Supplementary Material Titled “Stochastic Polynomial Interpolation for Uncertainty Quantification with Computer Experiments”

Matthias H.Y. Tan

*Department of Systems Engineering and Engineering Management,
City University of Hong Kong*

S1. Heuristic Explanation of Good Performance of SIP Fitted with Chebyshev Points

The theory of univariate interpolation can help us to understand the superior performance of the Chebyshev points for the SIP. We first review some results that will be used to explain the good performance of Chebyshev points. Let ζ denote the Lagrange polynomial, i.e., polynomial of degree $P = n - 1$, that interpolates a function F with domain $[-1, 1]$ at n points. Then,

$$\|\zeta - F\|_\infty \leq (1 + \lambda_n) \|\zeta^* - F\|_\infty, \quad (1)$$

where $\|\zeta - F\|_\infty = \sup_{u \in [-1, 1]} |\zeta(u) - F(u)|$, λ_n is the Lebesgue constant, and ζ^* is the minimax polynomial, i.e., the polynomial that minimizes the maximum error. Note that ζ^* depends on F but λ_n depends only on the choice of interpolation points and n . Let λ_n^* denote the minimum Lebesgue constant and λ_n^C denote the Lebesgue constant for the Chebyshev points.

Then, it is known that (Trefethen, 2013)

$$\lambda_n^* \geq 2\pi^{-1} \ln(n) + 0.52 \text{ and } \lambda_n^C \leq 2\pi^{-1} \ln(n) + 1. \quad (2)$$

Thus, $\lambda_n^C - \lambda_n^* \leq 0.5$ and the Chebyshev points can be considered optimal for practical purposes.

We shall now use the above theory to provide a heuristic explanation of the superior performance of Chebyshev points for the SIP. Set $\mathbf{X}_1$ to be the submatrix of the model matrix that consists of the values of the basis functions $\psi_\alpha, \alpha \in \Omega_1 = \{0, 1, \dots, n - 1\}$ evaluated at the design points. Note that in this case, $\mathbf{X}_1$ is a Vandermonde-like matrix, which is invertible (Gautschi, 1983). Let the columns of $\mathbf{X}_2$ consist of values of $\psi_\alpha, \alpha \in \Omega_2 = \{n, \dots, P\}$ evaluated at the design points and partition $\mathbf{X}$ and $\boldsymbol{\beta}$ so that $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2)$ and $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^T, \boldsymbol{\beta}_2^T)^T$. Let

$$\zeta_1(u) = \sum_{\alpha \in \Omega_1} \beta_\alpha \psi_\alpha(u) \text{ and } \zeta_2(u) = \sum_{\alpha \in \Omega_2} \beta_\alpha \psi_\alpha(u). \quad (3)$$

