

Published in IET Science, Measurement and Technology
 Received on 30th June 2014
 Revised on 29th August 2014
 Accepted on 29th September 2014
 doi: 10.1049/iet-smt.2014.0194



Correlation matrices in kriging assisted optimisation of electromagnetic devices

Song Xiao, Mihai Rotaru, Jan K. Sykulski

Electronics and Computer Science (ECS), University of Southampton, Southampton, SO17 1BJ, UK
 E-mail: jks@soton.ac.uk

Abstract: Kriging surrogate modelling facilitates efficient decision making regarding where to place the next point for evaluation during optimisation. This is particularly helpful in the design of electromagnetic devices where computationally expensive numerical field modelling needs to be used. The disadvantage, however, is that correlation matrices are required which, for problems with many design variables and multiple objectives, may grow in size leading to the need for page swapping and thus slowing down of what in principle should be a very fast process. In this study several methodologies to reduce the computational resources required in such problems are proposed. The efficiency of the proposed approach is demonstrated using an example of a large multi-parameter optimisation problem where kriging coupled with the average gradient value method is employed.

1 Introduction

Kriging [1–4] predicts the shape of the objective function by considering the spatial correlation of data based on limited information and thus offers an efficient and inexpensive surrogate to replace the computationally demanding numerical simulation (such as finite elements). The accuracy of the prediction can be estimated by the mean square error in kriging to assist in a decision on where to place the next evaluation point during optimisation iterations. The spatial correlation exists between the known points (vectors) of the objective function and all the unknown points, as well as among the known points (newly found points and initial sampling points). The way of calculating and storing the kriging correlation matrices was suggested in [5] and is shown schematically in the box marked M1 in Fig. 1.

This relies on the linear regression model

$$\hat{y}(x) = \sum_{k=1}^m \beta_k f_k(x) + \varepsilon(x) \quad (1)$$

and the Gaussian correlation model

$$R(\varepsilon(x^i), \varepsilon(x^j)) = \prod_{k=1}^n e^{-\theta_k |x_k^i - x_k^j|^{p_k}} \quad (2)$$

where the global function $\sum_{k=1}^m \beta_k f_k(x)$ and an additive Gaussian noise $\varepsilon(x)$ are integrated to the predicted value $\hat{y}(x)$ of the objective function; θ_k is the correlation among the data in the k -direction and p_k determines the ‘smoothness’ of (2). The most popular correlation function is given by the Gauss model where the value of p_k is

simply taken as equal to 2. For a given set of data, the maximum likelihood estimation optimises the value of θ and then the correlation model is brought into the regression model to evaluate the function with the best linear unbiased predictor [1]. Although theoretically kriging could be used for any type and size of optimisation, care must be taken with large problems (many variables and multi-objective) as the correlation matrices can grow rapidly.

2 Storage issue of correlation matrices

With the increase in the number of sampling points selected by kriging during the iterations, the amount of data produced by the correlation matrices (especially the correlation matrices between the known sampling points and the unknown points of the objective function), shown as M1 in Fig. 1, accumulates constantly throughout the optimisation process, which may become problematic especially when coping with large-scale multi-parameter tasks, resulting in a ‘combinatorial explosion’ [5]. In our previous work [6] a successive ‘zoom in’ strategy to alleviate the problem was proposed, where – in order to reduce the amount of data storage and utilise the installed physical memory capacity efficiently – the step sizes of the design vectors were increased while the test range reduced. However, the optimal step size is often problem dependent, thus if the ‘roughness’ of the initial test is set inappropriately, it is possible that certain regions of the search space containing important information (including the optimum) might be missed. How to utilise the test experience obtained from the initial-stage tests, to guide the model to define the test range and specific step size flexibly in each dimension, is discussed in this paper, with the aim to guarantee that the surrogate model converges to the

