of design sites.

Similar to (4.13) it can be shown that the computed eigenvalues satisfy $\overline{\lambda}_j = \lambda_j + \delta$ with $|\delta| \approx \varepsilon_{\mathsf{M}}\|R\|$, so that if $\min|\lambda_j| \lesssim \varepsilon_{\mathsf{M}} \max|\lambda_j| \Leftrightarrow \kappa(R) \gtrsim 1/\varepsilon_{\mathsf{M}}$, then the matrix is not significantly SPD. The computation was done in MATLAB, and $\mathsf{fl}(\kappa(R)) \gtrsim 10^{15}$ indicates that computed results may be dominated by rounding errors. This calls for a regularization as discussed in the paragraph after (4.16). Experiments showed that

$$\widehat{R} \;=\; R + \mu I \qquad \text{with} \qquad \mu \;=\; (10{+}m)\varepsilon_{\mathsf{M}} \tag{5.4}$$

is a good compromise between ensuring that the matrix is significantly SPD without changing the solution too much. This is illustrated in Figure 5.2, where the results for $\widehat{R}_{(\text{EXP})}$ cannot be distinguished from the unregularized $R_{(\text{EXP})}$.

It is generally agreed, see e.g. [3], that the reason for the poor conditioning of the GAUSS matrix is the distribution of the off-diagonal elements in R. This is illustrated in Figure 5.3. For the smaller θ-values it is seen that GAUSS leads to a more even distribution among small and large elements than EXP.

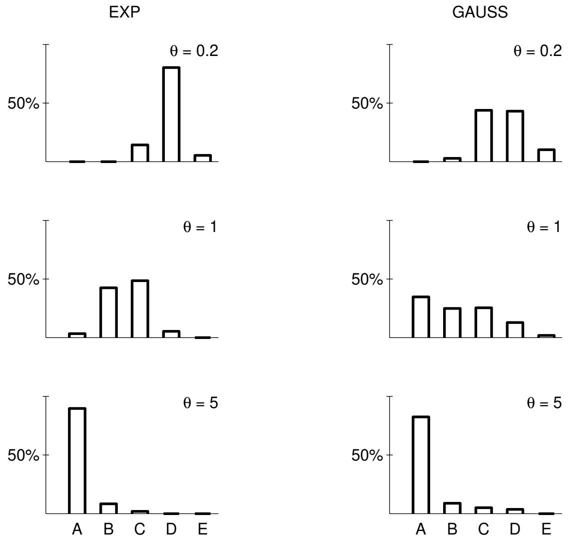

Figure 5.3. *Percentage of off-diagonal elements in the bins* A : $[0, 0.01]$, B : $]0.01, 0.1]$, C : $]0.1, 0.5]$, D : $]0.5, 0.9]$, E : $]0.9, 1]$ *R given by* (5.2) *and* (5.3) *with* $q = 14$

Next, Figure 5.4 shows how the two factors in (2.5) vary with θ. As already seen in Figure 5.2, the modification (5.4) from R to $\widehat{R}$ does not affect the results for EXP, but it has increasing effect on the GAUSS-results as θ decays. This, however, is the best we can do, and it does not spoil the essential information: The function $|R(\theta)|^{1/m}$ seems to grow monotonously from 0 to 1 as θ grows from 0 to ∞. The behaviour of σ^2 is more complex, but it has an asymptote at σ^2_∞, the variance for the simple least squares solution to (2.2).

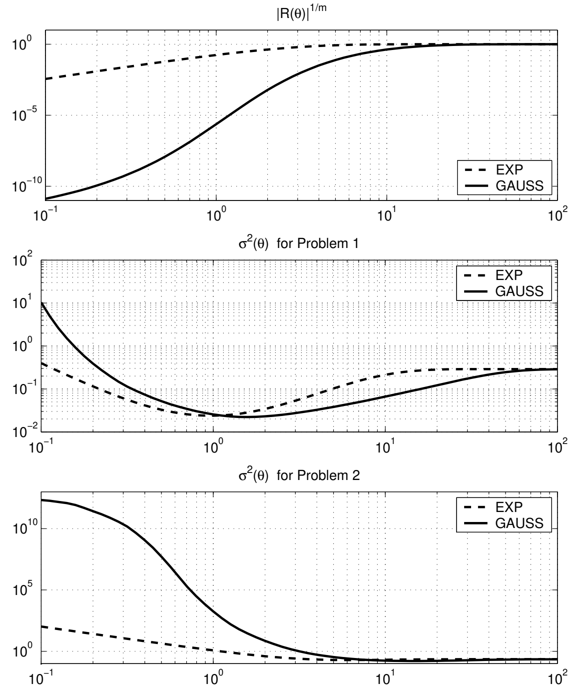

Figure 5.4. *Factors in* (2.5) *for* $0.1 \le \theta \le 100$. *$\widehat{R}$ given by* (5.2), (5.3) *and* (5.4) *with* q=14

The product ψ of the two functions is shown in Figure 5.5. In each of the four cases the function $\psi(\theta)$ has a unique minimizer,

	EXP	GAUSS
Υ_1	$\theta^* = 0.141$	$\theta^* = 0.178$
Υ_2	$\theta^* = 3.16$	$\theta^* = 1.26$

(5.5)

As expected, the optimizer for the faster oscillating Υ_2 is larger than the optimizer for Υ_1.

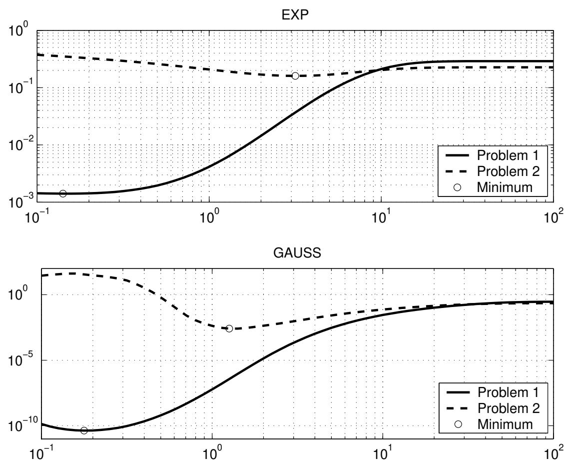

Figure 5.5. *$\psi = |\widehat{R}|^{1/m}\sigma^2$ for $0.1 \le \theta \le 100$. Experimental settings as in Figure 5.4*

In the remainder of this section we shall concentrate on properties of the GAUSS model. As we have seen, this is the hard one, and this is the type of model that has interest for surrogate modelling. Most of the results that we get will carry directly to correlation models of the EXP-type.

5.2. Use of Drop Tolerance

Figure 5.3 shows that for large values of θ many of the elements in R will be small, and it is tempting to ignore them. If a large number of elements are dropped, then R will be *sparse*, and this gives the possibility of a speed up by exploiting sparse matrix techniques. More specific, choose a threshold $\tau \in [0, 1[$ and define the reduced matrix $\overleftarrow{R} = \overleftarrow{R}^{(\tau)}$ by

$$(\overleftarrow{R})_{ij} = \begin{cases} 0 & \text{if } R_{ij} \le \tau \\ R_{ij} & \text{otherwise} \end{cases} \qquad (5.6)$$

Figure 5.6 shows the results for two τ-values. As a measure we use the *relative density* in $\overleftarrow{R}$ defined as

$$\text{rel. density} = \frac{\#\ \text{nonzeros in } \overleftarrow{R}}{m^2}\ , \qquad (5.7)$$

and for the sake of comparison we also give results for the models CUBIC and SPLINE treated in Sections 5.3 and 5.4. We get the expected

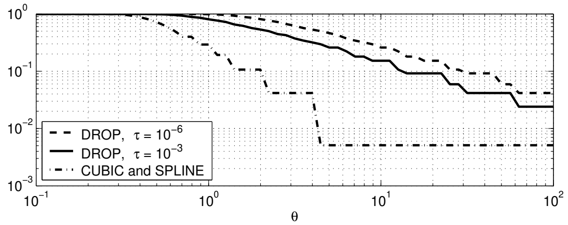

Figure 5.6. *Relative density in $\overleftarrow{R}$ defined by (5.6), (5.11) and (5.17). Design sites given by (5.3) with $q = 14$*

increasing sparsity as θ grows. For small values of θ no elements will be dropped, and we still need the stabilization as in (5.4).