Given $\mathbf{Y}$, μ and $\boldsymbol{\beta}_2$, the coefficient vector $\boldsymbol{\beta}_1$ is given by $\boldsymbol{\beta}_1 = \mathbf{X}_1^{-1}(\mathbf{Y} - \mu \mathbf{1}_n - \mathbf{X}_2 \boldsymbol{\beta}_2)$. Note that $\boldsymbol{\beta}_1$ is the coefficient vector for $\zeta_1(u)$ and $\boldsymbol{\beta}_2$ is the coefficient vector for $\zeta_2(u)$. Thus, for given $\mathbf{Y}$, μ and $\boldsymbol{\beta}_2$, $\zeta_1(u)$ is the Lagrange polynomial that interpolates the function $g(u) = y(u) - \mu - \zeta_2(u)$ at $u_1, \dots, u_n$, where y is the output function. We now look at the problem from the “posterior perspective”. The posterior distribution of $\boldsymbol{\beta}_1$ is the distribution of $\mathbf{X}_1^{-1}(\mathbf{Y} - \mu \mathbf{1}_n - \mathbf{X}_2 \boldsymbol{\beta}_2)$ conditional on $\mathbf{Y}$, and the posterior distribution of $\boldsymbol{\beta}_2$ can easily be obtained from the distribution of $(\mathbf{Y}^T, \boldsymbol{\beta}_2^T)^T | \boldsymbol{\theta}$, which is normal. This means that to simulate $\boldsymbol{\beta} | (\mathbf{Y}, \boldsymbol{\theta})$, we can first simulate $\boldsymbol{\beta}_2$ from $\boldsymbol{\beta}_2 | (\mathbf{Y}, \boldsymbol{\theta})$, and then set $\boldsymbol{\beta}_1 = \mathbf{X}_1^{-1}(\mathbf{Y} - \mu \mathbf{1}_n - \mathbf{X}_2 \boldsymbol{\beta}_2)$. Thus, each sample of $\boldsymbol{\beta}_2$ from the posterior distribution of $\boldsymbol{\beta}_2$ gives a different ζ_2 and hence, a different g and a different ζ_1 that interpolates g at the design points. However, by the discussion in the previous paragraph (in particular, (1) and (2)) and the fact that ζ_1 is a Lagrange polynomial, we expect $\|\mathbf{Y} - y\|_\infty = \|\mu + \zeta_1 + \zeta_2 - y\|_\infty = \|\zeta_1 - g\|_\infty$ to be small for all posterior realizations of $\boldsymbol{\beta}_2$ when Chebyshev points are used. This suggests that the SIP credible interval would tend to be tight around the true output and the predictions would tend to be more accurate when the design is a set of Chebyshev points. On the other hand, when equally spaced points are used, ζ_1 can suffer from the Runge phenomenon for some realizations of $\boldsymbol{\beta}_2$. This explains why the SIP credible intervals are wide near the endpoints in Figures 1c-1f.

S2. Design of Follow-Up Runs

In some situations, it is desired to perform follow-up runs after the initial experiment to reduce interpolation uncertainty. We propose a procedure that adds one run at a time at the point of maximum prediction variance to construct a SIP with sufficiently low interpolation uncertainty. Sequential designs schemes that add follow-up runs in this fashion have been employed in many computer experiment problems (e.g., Gramacy and Lee (2009)).

Let $\mathbf{x}(\mathbf{u})$ be the row vector of polynomial basis functions evaluated at $\mathbf{u}$, i.e., the $\psi_\alpha(\mathbf{u})$'s, and let $Y(\mathbf{u})$ be the output at $\mathbf{u}$. Then, the next design point is chosen to be a point $\mathbf{u}^*$ such that $\text{var}(Y(\mathbf{u}^*)|\mathbf{Y}, \boldsymbol{\theta}) = \max_{\mathbf{u} \in \chi} [\text{var}(Y(\mathbf{u})|\mathbf{Y}, \boldsymbol{\theta})] > 0$. This procedure is repeated until the SIP has sufficiently low maximum prediction variance or is sufficiently accurate. Note that

$$\begin{aligned} \text{var}(Y(\mathbf{u})|\mathbf{Y}, \boldsymbol{\theta}) &= \mathbf{x}(\mathbf{u})\mathbf{S}^{1/2} \left[\mathbf{I} - (\mathbf{S}^{1/2}\mathbf{X}^T)(\mathbf{X}\mathbf{S}^{1/2}\mathbf{S}^{1/2}\mathbf{X}^T)^{-1}(\mathbf{X}\mathbf{S}^{1/2}) \right] \mathbf{S}^{1/2}\mathbf{x}(\mathbf{u})^T \sigma^2 \\ &= \sigma^2 \left\| \left[\mathbf{I} - (\mathbf{S}^{1/2}\mathbf{X}^T)(\mathbf{X}\mathbf{S}^{1/2}\mathbf{S}^{1/2}\mathbf{X}^T)^{-1}(\mathbf{X}\mathbf{S}^{1/2}) \right] \mathbf{S}^{1/2}\mathbf{x}(\mathbf{u})^T \right\|_2^2, \end{aligned} \quad (5)$$