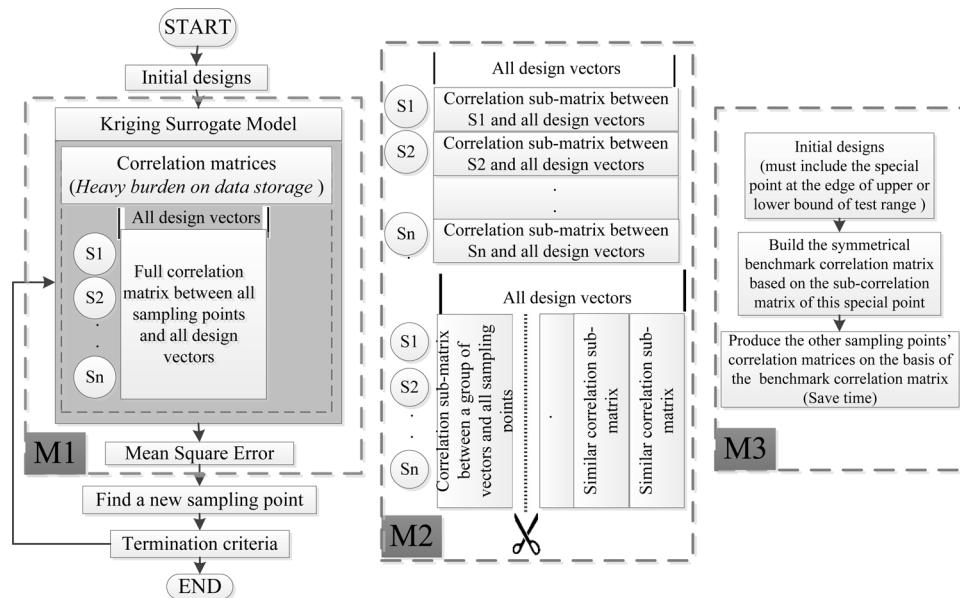


Fig. 1 Schemes of producing correlation matrices

global optimum or, at worst, the region nearby the global optimum.

There are two ways in which the correlation matrices can be divided into sub-elements that can be manipulated in an efficient way: by partitioning the correlation matrix in terms of sampling points or via the design vectors, as shown in box M2 of Fig. 1. The splitting of the correlation matrix by sampling points, which is only adding sub-matrices between each sampling point and all design vectors, rather than inserting the full-version correlation matrix into the calculation, is applied during the process of producing the predicted response surface. However, during the process of estimating the mean square error in the predictor, a unique way of partitioning the correlation matrix is by dividing it into sub-matrices via the design vector. This partitioning approach can be time-consuming, however, because each cut requires rebuilding the full correlation matrix for 'grabbing' a slice of the sub-matrix. Using only one of

these approaches may not give the expected computational improvement, thus both schemes in M2 have been used for different purposes, one for producing a response surface, the other for producing the mean square error. Together they can maintain correlation matrices at acceptable sizes that should not exceed the available memory space and thus manage the available RAM more efficiently.

Before explaining the details of the proposed approach it may be useful to make some more general remarks. First, the identified problems associated with large amount of data are not unique to algorithms based on kriging approach and may occur when other surrogate modelling techniques are used; thus some of the techniques developed here will be of more general interest. Secondly, no optimisation technique can guarantee finding a global optimum but kriging – as demonstrated by other authors and our previous publications – appears to be particularly effective in maximising the chances of determining one, while at the