Intuitively, the dropping of small elements gets us closer to the unit matrix, i.e., we should get a "more positive definite" matrix. This, however, is not the case, as we can see in Figure 5.7. There is a gap between $\theta \simeq 0.63$ and $\theta \simeq 4.0$, where $\widehat{R}$ is indefinite. Outside the gap the results agree with Figure 5.5.

This unexpected behaviour can be explained as follows: If we change R to $R+\Delta$, then the eigenvalues change,

$$\begin{aligned} \lambda_j(R+\Delta) &= \lambda_j(R) + \delta_j \\ &\simeq \lambda_j(R) + v_j^\top \Delta v_j\ , \end{aligned} \qquad (5.8)$$

where the estimate of δ_j follows from properties of the Rayleigh quotient [14, Section 55], and presumes that the matrix Δ has small elements. For the current problem, let R_{rs} be so small that we decide to drop it. Then we also drop R_{sr}, and assuming that these are the

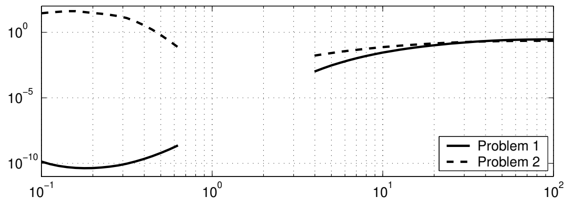

Figure 5.7. *ψ computed with* GAUSS *and $\tau = 10^{-6}$ in* (5.6)
Other settings as defined in Figures 5.4-5

only elements dropped, we have a perturbed matrix $\overleftarrow{R} = R + \Delta$ with $\Delta_{rs} = \Delta_{sr} = -R_{rs}$ as the only nonzero elements in Δ. Applying (5.8) we get

$$\lambda_j(\overleftarrow{R}) \;\simeq\; \lambda_j(R) - 2R_{rs}V_{jr}V_{js} \;\geq\; \lambda_j(R) - R_{rs}\ . \tag{5.9}$$

The lower bound follows from the normalization of v_j:

$$|V_{jr}V_{js}| \;\leq\; |V_{jr}|\sqrt{1 - V_{jr}^2} \;\leq\; 0.5 \quad \text{for } |V_{jr}| \leq 1\ .$$

Thus, if $\lambda_j(R) \leq R_{rs}$, then there is a risk that $\overleftarrow{R}$ is singular or indefinite. If we drop all elements smaller than the threshold τ, then Δ has contributions from all the dropped elements, and from (5.8) it can be shown that

$$\delta_j \;\geq\; -\nu \cdot \tau \;>\; -m \cdot \tau\ ,$$

where ν is the maximum number of elements dropped in a row. Combining this with (4.15) it is seen that it is possible to guarantee that $\widehat{R}$ is positive definite if we choose $\mu = m\tau$. In Section 5.5 we give results obtained with the regularization

$$\widehat{R} \;=\; \overleftarrow{R} + \mu I \quad \text{with} \quad \mu = (10{+}m)\varepsilon_{\mathrm{M}} + \sqrt{m}\cdot\tau\ . \tag{5.10}$$

5.3. Local Support

There is another way that a sparse R may arise, viz. through other choices of correlation model. The cubic correlation family [8] is one

such example

$$\begin{aligned} \mathcal{R}_j(\theta, d) \;&=\; 1 - \frac{3(1-\rho)}{2+\omega}\,\xi_j^2 + \frac{(1-\rho)(1-\omega)}{2+\omega}\,\xi_j^3 \\ &\qquad \text{with} \quad \xi_j = \min\{\theta_j|d_j|, 1\}\ . \end{aligned} \tag{5.11}$$

In Figure 5.8 we show

$$\begin{aligned} \mathcal{R}_j^{(1)}(\theta, d) \;&=\; 1 - 3\xi_j^2 + 2\xi_j^3\ , \\ \mathcal{R}_j^{(2)}(\theta, d) \;&=\; 1 - 1.5\xi_j^2 + 0.5\xi_j^3\ , \end{aligned} \tag{5.12}$$

corresponding to $(\rho, \omega) = (0, -1)$ and $(\rho, \omega) = (0, 0)$, respectively. As

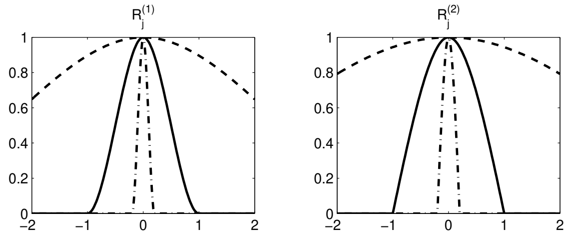

Figure 5.8. *Cubic correlation models,* (5.12).
Dashed, full and dash-dotted line: $\theta_j = 0.2,\ 1,\ 5$.

in Figure 5.1 a larger θ_j reduces the region of significant correlation, and as GAUSS both models have a well defined horizontal tangent at $d_j = 0$. From (5.11) we see that

$$\mathcal{R}_j(\theta, d) \;=\; \rho \qquad \text{for } |d_j| \geq D_j \equiv 1/\theta_j\ . \tag{5.13}$$

This is zero for both $\mathcal{R}_j^{(1)}$ and $\mathcal{R}_j^{(2)}$, so these models may lead to a sparse R, see Figure 5.6. As regards approximation characteristics, we see that

$$\frac{\partial \mathcal{R}_j^{(1)}}{\partial d}\,(\theta, D_j) = 0, \qquad \frac{\partial \mathcal{R}_j^{(2)}}{\partial d}\,(\theta, D_j) = -1.5\theta_j\ .$$

This implies that $\mathcal{R}_j^{(1)}$ is better suited when the Kriging model is used to approximate a continuously differentiable function Υ. This is in contrast to the use in statistics: In [11] it is shown that the two parameters in (5.11) have the statistical interpretation $\rho = \mathrm{corr}(\Upsilon(0), \Upsilon(D_j))$ and $\omega = \mathrm{corr}(\Upsilon'(0), \Upsilon'(D_j))$, and that a proper correlation model (that can lead to a positive definite R) is obtained only if the parameters satisfy

$$\rho \in [0,1], \quad \omega \in [0,1] \quad \text{and} \quad \rho \ \geq \ \frac{5\omega^2 + 8\omega - 1}{\omega^2 + 4\omega + 7} \ . \tag{5.14}$$

These conditions are satisfied by $\mathcal{R}_j^{(2)}$ but not by $\mathcal{R}_j^{(1)}$.