which is, except for the factor σ^2 , the 2-norm of the projection of $\mathbf{S}^{1/2}\mathbf{x}(\mathbf{u})^T$ onto the orthogonal complement of the column space of $\mathbf{S}^{1/2}\mathbf{X}^T$. Thus, $\text{var}(Y(\mathbf{u})|\mathbf{Y}, \boldsymbol{\theta}) > 0$ if and only if $\mathbf{S}^{1/2}\mathbf{x}(\mathbf{u})^T$ does not lie in the column space of $\mathbf{S}^{1/2}\mathbf{X}^T$. Since $\mathbf{S}^{1/2} > 0$, it follows that if $\text{var}(Y(\mathbf{u}^*)|\mathbf{Y}, \boldsymbol{\theta}) > 0$, then $\mathbf{x}(\mathbf{u}^*)^T$ does not lie in the column space of $\mathbf{X}^T$. In other words, $D \cup \{\mathbf{u}^*\}$ yields a model matrix with full row rank.

We can use the leave-one-out cross validation errors to measure the accuracy of SIP and to determine when to stop the sequential procedure. Let $\mathbf{Y}_{-i} = (Y_1, \dots, Y_{i-1}, Y_{i+1}, \dots, Y_n)^T$. Then, the predicted value of the output at $\mathbf{u}_i$ using only the data $\mathbf{Y}_{-i}$ is $E(Y_i|\mathbf{Y}_{-i}, \boldsymbol{\theta})$, and the prediction error is $e_i = Y_i - E(Y_i|\mathbf{Y}_{-i}, \boldsymbol{\theta})$. Thus, the vector of leave-one-out prediction errors is $\mathbf{e} = (e_1, \dots, e_n)^T$, and we can measure prediction accuracy with

$$R_{pred}^2 = 100[1 - \|\mathbf{e}\|_2^2 / \sum_{i=1}^n (Y_i - \bar{Y})^2], \quad (6)$$

where $\bar{Y} = \sum_{i=1}^n Y_i / n$. We can stop the sequential procedure when R_{pred}^2 is sufficiently large. A short-cut formula for computing $\mathbf{e}$ is given in Proposition 1.

Proposition 1: The vector of leave-one-out prediction errors $\mathbf{e}$ is given by $\mathbf{e} = \mathbf{Y} -$

$$\begin{aligned} & \left(E(Y_1|\mathbf{Y}_{-1}, \boldsymbol{\theta}), \dots, E(Y_n|\mathbf{Y}_{-n}, \boldsymbol{\theta}) \right)^T = \mathbf{R}[(\mathbf{X}\mathbf{S}\mathbf{X}^T)^{-1}](\mathbf{Y} - \mu\mathbf{1}_n), \text{ where } \mathbf{R} = \text{diag}\{1/ \\ & [(\mathbf{X}\mathbf{S}\mathbf{X}^T)^{-1}]_{11}, \dots, 1/[(\mathbf{X}\mathbf{S}\mathbf{X}^T)^{-1}]_{nn}\}, \text{ and } [(\mathbf{X}\mathbf{S}\mathbf{X}^T)^{-1}]_{ii} \text{ is the } i\text{th diagonal element of } (\mathbf{X}\mathbf{S}\mathbf{X}^T)^{-1}. \end{aligned}$$

Proof:

Let $\mathbf{x}_n$ be the n th row of $\mathbf{X}$, where $\mathbf{X}$ is the model matrix, and let $\mathbf{X}_{-n}$ be the matrix obtained from $\mathbf{X}$ by deleting the n th row. Then,

$$(\mathbf{X}\mathbf{S}\mathbf{X}^T)^{-1} = \left[\begin{pmatrix} \mathbf{X}_{-n} \\ \mathbf{x}_n \end{pmatrix} \mathbf{S} \begin{pmatrix} \mathbf{X}_{-n}^T & \mathbf{x}_n^T \end{pmatrix} \right]^{-1} = \begin{bmatrix} \mathbf{X}_{-n}\mathbf{S}\mathbf{X}_{-n}^T & \mathbf{X}_{-n}\mathbf{S}\mathbf{x}_n^T \\ \mathbf{x}_n\mathbf{S}\mathbf{X}_{-n}^T & \mathbf{x}_n\mathbf{S}\mathbf{x}_n^T \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{C}^T & A \end{bmatrix},$$

where $\mathbf{B} = \mathbf{D} + \mathbf{D}(\mathbf{X}_{-n}\mathbf{S}\mathbf{x}_n^T)A(\mathbf{x}_n\mathbf{S}\mathbf{X}_{-n}^T)\mathbf{D}$, $A = [\mathbf{x}_n\mathbf{S}\mathbf{x}_n^T - (\mathbf{x}_n\mathbf{S}\mathbf{X}_{-n}^T)\mathbf{D}(\mathbf{X}_{-n}\mathbf{S}\mathbf{x}_n^T)]^{-1}$, $\mathbf{C} = -\mathbf{D}(\mathbf{X}_{-n}\mathbf{S}\mathbf{x}_n^T)A$, and $\mathbf{D} = (\mathbf{X}_{-n}\mathbf{S}\mathbf{X}_{-n}^T)^{-1}$.

This gives

$$(\mathbf{X}\mathbf{S}\mathbf{X}^T)^{-1}(\mathbf{Y} - \mu\mathbf{1}_n) = \begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{C}^T & A \end{bmatrix} \begin{pmatrix} \mathbf{Y}_{-n} - \mu\mathbf{1}_{n-1} \\ Y_n - \mu \end{pmatrix} = \begin{pmatrix} \mathbf{B}(\mathbf{Y}_{-n} - \mu\mathbf{1}_{n-1}) + \mathbf{C}(Y_n - \mu) \\ \mathbf{C}^T(\mathbf{Y}_{-n} - \mu\mathbf{1}_{n-1}) + A(Y_n - \mu) \end{pmatrix}.$$

The n th element of $(\mathbf{X}\mathbf{S}\mathbf{X}^T)^{-1}(\mathbf{Y} - \mu\mathbf{1}_n)$ is

$$\begin{aligned} [(\mathbf{X}\mathbf{S}\mathbf{X}^T)^{-1}(\mathbf{Y} - \mu\mathbf{1}_n)]_n &= \mathbf{C}^T(\mathbf{Y}_{-n} - \mu\mathbf{1}_{n-1}) + A(Y_n - \mu) \\ &= -A\mathbf{x}_n\mathbf{S}\mathbf{X}_{-n}^T\mathbf{D}(\mathbf{Y}_{-n} - \mu\mathbf{1}_{n-1}) + A(Y_n - \mu) \\ &= A\{-\mathbf{x}_n\mathbf{S}\mathbf{X}_{-n}^T(\mathbf{X}_{-n}\mathbf{S}\mathbf{X}_{-n}^T)^{-1}(\mathbf{Y}_{-n} - \mu\mathbf{1}_{n-1}) + (Y_n - \mu)\}. \end{aligned}$$

By (6),

$$E(Y_n | \mathbf{Y}_{-n}, \boldsymbol{\theta}) = \mu + (\mathbf{x}_n\mathbf{S}\mathbf{X}_{-n}^T)(\mathbf{X}_{-n}\mathbf{S}\mathbf{X}_{-n}^T)^{-1}(\mathbf{Y}_{-n} - \mu\mathbf{1}_{n-1}).$$