For the test problem (5.3) we get the results shown in Figure 5.9. Both models have a θ-region (about $[0.32, 2.5]$) where R is not SPD:

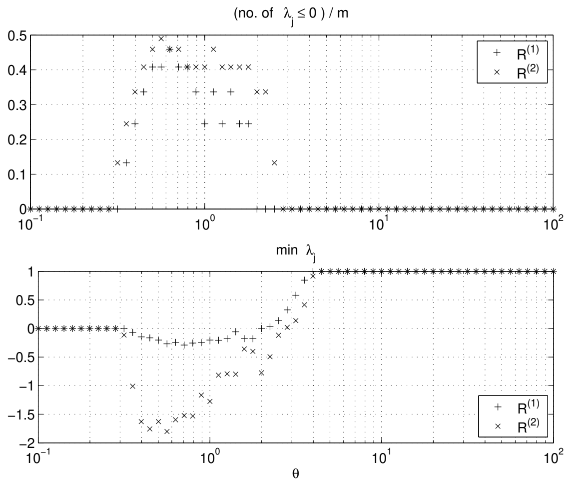

Figure 5.9. *R given by* (5.12) *and* (5.3) *with* $q = 14$

up to almost half of the eigenvalues can be negative. The bottom plot shows that if we should use a modification like (4.14), then we would have to use $\mu \simeq 0.29$ for $\mathcal{R}^{(1)}$ and $\mu \simeq 1.8$ for $\mathcal{R}^{(2)}$. Thus, also

with respect to providing a proper correlation matrix, model $\mathcal{R}^{(1)}$ is preferable to $\mathcal{R}^{(2)}$, but none of them is fully suited to cover the desired range of θ-values.

We use CUBIC to designate $\mathcal{R}^{(1)}$. Its sparsity properties are illustrated in Figure 5.6, and it is implemented as `corrcubic` in the DACE Toolbox.

5.4. Cubic Spline

We are interested in a correlation model that

- Shares the properties of GAUSS and $\mathcal{R}^{(1)}$, (5.2) and (5.12), that it is suited for approximation of continuously differentiable functions.
- Can generate correlation matrices that are sparse and are not too ill conditioned.
- Is easy to evaluate.

A *cubic spline*, see e.g. [4] or [5], satisfies these demands. We experimented with several formulations, and settled for the following: As in (5.11) we let

$$\xi_j \ = \ \theta_j |d_j| \ , \tag{5.15}$$

and define a cubic spline $\mathcal{R}^{(a)}$ on the *knots* $\{0, a, 1\}$ with $0<a<1$. The piecewise 3rd order polynomial

$$\mathcal{R}_j^{(a)}(\theta, d) \ = \ \begin{cases} 1 - \dfrac{3}{a}\,\xi_j^2 + \dfrac{1+a}{a^2}\,\xi_j^3 & \text{for} \ \ 0 \leq \xi_j \leq a \\ \dfrac{1}{1-a}\,(1-\xi_j)^3 & \text{for} \ \ a < \xi_j < 1 \\ 0 & \text{for} \ \ \xi_j \geq 1 \end{cases} \tag{5.16}$$

is twice continuously differentiable, and is therefore a cubic spline. It satisfies the boundary conditions

$$g(0) = 1, \quad g'(0) = g(1) = g'(1) = g''(1) = 0 \ ,$$

with $g(\xi_j) = \mathcal{R}_j^{(a)}(\theta, d)$. Figure 5.10 shows the spline for two a-values

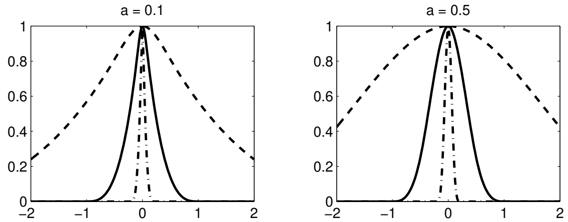

Figure 5.10. *Cubic spline models* (5.16) *for* $|d_j| \le 2$.
Dashed, full and dash-dotted line: $\theta_j = 0.2,\ 1,\ 5$.

The spline has an inflection point at $\xi = a/(1+a)$, which decreases as $a \searrow 0$. This is equivalent with the peak becoming narrower, and the spline approaches the EXP model in Figure 5.1. It is also reflected in the conditioning of the correlation matrix, as shown in Figure 5.11.

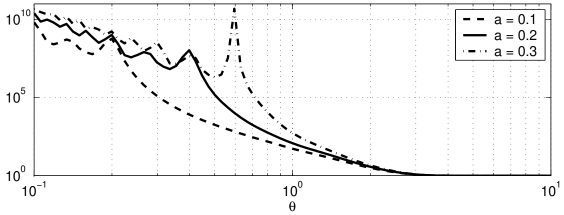

Figure 5.11. *Condition number of* $\widehat{R}$ *given by* (5.4), (5.16) *and* (5.3) *with* $q = 14$

Compared with Figure 5.2 the cubic spline results are between the EXP and GAUSS results. Generally, a smaller a-value gives a smaller condition number. The flutter at the left hand end probably has the same explanation as discussed below in connection with (5.21). The amplitude of the last peak in the flutter seems to grow with a, and further investigation showed that there is a small interval around $a = 0.4$, where R is indefinite in a small θ-interval, similar to the cubic functions in Section 5.3. Further, an investigation of the error as in

Section 5.5 showed insignificant difference between the three a-values in Figure 5.11. As a compromise between robustness and smoothness we decided to use $a = 0.2$, and (5.16) takes the form

$$\mathcal{R}_j(\theta, d) = \begin{cases} 1 - 15\xi_j^2 + 30\xi_j^3 & \text{for } 0 \le \xi_j \le 0.2 \\ 1.25(1-\xi_j)^3 & \text{for } 0.2 < \xi_j < 1 \\ 0 & \text{for } \xi_j \ge 1 \end{cases} \tag{5.17}$$

We refer to this as SPLINE. Its sparsity properties are illustrated in Figure 5.6, and it is implemented as `corrspline` in the DACE Toolbox.

Figure 5.12 shows the corresponding objective function ψ, cf. Figures 5.5 and 5.7. Note that for the faster oscillating Υ_2 the objective function ψ has local minima to the left of the global minimum.

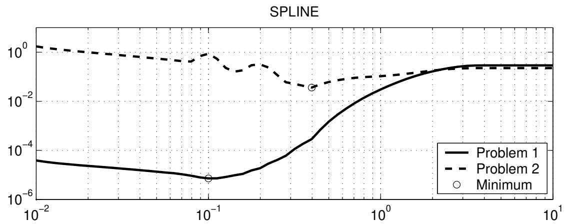

Figure 5.12. $\psi(\theta) = |\widehat{R}(\theta)|^{\frac{1}{m}} \cdot \sigma^2(\theta)$ *for* $0.01 \le \theta \le 10$.
$\widehat{R}$ *given by* (5.17), (5.3) *and* (5.4) *with* $q = 14$

5.5. Approximation Error

In this section we look at the error of the Kriging estimator $\hat{y}(x)$ (3.1) as an approximation to a function $\Upsilon : \mathbb{R}^n \mapsto \mathbb{R}$. The estimator is determined by a given set of design points, $(s_i, \Upsilon(s_i))$, $i{=}1,\ldots,m$.

We use the design points defined in (5.3), and to avoid possible boundary effects, we evaluate the predictor on an interior subregion,

$$\text{Test sites:} \quad \mathcal{T} = 41{\times}41 \text{ equidistant mesh over } [1,4]{\times}[2,8] \tag{5.18}$$

For a chosen correlation model with parameters θ we define the error measure

$$E_k(\theta) \;=\; \max_{x\in\mathcal{T}} |\hat{y}_{(k)}(x) - \Upsilon_k(x)| \;, \tag{5.19}$$

and the measure for the estimated MSE, (3.3),

$$\Phi_k(\theta) \;=\; \big(\max_{x\in\mathcal{T}} |\varphi_{(k)}(x)|\big)^{0.5} \;. \tag{5.20}$$

Index (k) indicates that the Kriging model is fitted to data from Υ_k. The squareroot is included in (5.20) to ease comparison of the two error measures.

Figure 5.13 shows the error measures for five correlation models. Note the remarkable agreement between the two error measures, as regards the best θ-value.

Figure 5.14 shows how the models GAUSS, DROP and SPLINE converge as the number of design sites increases. Note the fast convergence of GAUSS, while the other two models have almost identical and slower convergence.

There are two disappointing characteristics with the results from the SPLINE model:

1. The flutter in E_k.

2. The slow convergence.

Complaint no. 1 is shared by the CUBIC model but not by the other three models. It is caused by the product form of the correlation (5.1). If we change (5.15) to

$$\xi \;=\; \|\,\theta\odot d\,\| \;, \tag{5.21}$$

and use (5.16) with ξ_j replaced by ξ to get $\mathcal{R}(\theta,d)$, then the flutter disappears. However, we did not pursue this line further because a has to be much smaller in order to ensure SPD in a reasonable θ-range, and the error is of the same order of magnitude as with the model defined by (5.16). The other models shown in Figure 5.13 have (5.21)

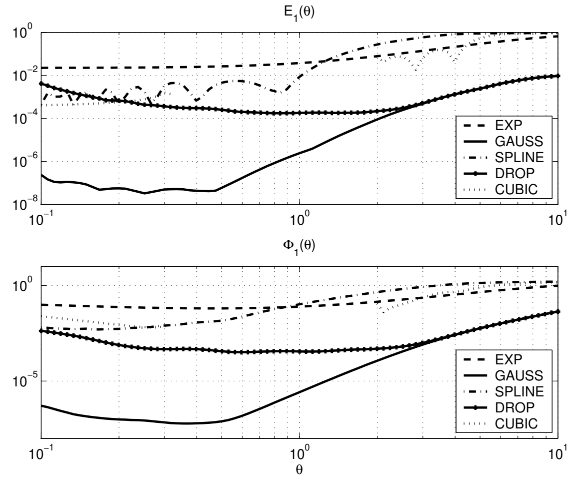

Figure 5.13. *Error measures* (5.19) *and* (5.20) *for Problem 1 with* EXP *and* GAUSS (5.2)*;* SPLINE (5.17)*;* DROP (5.6) *with* $\tau = 10^{-6}$ *and* μ *given by* (5.10)*;* CUBIC*:* $\mathcal{R}^{(1)}$ (5.12). $q = 14$

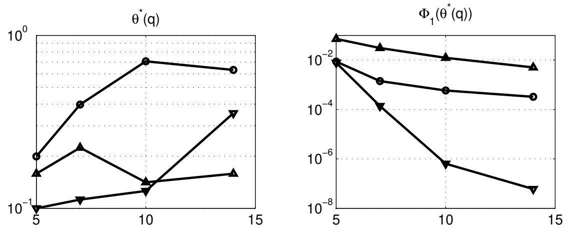

Figure 5.14. θ^* *and* $\Phi_1(\theta^*)$ *as functions of* $q = \sqrt{m}$ *in* (5.3). GAUSS*:* $-\triangledown-$, DROP*:* $-\circ-$, SPLINE*:* $-\triangle-$

built in, since

$$\begin{aligned}
\text{EXP}: \quad \mathcal{R}(\theta,x,s) &= \textstyle\prod_j \exp(-\theta_j|d_j|) \;=\; \exp(-\|\theta\odot d\|_1) \\
\text{GAUSS}: \quad \mathcal{R}(\theta,x,s) &= \textstyle\prod_j \exp(-\theta_j d_j^2) \;=\; \exp(-\|\theta^{1/2}\odot d\|_2^2)
\end{aligned}$$

Complaint no. 2, or – maybe more accurately: why GAUSS performs so well – is harder to explain. We have not found any prior satisfactory explanation of this, but here is our attempt at a partial explanation: Consider (3.1) at a point close to the design site s_k,

$$\hat{y}(s_k+h) \;=\; f(s_k+h)\beta^* + \gamma^{*\top} r(s_k+h) \ .$$

With the regression model $\mathcal{F}(x) = 1$ and introducing (3.2) and (3.4) this takes the form

$$\begin{aligned} \hat{y}(s_k+h) \;&=\; \beta^* + (Y-\beta^*)^\top R^{-1} r(s_k+h) \\ &\simeq\; \beta^* + (Y-\beta^*)^\top R^{-1}\big(r(s_k) + J_r(s_k)h\big) \ , \end{aligned} \tag{5.22}$$

where J_r is the Jacobian. It follows that

$$\begin{aligned} \hat{y}(s_k) \;&=\; \beta^* + (Y-\beta^*)^\top R^{-1} R_{:,k} \\ &=\; \beta^* + (Y-\beta^*)^\top e_k \\ &=\; y_k \\ &=\; \Upsilon(s_k) \ , \end{aligned} \tag{5.23}$$

i.e., the Kriging predictor interpolates the design points.

Figure 5.15 shows the vectors γ^* and $r(s_k)$ for the GAUSS and SPLINE models, computed with a θ-value close to the optimizer, cf. Figure 5.14, and s_k given in the legend. For comparison, $\beta^*_{\text{GSS}} = -0.3588$, $\beta^*_{\text{SPL}} = -0.2770$ and the function value normalized by (2.1) is $(\Upsilon_1(s_k) - \mu(Y))/\sigma(Y) = 0.5565$.

For both models the residual vector $Y - \beta^* e$ has elements of order of magnitude 1, and the ill conditioning of R_{GSS} implies that its inverse has large elements. This is reflected in the components of γ^*_{GSS}. In the computation of $\hat{y}(s_k)$ (5.23) there will be serious cancellation error, and this is verified by computation. For the point s_k given in Figure 5.15 we find

$$|\hat{y}_{\text{GSS}}(s_k) - \Upsilon(s_k)| \;=\; \texttt{6.99e-9}, \quad |\hat{y}_{\text{SPL}}(s_k) - \Upsilon(s_k)| \;=\; \texttt{1.52e-13} \ .$$

The SPLINE result is as accurate as we can hope for with $\varepsilon_{\text{M}} = \texttt{2.22e-16}$.

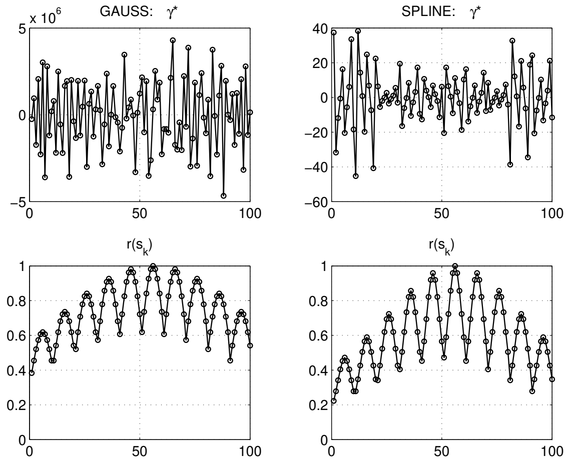

Figure 5.15. *Factors in* (5.23) *for Problem 1 in* (5.3) *with* $q = 10$. $\theta = 0.16$, $s_k = [\frac{25}{9} \ \ \frac{50}{9}]$

Next, we look at the behaviour close to s_k, as expressed by (5.22). Introducing (5.23) we can write it in the form

$$\begin{aligned} \hat{y}(s_k+h) \;&\simeq\; \Upsilon(s_k) + g(s_k)^\top h \\ \text{with} \quad g_j \;&=\; \frac{\sigma(\overline{Y})}{\sigma(\overline{S}_{:,j})}\,(J_r)^\top_{:,j}\gamma^* \ . \end{aligned} \tag{5.24}$$

In terms of normalized variables the Jacobians of the two models are

$$\begin{aligned} \text{GAUSS}: \quad \big(J_r(s_k)\big)_{ij} \;&=\; -2\theta_j d_{ij} R_{ik} \ , \\ \text{SPLINE}: \quad \big(J_r(s_k)\big)_{ij} \;&=\; \theta_j \text{sign}(d_{ij})\Omega(\xi_j)\textstyle\prod_{\ell\neq j} \mathcal{R}_\ell(\theta, d_i) \ , \end{aligned}$$

Here, d_{ij} is the jth component in the vector $d_i = s_k - s_i$, and

$$\Omega(\xi) \;=\; \begin{cases} -30\xi + 90\xi^2 & \text{for} \ \ 0 \le \xi \le 0.2 \\ -3.75(1-\xi)^2 & \text{for} \ \ 0.2 < \xi_j < 1 \\ \quad 0 & \text{for} \ \ \xi \ge 1 \end{cases}$$

With the data from above we get

$$g^{\text{GSS}} = \begin{bmatrix} 0.0322 \\ -0.4596 \end{bmatrix}, \qquad g^{\text{SPL}} = \begin{bmatrix} 0.0359 \\ -0.4614 \end{bmatrix}.$$

For comparison, the gradient of Υ agrees with g^{GSS} on the four decimals shown; the maximum relative difference is `7.8e-7`. This means that close to s_k the Kriging model based on GAUSS is very close to a first order Taylor expansion, while the SPLINE model disagrees on the second decimal in the gradient.