Thus, we obtain

$$\begin{aligned} Y_n - E(Y_n | \mathbf{Y}_{-n}, \boldsymbol{\theta}) &= Y_n - \mu - (\mathbf{x}_n\mathbf{S}\mathbf{X}_{-n}^T)(\mathbf{X}_{-n}\mathbf{S}\mathbf{X}_{-n}^T)^{-1}(\mathbf{Y}_{-n} - \mu\mathbf{1}_{n-1}) \\ &= [(\mathbf{X}\mathbf{S}\mathbf{X}^T)^{-1}(\mathbf{Y} - \mu\mathbf{1}_n)]_n / A. \end{aligned}$$

By symmetry, we conclude that $\mathbf{e} = \mathbf{R}[(\mathbf{X}\mathbf{S}\mathbf{X}^T)^{-1}](\mathbf{Y} - \mu\mathbf{1}_n)$.

S2.1 Simulation Study 2

In this study, we investigate the effectiveness of the proposed sequential procedure. For the Rosenbrock and Tilden test functions, initial cosine maximin LHD's of size $7d$ are used.

Design points are added until the size of the design is $10d$. We search for the maximum variance

point $\mathbf{u}^*$ by optimizing the prediction variance function $\text{var}(Y(\mathbf{u})|\mathbf{Y}, \boldsymbol{\theta})$ using the Nelder -Mead simplex algorithm (fminsearch function in Matlab) with $2d$ starting points. The starting points are the $2d$ points in the current test set with largest prediction variance. The test set at each iteration of the sequential design consists of points from a 1000-run maximin LHD (same for all iterations) and 2^d corner points less those points that are identical to any design point. Note that the search procedure uses multiple starting points to attempt to find the global maximum point $\mathbf{u}^*$ of the function $\text{var}(Y(\mathbf{u})|\mathbf{Y}, \boldsymbol{\theta})$ due to the multimodality of the function. For this study, we focus only on the performance of the sequential procedure for Legendre-SIP's. A total of 100 simulations for each test function are run. We observe in the simulations that the sequential procedure tends to add many boundary and corner points to the design because these points often have large prediction variances.

Table S1: Simulation Means of Initial and Final Values of R_{pred}^2 , PRMSE, Percent Coverage, and Maximum and Average Prediction Standard Deviation over Range of Function in Test Set

	Rosenbrock	Tilden
Initial n	21	28
Final n	30	40
Initial R_{pred}^2	55.9	97.5
Final R_{pred}^2	96.6	97.9
Initial PRMSE	49.5	13.1
Final PRMSE	12.2	8.6
Initial Percent Coverage	93.7	89.5
Final Percent Coverage	99.3	90.3
Initial Max StD / Range	0.3611	0.1194
Final Max StD / Range	0.0649	0.0429
Initial Ave StD / Range	0.0594	0.0209
Final Ave StD / Range	0.0242	0.0152

Table S1 shows the simulation means of the initial and final values of R_{pred}^2 , PRMSE, percent coverage, and maximum and average prediction standard deviation expressed as a

fraction of the range of true function values in the test set. A comparison of Table 2, Table 3, and Table S1 indicates that the sequential procedure does not yield improvements in PRMSE and percent coverage over cosine maximin LHD's of size $10d$. A comparison of Table 4 and Table S1 also indicates that the sequential procedure does not provide improvements in the average standard deviation over cosine maximin LHD's of size $10d$. However, Table 4 and Table S1 show that the sequential procedure yields substantially smaller maximum standard deviation on the average. In particular, the simulation mean of the maximum standard deviation over range of function is reduced from 0.141 to 0.065 for the Rosenbrock function and from 0.090 to 0.043 for the Tilden function. A smaller maximum standard deviation can be an important advantage.

Table S1 suggests that improvements in R^2_{pred} is a good indicator of improvements in PRMSE. For the Rosenbrock function, there is a large difference between the mean initial and final values of R^2_{pred} and also a large difference between the mean initial and final PRMSE. For the Tilden function, the changes in R^2_{pred} and PRMSE are both small. The observation that improvements in R^2_{pred} is a good indicator of improvements in PRMSE is important because R^2_{pred} can be computed from the experiment data whereas PRMSE is not known in practice.