This behaviour is worth further investigation. It should be mentioned that we have experimented also with other problems and with other choices of the regression function $\mathcal{F}$, and got similar results.

6. Optimize Parameters

By comparing Figures 5.5, 5.12 and 5.13 we see a good agreement between the minimizing θ-value for the error measure Φ (5.20) and the optimizer for the objective function ψ defined in (2.5).

Figure 6.1 shows that the smooth behavior of $\psi(\theta)$ that was found in the isotropic case is also found when the the components of θ are allowed to differ.

An easier identification of the minimizer is obtained if we look at *level curves* for the objective function, see Figure 6.2.

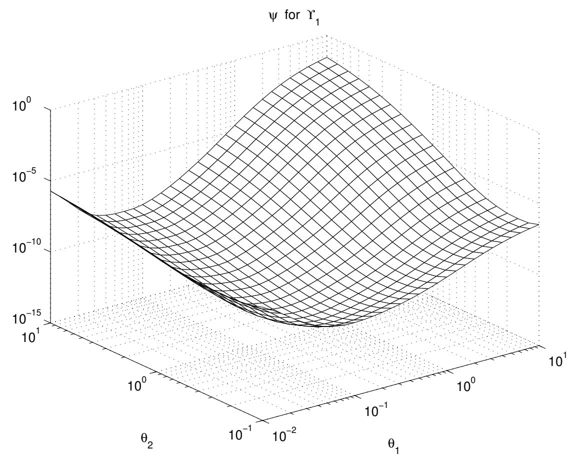

Figure 6.1. *$\psi(\theta)$ (2.5) for Υ_1 in (5.3) with $q = 14$. $\theta \in [0.01, 10]\times[0.1, 10]$.* GAUSS *model*

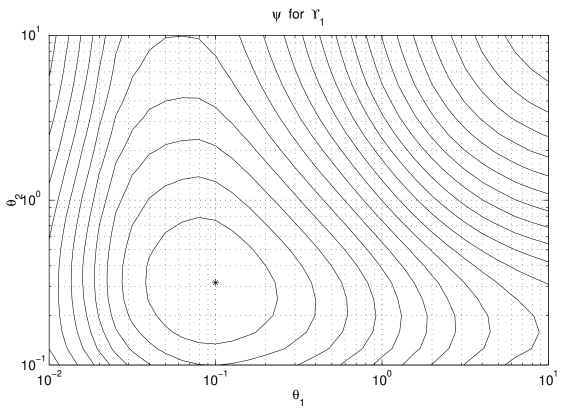

Figure 6.2. *Level curves for $\psi(\theta)$ for Υ_1 in (5.3) with $q = 14$.* GAUSS *model. The asterisk marks the minimizer, $\theta^* = [0.100 \;\; 0.316]$*

6.1. Algorithm

Let the parameter vector θ have q components, e.g., $q=1$ for the isotropic models treated in Section 5 and $q=2$ in Figures 6.1 and 6.1. We seek (an approximation to) θ^* in the region $0 < \ell_j \le \theta_j \le u_j$, $j=1,\ldots,q$. The program `dacefit` can handle the following cases and mixtures of them,

- Some θ_j are fixed	indicated by	$\ell_j = u_j$
- Warm start	indicated by	$\ell_j \le \theta_j \le u_j$
- Cold start	indicated by	$\theta_j < \ell_j$ or $\theta_j > u_j$

We shall only discuss cases where there is at least one unknown parameter.

The algorithm for minimizing the function ψ, (2.5), should take into account that

1. each evaluation of ψ is expensive. It involves the evaluation and factorization of $R(\theta)$ and the solution of (2.8), but
2. Figures 5.5, 5.12, 5.13 and 6.2 indicate that there is no point in finding the minimizer with great accuracy, and
3. the function is well behaved – at least if we approach the minimizer from above, but
4. computation of the gradient of ψ with respect to the components of θ is possible, but would involve considerable extra effort.

These considerations lead us to choose a pattern search method. More specific, we use the following modified version of the *Hooke & Jeeves* method, see e.g. [9, Section 2.4]. Rather than the usual approach, where the parameters get absolute changes, we work with a vector Δ of relative changes.

The main algorithm is

Algorithm Optimize θ
Given $\theta^{(0)}$, ℓ, u
$[\theta, \Delta] := \mathsf{start}(\theta^{(0)}, \ell, u)$
for $k = 1, \ldots, k_{\max}$ 1°
  $\widehat{\theta} := \theta$
  $\theta := \mathsf{explore}(\theta, \Delta, \ell, u)$
  $[\theta, \Delta] := \mathsf{move}(\widehat{\theta}, \theta, \Delta)$
  $\Delta := \mathsf{rotate}(\Delta)$ 2°
end

Remarks

1° Experiments showed that $k_{\max} = \max\{2,\ \min\{q, 4\}\}$ is a good compromise between efficiency and desired accuracy.

2° In order to avoid "backtracking", the components of Δ are different, and we rotate the components by taking the indices in the order $2, \ldots, q, 1$.

The easy case for the starting algorithm is when there is only one free parameter: start the search close to the upper bound. If we have two or more cold start parameters, we have to take into account that ψ may have several local minima, as illustrated in Figures 6.4 and 6.5. The probability of landing in a wrong local minimum was considerably reduced by using an elaborate starting procedure. Except for check of legal inputs ($0 < \ell_j \le u_j$ etc.) this has the form

Algorithm $[\theta, \Delta] := \mathsf{start}(\theta^{(0)}, \ell, u)$
$\theta := \theta^{(0)}$; $\mathcal{N} := \emptyset$ 3°
for $j = 1, \ldots, q$
  if $\ell_j = u_j$ **then** $\Delta_j := 1$; $\theta_j := u_j$ 4°
  else
    $\Delta_j := 2^{j/(q+2)}$ 2°
    if $\theta_j < \ell_j$ **or** $\theta_j > u_j$ **then**
      $\theta_j := (\ell_j u_j^7)^{1/8}$; $\mathcal{N} := \mathcal{N} \cup j$ 5°
    end
  end
end

if $\#\mathcal{N} > 1$ **then**
$\widehat{\theta} := \theta; \quad \mathcal{J} := \mathcal{N}_1$ 6°
for $k = 1, \ldots, \#\mathcal{N}$
$j := \mathcal{N}_k; \quad \overline{\theta} := \widehat{\theta}$
$v := e; \quad v_{\mathcal{N}} := \frac{1}{2}; \quad v_j := \frac{1}{16}$
$\alpha := \min\{\ln(\ell_{\mathcal{N}} \oslash \theta_{\mathcal{N}}) \oslash \ln v_{\mathcal{N}}\}$
$v := v^{\alpha/5}$ 7°
for $i = 1, 2, 3, 4$
$\vartheta := v^i \odot \widehat{\theta}$
if $\psi(\vartheta) \le \psi(\overline{\theta})$ **then**
$\overline{\theta} := \vartheta;$
if $\psi(\vartheta) \le \psi(\theta)$ **then** $\theta := \vartheta; \quad \mathcal{J} := j$
else **break** 8°
end
end
Swap Δ_1 and $\Delta_{\mathcal{J}}$ 9°
end

Remarks

3° $\mathcal{N}$ is the set of indices for which a proper starting value is not given.

4° Equality constraint.

5° No proper starting point given. Choose a point close to the upper bound.

6° For each component of θ without a proper starting value, try up to four points with that component reduced considerably faster than the others; cf. 7° and Figures 6.3 - 6.5.

7° v is determined so that $v^5 \odot \widehat{\theta}$ hits a lower bound.

8° Stop the i-loop: we have passed a local minimum.

9° Direction number $\mathcal{J}$ had the largest step in the introductory search, and should have the smallest step now; cf. 2°.

The loop in optimize starts by an *explore* step, where each free pa-

rameter θ_j in turn is increased in an attempt to reduce the objective function. If this fails, then a decreased θ_j-value is tried. Parameter values at a bound are only allowed one-sided change.

Algorithm $\theta :=$ explore$(\theta, \Delta, \ell, u)$
for $j = 1, \ldots, q$
if $\ell_j < u_j$ **then**
$\vartheta := \theta$
if $\theta_j = \ell_j$ **then** $\vartheta_j := \ell_j * \Delta_j^{1/2}; \quad$ atbd := **true**
elseif $\theta_j = u_j$ **then** $\vartheta_j := u_j / \Delta_j^{1/2}; \quad$ atbd := **true**
else $\vartheta_j := \min\{\theta_j * \Delta_j,\ u_j\}; \quad$ atbd := **false**
if $\psi(\vartheta) < \psi(\theta)$ **then** $\theta := \vartheta$
elseif **not** atbd **then**
$\vartheta_j := \max\{\theta_j / \Delta_j,\ \ell_j\}$
if $\psi(\vartheta) < \psi(\theta)$ **then** $\theta := \vartheta$
end
end
end

Finally, the change from $\widehat{\theta}$ to θ may indicate a *pattern*. This is investigated in

Algorithm $[\theta, \Delta] :=$ move$(\widehat{\theta}, \theta, \Delta)$
if $\widehat{\theta} = \theta$ **then** $\Delta := \Delta^{1/5}$
else
$v := \theta \oslash \widehat{\theta}; \quad$ notstop := **true**
while notstop
$\vartheta := \theta \odot v$
if **any**$(\vartheta_j \le \ell_j$ **or** $\vartheta_j \ge u_j)$ **then**
notstop := **false**
$\alpha^* := \max\{\alpha \mid \theta_j v_j^{\alpha} \ge \ell_j$ **and** $\theta_j v_j^{\alpha} \le u_j\}$
$\vartheta := \theta \odot v^{\alpha^*}$
end
if $\psi(\vartheta) < \psi(\theta)$ **then** $\theta := \vartheta; \quad v := v^2$

else notstop := **false**
end
$\Delta := \Delta^{1/4}$
end

The performance of the algorithm is illustrated in Figures 6.3 - 6.5

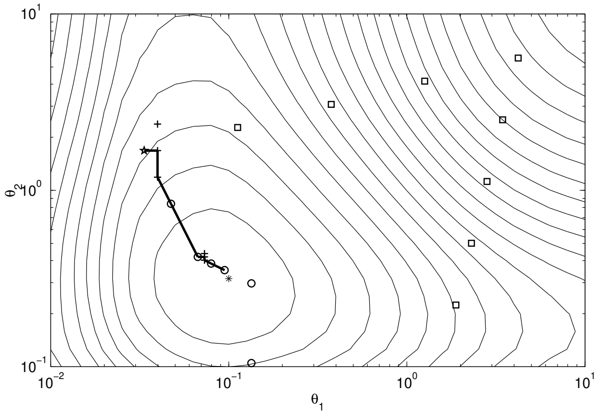

Figure 6.3. *Search path.* GAUSS *model with* Υ_1 *from* (5.3), $q = 14$.
Squares: Points tried in start. *Star: starting point.*
Plus: explore *step. Ring:* move *step.*
Asterisk: as in Figure 6.2

The swapping of the coordinates in Figure 6.5 was made to illustrate that the algorithm may also choose to reduce θ_2 faster.

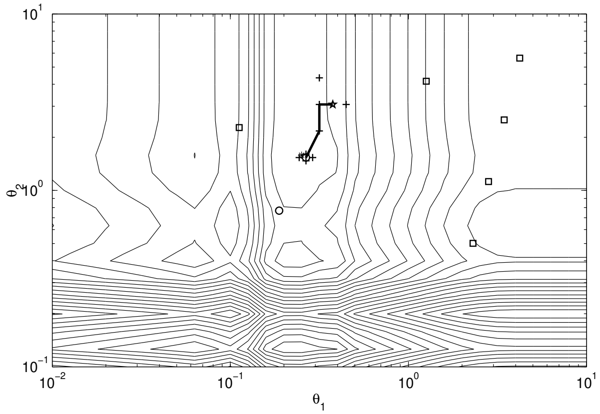

Figure 6.4. *Search path.* SPLINE *model with* Υ_1 *from* (5.3), $q = 14$.
Legend as in Figure 6.3

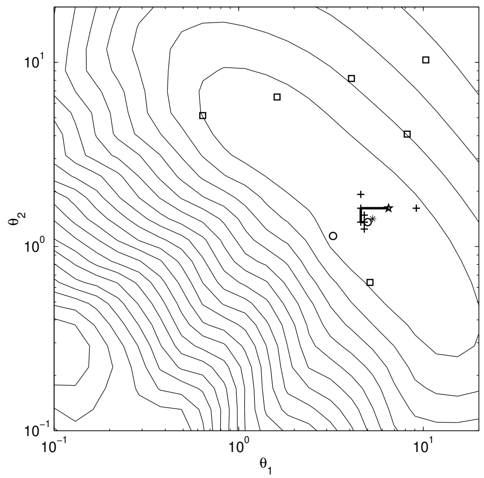

Figure 6.5. *Search path for the data from* data1 *in the DACE Toolbox, with swapped coordinates.* $m = 75$. GAUSS *model.*
Legend as in Figure 6.3

6.2. Testing

We have tested the algorithm on the 5 problems given in Table 6.1. Problems 4 and 5 are immediate generalizations of (5.3).

pno	Description	ℓ	u
1	Data given in `data1` in the DACE Toolbox. m=75. Test sites: 31^2 grid on $[20, 80]^2$. $\|Y\|_\infty = 44.7$.	.1 .1	20 20
2	Data with Υ_1 as defined by (5.3) with q=14, m=196. Test sites given by (5.18). $\|Y\|_\infty \simeq 1$.	.01 .1	10 10
3	Data with Υ_2 as defined by (5.3) with q=14, m=196. Test sites given by (5.18). $\|Y\|_\infty \simeq 1$.	.01 .1	10 10
4	Data with $\Upsilon_1(x) = \prod_j \sin \frac{1}{2} x_j$, where the design sites are from a uniform q^3 grid on $[0, 5] \times [0, 10] \times [0, 15]$. q=10, m=1000. Test sites: 11^3 grid on $[1, 4]\times[2, 8]\times[3, 12]$ $\|Y\|_\infty \simeq 1$.	.01 .1 .1	10 10 10
5	Data with $\Upsilon_2(x) = \prod_j \sin 2x_j$. Design and test sites as in problem 4. $\|Y\|_\infty \simeq 1$.	.01 .1 .1	10 10 10

Table 6.1. *Test problems*

For each problem we use both the GAUSS and the SPLINE model, and in each case we find both the isotropic and anisotropic solution. In the latter case we also test the warm start capability, by refining the θ. In the testing we also give the error measure Φ defined by (5.20).