References

1. Gautschi, W. (1983). "The condition of Vandermonde-like matrices involving orthogonal polynomials," *Linear algebra and its applications*, 52, 293-300.
2. Gramacy, R.B. and Lee, H.K.H. (2009). "Adaptive design and analysis of supercomputer experiments," *Technometrics*, 51(2), 130-145.

Appendix A: Test Functions for Simulation Study 1

Name	Function
Rosenbrock $d = 3$	$y(\mathbf{u}) = \sum_{i=1}^2 [(u_i - 1)^2 + 100(u_{i+1} - u_i^2)^2]$
Tilden Function $d = 4$	$y(\mathbf{u}) = [2v_1v_4\exp(-v_2/300) + 1/v_3]^{-1},$ $v_1 = (1.3465 \times 10^7)(u_1 + 1) + 8.97 \times 10^6,$ $v_2 = (500)(u_2 + 1), v_3 = (0.1)(u_3 + 1) + 1,$ $v_4 = 4.75 \times 10^{-7}(u_4 + 1) + 5 \times 10^{-8}.$ <p>$v_1 = \text{chemical reaction rate constant}, v_2 = \text{temperature},$ $v_3 = \text{initial state of the chemical reaction system}, v_4 = \text{time}$</p>
Weld Shear Stress Function $d = 5$	$y(\mathbf{u}) = 0.5 \ln[t_1^2 + 2t_1t_2v_2/(2R) + t_2^2],$ $R = \sqrt{\frac{v_2^2}{4} + \left(\frac{v_1 + v_3}{2}\right)^2}, J = \sqrt{2}v_1v_2 \left[\frac{v_2^2}{12} + \left(\frac{v_1 + v_3}{2}\right)^2 \right],$ $M = v_4(v_2/2 + v_5), t_1 = v_4/(\sqrt{2}v_1v_2), t_2 = MR/J,$ $v_1 = 0.05(u_1 + 1) + 0.15, v_2 = 4(u_2 + 1) + 2, v_3 = 4(u_3 + 1) + 2, v_4 = 1000(u_4 + 1) + 4000, v_5 = (u_5 + 1) + 4$ <p>$v_1 = \text{weld height}, v_2 = \text{weld length}, v_3 = \text{bar thickness},$ $v_4 = \text{force acting on the beam},$ $v_5 = \text{distance from load to support}.$</p>
Two-Bar Truss Safety Constraint Function $d = 6$	$y(\mathbf{u}) = s/\sigma_c - 1$ $s = \frac{v_3\sqrt{v_2^2 + v_4^2}}{\pi v_1v_2v_5}, \sigma_c = \frac{\pi^2v_6(v_1^2 + v_5^2)}{8(v_2^2 + v_4^2)}$ $v_1 = 1.5(u_1 + 1) + 1, v_2 = 22.5(u_2 + 1) + 5,$ $v_3 = 3300(u_3 + 1) + 29700, v_4 = 3(u_4 + 1) + 27,$ $v_5 = 0.01(u_5 + 1) + 0.09, v_6 = 3 \times 10^6(u_6 + 1) + 27 \times 10^6.$ <p>$v_1 = \text{mean diameter of tubular members}, v_2 = \text{height of the truss},$ $2v_3 = \text{load on the truss}, 2v_4 = \text{span of the truss},$ $v_5 = \text{thickness of the truss members}, v_6 = \text{Young's modulus}.$</p>
Product Peak Function $d = 7$	$y(\mathbf{u}) = \prod_{i=1}^7 (1/c_i^2 + v_i^2)$ $v_i = (u_i + 1)/2, i = 1, \dots, 7.$ $c_i = 29(8 - i)/112, i = 1, \dots, 7.$