Further, we give results from replacing the algorithm of Section 6.1 with `fminsearch` from the MATLAB Optimization Toolbox, Version 2. To be able to make a fair comparison, we let it work on the variables $\zeta = \ln \theta$, give it the same starting point, $\zeta_j = (\ln \ell_j + 7 \ln u_j)/8$, and use the very coarse stopping criteria given by


```
optimset('TolX',.005, 'MaxIter',100*q, ...
         'MaxFunEvals',500*q)
```

pno	Meth.	nval	θ^*	$\psi(\theta^*)$	$\Phi(\theta^*)$
1	df	12	2.58	4.33e-02	1.27
	fms	23	2.67	4.31e-02	1.31
	grid	41	2.82	2.43e-01	1.39
2	df	13	.166	1.50e-10	1.17e-07
	fms	29	.184	1.46e-10	1.04e-07
	grid	61	.178	4.25e-11	1.07e-07
3	df	11	1.33	1.11e-02	7.46e-04
	fms	23	1.33	1.11e-02	7.45e-04
	grid	61	1.26	2.56e-03	6.92e-04
4	df	14	.264	7.06e-08	1.42e-05
	fms	27	.315	5.98e-08	1.70e-05
	grid	61	.316	8.19e-09	1.71e-05
5	df	5	10.0	2.68e-01	3.48e-01
	fms	37	10.0	2.68e-01	3.48e-01
	grid	61	10.0	2.68e-01	3.48e-01

Table 6.2. *Isotropic* GAUSS *model.*

`df` *The algorithm from Section 6.1, implemented in* `dacefit`
`fms` `fminsearch` *as described above*
grid *Minimum over logarithmic equidistant grid over* $[\ell, u]$.
nval *grid points*

The results in Tables 6.2 - 6.5 give rise to the following remarks,

1. The algorithm used in `dacefit` is robust. In all cases it finds the right local minimum.

2. As expected, it is easier to optimize θ for an isotropic model than an anisotropic model, but the latter normally gives a better result in terms of smaller values both for ψ and Φ.

pno	Meth.	nval	θ^*	$\psi(\theta^*)$	$\Phi(\theta^*)$
1	`df`	13	.203	2.00e-02	3.56e-01
	`fms`	31	.195	1.97e-02	3.42e-01
	grid	41	.200	1.11e-01	3.50e-01
2	`df`	10	.111	2.51e-05	5.75e-03
	`fms`	30	.101	2.46e-05	6.15e-03
	grid	61	.100	7.21e-06	6.26e-03
3	`df`	13	.418	1.78e-01	1.20e-01
	`fms`	27	.394	1.59e-01	1.32e-01
	grid	61	.398	3.62e-02	1.30e-01
4	`df`	5	4.22	9.99e-01	5.44e-01
	`fms`	15	4.22	9.99e-01	5.44e-01
	grid	61	.141	1.50e-05	2.75e-02
5	`df`	5	4.22	9.99e-01	5.16e-01
	`fms`	15	4.22	9.99e-01	5.16e-01
	grid	61	3.16	1.23e-01	4.90e-01

Table 6.3. *Isotropic* SPLINE *model.*
Legend as in Table 6.2

3. Generally a smaller value for the objective function ψ corresponds to a smaller value of the error measure Φ, but there are enough exceptions to this rule to confirm our statement that is does not make sense to compute the minimizer with higher accuracy.
4. The results for the GAUSS model with Problem 3 are surprising: the optimal ψ is *decreased* by a factor 100 when we change from isotropic to anisotropic model, but the error measure is *increased* by the same factor. This should be investigated further.
5. Generally, `fminsearch` gives essentially the same solution as our special purpose algorithm, with the ratio of function evaluations varying between 2 and 9. In the case illustrated in Figure 6.5 (pno=1 in Table 6.3) `fminsearch` finds a wrong local minimum (which leads to a smaller error measure, however.)

pno	Method	nval	θ^*	$\psi(\theta^*)$	$\Phi(\theta^*)$
1	`df`	16	1.36, 4.79	3.71e-02	1.17
	warm	9	1.36, 4.96	3.71e-02	1.17
	`fms`	59	4.24, 1.39	4.00e-02	9.77e-01
	grid	441	1.41, 5.32	2.11e-01	1.21
2	`df`	21	.0947, .353	6.44e-11	7.36e-08
	warm	10	.0884, .341	6.30e-11	7.77e-08
	`fms`	81	.0911, .304	6.16e-11	8.17e-08
	grid	651	.1000, .316	1.83e-11	7.63e-08
3	`df`	13	.487, 4.16	6.71e-04	5.32e-02
	warm	11	.408, 2.17	6.69e-04	4.46e-02
	`fms`	74	.491, 3.85	6.68e-04	4.28e-02
	grid	651	.501, 3.98	1.52e-04	4.44e-02
4	`df`	38	0.0670, .277, .554	7.33e-09	1.39e-04
	warm	19	.0783, .302, .783	6.03e-09	3.43e-04
	`fms`	174	.0806, .296, .754	6.01e-09	2.99e-04
	grid	1936	.1000, .251, .631	9.48e-10	2.99e-04
5	`df`	27	.273, 3.07, 3.07	4.75e-01	1.31
	warm	19	.273, 3.07, 3.07	4.75e-01	1.31
	`fms`	30	4.22, 5.62, 5.62	9.99e-01	5.24e-01
	grid	1936	.251, 3.98, 3.98	6.04e-02	1.65

Table 6.4. *Results with anisotropic* GAUSS *model.*
Legend as in Table 6.2

6. With the anisotropic SPLINE model in Table 6.4 `fminsearch` stops prematurely for pno $= 2, 3, 5$, probably because `TolX` was chosen too large.
7. In Section 5.5 we found that GAUSS was surprisingly accurate and much better than SPLINE. Comparing Tables 6.3 and 6.4 we see that for pno=1 we get more accurate results by means of the SPLINE correlation model.

pno	Method	nval	θ^*	$\psi(\theta^*)$	$\Phi(\theta^*)$
1	`df`	19	.100, .255	`1.89e-02`	`2.54e-01`
	warm	7	.100, .361	`1.53e-02`	`2.59e-01`
	`fms`	96	.100, .366	`1.52e-02`	`2.58e-01`
	grid	441	.100, .376	`7.98e-02`	`2.58e-01`
2	`df`	23	.0670, .148	`2.01e-05`	`7.88e-03`
	warm	11	.0670, .114	`1.86e-05`	`7.10e-03`
	`fms`	24	4.22, 5.62	`9.95e-01`	1.08
	grid	651	.0631, .126	`5.85e-06`	`7.88e-03`
3	`df`	17	.266, 1.54	`1.20e-01`	`4.85e-01`
	warm	10	.266, 1.48	`1.20e-01`	`4.64e-01`
	`fms`	24	4.22, 5.62	`9.95e-01`	`9.55e-01`
	grid	651	.251, 1.58	`2.79e-02`	`5.39e-01`
4	`df`	19	.795, 10.0, 10.0	`3.44e-01`	`8.18e-01`
	warm	14	0.782, 10.0, 10.0	`3.44e-01`	`8.40e-01`
	`fms`	191	0.783, 10.0, 10.0	`3.44e-01`	`8.39e-01`
	grid	1936	.631, 10.0, 10.0	`4.68e-02`	1.26
5	`df`	27	.273, 3.07, 3.07	`4.75e-01`	1.31
	warm	19	.273, 3.07, 3.07	`4.75e-01`	1.31
	`fms`	30	4.22, 5.62, 5.62	`9.99e-01`	`5.24e-01`
	grid	1936	.251, 3.98, 3.98	`6.04e-02`	1.65

Table 6.5. *Results with anisotropic* SPLINE *model. Legend as in Table 6.2*

6.3. Computing Time

The computational effort in `dacefit` is dominated by the Cholesky factorization (2.6), which is an $\mathcal{O}(m^3)$ process. This is performed *nval* times, where *nval* is the number of function evaluations during the optimization. From Section 6.2 it follows that *nval* grows slowly with the number of free elements in θ, and as a simple model for the execution time we can take

$$T_{\texttt{dacefit}} \simeq a \cdot m^3 \ . \qquad (6.1)$$

Suppose that we want to compute ν values with `predictor`, at the sites $x \in \mathbb{R}^{\nu\times n}$. This involves the computation of the $\nu\times p$ regression matrix $\mathcal{F}(x)$ and the $m\times\nu$ correlation matrix $r(x)$ and performing the

inner products in (3.1). The effort grows linearly with ν and m, and

$$T^{\hat{y}}_{\texttt{predictor}} \simeq b \cdot \nu \cdot m \ . \qquad (6.2)$$

If we also want the MSE, we further have to perform the $\mathcal{O}(\nu \cdot m^2)$ transformation $\widetilde{r} = C^{-1} r(x)$, and

$$T^{\hat{y},\varphi}_{\texttt{predictor}} \simeq c \cdot \nu \cdot m^2 \ . \qquad (6.3)$$

These considerations are corroborated by Figure 6.6.

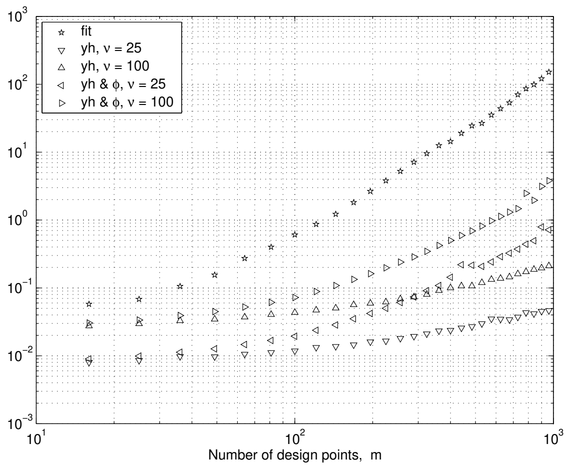

Figure 6.6. *Times in seconds on a Sunfire 10k for* `dacefit` *and* `predictor`*.*
Problems generated by Υ_1 in (5.3) for $q = 4, 5, \ldots, 31$.
GAUSS *model with $\ell = [.01,\ .01]$, $u = [10,\ 10]$*

7. Conclusion

The MATLAB functions in the DACE toolbox version 2.0 seem to work well. As pointed out in this report, there are, however, some open questions that need further investigation

- Why is GAUSS such a good correlation model for Kriging a smooth function?
- Is it possible to find a model that combines the good sparsity properties and well-conditioning of the SPLINE model with better approximation properties?
- The surprising results with GAUSS on the test problem (5.3) when we change from isotropic to anisotropic model.

Currently we have the following plans for further items in the toolbox,

- A regression model that uses products of cubic splines in the n dimensions (with bicubic splines as a special example, when $n = 2$).
- An algorithm for optimization of "expensive" functions, where $\hat{y}$ is used as a surrogate for the functions. Basic ideas as described in e.g., [1], [2] and [13].

8. Notation

$m,\ n$	number of design sites and their dimensionality
p	number of basis functions in regression model
q	number of elements in θ
$\mathcal{F}(\beta, x)$	regression model, $\mathcal{F}(\beta, x) = f(x)^\top \beta$
$\mathcal{R}(\theta, w, x)$	correlation function
C	Cholesky factorization of R, $R = C^T C$
f_j	basis function for regression model
f	p-vector, $f(x) = [f_1(x) \ \cdots \ f_p(x)]^\top$
F	expanded design $m \times p$-matrix, see Section 2
$\tilde{F},\ \tilde{Y}$	transformed data, see (2.7)
R	$m \times m$-matrix of stochastic-process correlations
r	m-vector of correlations
S	$m \times n$ matrix of design sites
s_i	ith design site, vector of length n. $s_i^\top = S_{i,:}$
$U \Sigma V^\top$	SVD – Singular Value Decomposition, see (4.2)
v_j	eigenvector, see (4.7)
x	n-dimensional trial point
x_j	jth component in x
$X_{i,:},\ X_{:,j}$	ith row and jth column in matrix X, respectively
Y	m-vector of responses
y_i	response at ith design site, $y_i = \Upsilon(s_i)$
$\hat{y}$	predicted response, see (3.1)
β	p-vector of regression parameters, see (2.10)
γ	$m \times q$-matrix of correlation constants, see (3.1)
θ	parameters of correlation model, q-vector

λ_j	eigenvalue, see (4.7)
σ^2	process variance, see (2.11)
Σ_{jj}	singular value
Υ	background function, $\Upsilon: \mathbb{R}^n \mapsto \mathbb{R}$
$\varphi(x)$	mean squared error of $\hat{y}$, see (3.3)
$\odot$	elementwise (*Hadamard*) multiplication
$\oslash$	elementwise division
MSE	mean squared error, p 5
GAUSS	Gauss correlation model, see (5.2)
SPD	symmetric, positive definite
SPLINE	Cubic spline correlation model, see (5.17)

References

[1] M.H. Bakr, J.W. Bandler, M.A. Ismail, J.E. Rayas-Snchez, Q. J. Zhang, *Neural Space Mapping Optimization for EM-Based Design.* IEEE Trans. Microwave Theory Tech., **48** pp 2307-2315, 2000.

[2] A.J. Booker, J.E. Dennis, P.D. Frank, D.B. Serafini, V. Torczon, M.W. Trosset, *A Rigorous Framework for Optimization of Expensive Functions by Surrogates.* Structural Optimization **17.1**, pp 1-13, 1999.

[3] R. Ababou, A.C. Bagtzoglou, E.F, Wood, *On the Condition Number of Covariance Matrices in Kriging, Estimation, and Simulation of Random Fields.* Mathematical Geology **26.1**, pp 99-133, 1994.

[4] C. de Boor, *A Practical Guide to Splines.* Springer Verlag, New York, USA, 1978.

[5] P. Dierckx, *Curve and Surface Fitting with Splines.* Monographs on Numerical Analysis, Oxford University Press, Oxford, England, 1993.

[6] G. Golub, C. Van Loan, *Matrix Computations.* Johns Hopkins University Press, Baltimore, USA, 3rd edition, 1996.

[7] P.C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems.* SIAM, Philadelphia, PA, 1998.

[8] J.R. Koehler, A.B. Owen, *Computer Experiments.* Handbook of Statistics **13**, pp 261-308, Amsterdam, 1996.

[9] J. Kowalik, M.R. Osborne, *Methods for Unconstrained Optimization Problems.* Elsevier, New York, USA, 1968.

[10] S.N. Lophaven, H.B. Nielsen, J. Søndergaard, *DACE – A Matlab Kriging Toolbox, Version 2.0.* Report IMM-REP-2002-12, Informatics and Mathematical Modelling, Technical University of Denmark, 34 pages, 2002. Available at http://www.imm.dtu.dk/~hbn/publ/TR0212.ps

[11] M. Mitchell, M. Morris, D. Ylvisaker, *Existence of Smoothed Stationary Processes on an Interval.* Stochastic Processes and their Applications **35**, pp 109-119, 1990.

[12] J. Sacks, W.J. Welch, T.J. Mitchell, H.P. Wynn, *Design and Analysis of Computer Experiments*, Statistical Science, vol. 4, no. 4, pp 409-435, 1989.

[13] C.M. Siefert, V. Torczon, M.W. Trosset, *Model-Assisted Pattern Search: Examples.* To appear in Optimization and Engineering, 2002.

[14] J.H. Wilkinson, *The Algebraic Eigenvalue Problem.* Oxford University Press, London, 1965.