Borehole Flow Rate Function $d = 8$	$y(\mathbf{u}) = \frac{2\pi v_3(v_4 - v_6)}{\ln(v_2/v_1) \left[1 + \frac{2v_7v_3}{\ln(v_2/v_1) v_1^2 v_8} + \frac{v_3}{v_5} \right]},$ $v_1 = 0.05(u_1 + 1) + 0.05, v_2 = 24950(u_2 + 1) + 100,$ $v_3 = 26265(u_3 + 1) + 63070, v_4 = 60(u_4 + 1) + 990,$ $v_5 = 26.45(u_5 + 1) + 63.1, v_6 = 60(u_6 + 1) + 700,$ $v_7 = 280(u_7 + 1) + 1120, v_8 = 1095(u_8 + 1) + 9855.$ $v_1 = \text{radius of borehole}, v_2 = \text{radius of influence},$ $v_3 = \text{transmissivity of upper aquifer},$ $v_4 = \text{potentiometric head of upper aquifer},$ $v_5 = \text{transmissivity of lower aquifer},$ $v_6 = \text{potentiometric head of lower aquifer},$ $v_7 = \text{length of borehole}, v_8 = \text{hydraulic conductivity of borehole}.$
--	--

Appendix B: Details of Optimization Procedure for Finding the MAPE of $\mathbf{r}$ and the MLE of the Correlation Parameters for the GP Model

To find the MAPE of $\mathbf{r}$, i.e., the value of $\mathbf{r}$ that minimizes the function $\Lambda(\mathbf{r}) = -2 \ln[p(\mathbf{Y} | (\hat{\mu}, \hat{\sigma}^2, \mathbf{r}^T)^T, P)p(\mathbf{r})]$, the Nelder-Mead algorithm for unconstrained optimization implemented in the Matlab `fminsearch` function is employed. Because $\mathbf{r}$ is restricted to $(0,1)^d$, we make the transformation $r_i = \Phi(h_i)$, where Φ is the standard normal cumulative distribution function. The Nelder-Mead algorithm finds the optimal value of $\mathbf{h} = (h_1, \dots, h_d)^T$, which is unrestricted. Because the function $\Lambda(\mathbf{r})$ can have more than one local minimum, we optimize with two starting points, $h_i = \Phi^{-1}(0.25), i = 1, \dots, d$ and $h_i = \Phi^{-1}(0.75), i = 1, \dots, d$, and choose the better of the two optimal solutions. For $d \leq 5$, the parameters `MaxIter`, `TolX`, and `TolFun` for `fminsearch` are set to 10^6 , 10^{-2} , and 0.1; for $d \geq 6$, the parameters `MaxIter`, `TolX`, and `TolFun` for `fminsearch` are set to 10^7 , 0.05, and 0.3.

For the GP model, the Gaussian correlation function is $\mathcal{C}(\mathbf{z}) = \prod_{i=1}^d R_i^{z_i^2}$, where $\mathbf{R} = (R_1, \dots, R_d)^T \in (0,1)^d$. The method we use for finding the MLE of $\mathbf{R}$ is the same as the method described above for finding the MAPE of $\mathbf{r}$, except that the objective function is now the log likelihood. The Matérn correlation function is $\mathcal{C}(\mathbf{z}) = \prod_{i=1}^d (\rho_i^2 + 3\rho_i + 3)\exp(-\rho_i)/3$, where

$\rho_i = 2\sqrt{2.5}|z_i|/R_i$. The MLE of $h_i = \ln(R_i)$ is obtained by optimization with the Nelder-Mead algorithm. Two starting points are used $h_i = \ln(0.89), i = 1, \dots, d$ and $h_i = \ln(2.24), i = 1, \dots, d$. Note that for $d = 1$, $C(1) = 0.25$ and $C(1) = 0.75$ when $R_1 = 0.89$ and $R_1 = 2.24$ respectively, which give identical starting correlations as for the Gaussian correlation function.