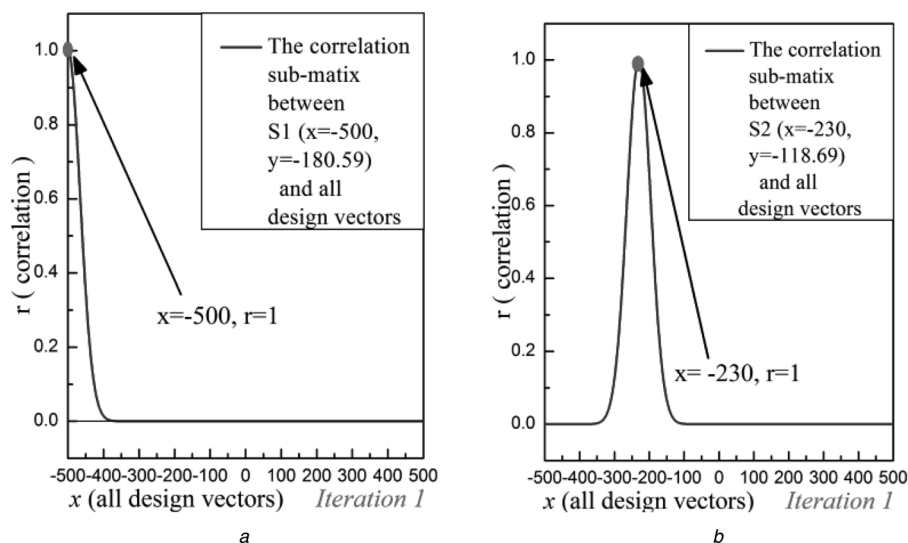


Fig. 2 Correlation sub-matrices for

a Point 1
b Point 2

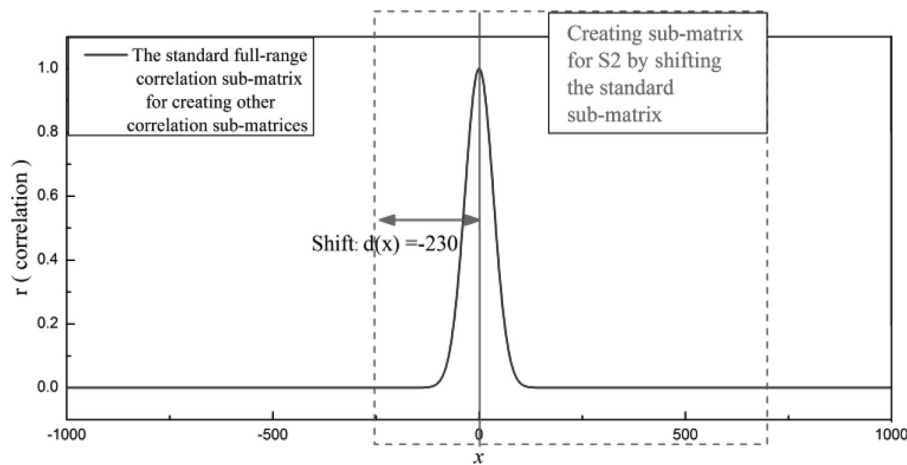


Fig. 3 'Standard' sub-matrix and associated 'window'

same time offering savings in associated computation times. Moreover, in accordance with the no free lunch theorem, its performance is problem dependent but has been shown before to be particularly efficient when solving problems relying on time consuming numerical modelling, such as finite element electromagnetic field simulation. Finally, it is worth emphasising that the computer memory issues resulting from the combinatorial explosion mentioned above are not only a feature of the kriging methodology but are also associated with a particular implementation of the method.

3 Structure of the correlation matrices

It has been observed that the sub-matrices created when new sampling points are added during iterations have similarities that can be exploited to reduce the memory requirements. In particular, the 'shape' of the distribution of values in a given row of the matrix will be preserved (a 'benchmark') but shifted depending on the position of the point. Thus instead of creating and storing the whole matrix it may be sufficient to duplicate information. This is best explained using examples, therefore in the following sections several cases are discussed and analysed, starting with a single variable problem, gradually moving to multi-variable problems of up to eight variables. The single and two variable problems are exemplified using analytical functions, whereas the multi variable problems are illustrated with the help of TEAM 22 (SMES Optimisation Benchmark) and TEAM 25 (Optimisation of Die Press Model) electromagnetic optimisation problems.

3.1 Single-variable numerical test

The single-variable Schwefel function [7] was first used

$$f(x) = \sum_{i=1}^d -x_i \sin(\sqrt{|x_i|}) \quad (3)$$

where $d=2$, with six initial sampling points $x = (-500, -230,$

$-250, -40, 160, 500)$, paying attention that the first one was on the edge of the region. A full correlation sub-matrix was then created for this point and the distribution is shown in Fig. 2a. It was then observed that all points have a similar distribution, with Fig. 2b giving an example for the second point. It is therefore possible to set up a 'standard' sub-matrix (as depicted in Fig. 3) and then shift it (using a 'window') to an appropriate position depending on the location of the point under consideration (effectively a 'cut and paste' technique), hence use storage space far more economically.

3.2 Two-variable numerical test

The following analytical function was used and tested for the two variable case (see (4)). Although this function can have n variables, for the example discussed here n was set to two ($n=2$) in the range $0 \leq x_i \leq 27$. The point $(x_1=0, x_2=0, y=9.43)$ was selected to create a standard correlation sub-matrix to be used by all remaining and new points, as illustrated graphically in Fig. 4.

To assess achievable savings in computer memory and associated computing times, the graphs in Fig. 5 have been produced, showing and comparing three cases: the full correlation matrix, the reduced size using the idea of a standard sub-matrix and the joint usage of the sub-matrix and matrix partitioning described previously in [5]. Note that for this particular comparison (as well as later on in Fig. 6) the three approaches are equivalent to M1, M2 and a combination of M2 and M3 explained previously in Fig. 1.

The test demonstrates that significant savings can be made, if required, of necessary memory requirements, by applying the concept of duplicating the standard correlation sub-matrix. This can be combined with the idea of matrix partitioning. The added bonus is that the computing times can also be reduced, especially as the iterations progress, although in the case of matrix partitioning this benefit may be lost somewhat towards the end of the iterative process. When applying the correlation matrix partitioning scheme, the splitting by the design vectors requires more time to produce the correlation matrix repeatedly for creating a sub-matrix; thus the sub-matrix duplication assisted by

$$y = 10 - \sum_{i=1}^n \left[\frac{3.5}{1 + (x_i - 5)^2} + \frac{2.2}{1 + (x_i - 15)^2/10} + \frac{1.2}{1 + (x_i - 25)^2/30} \right] \quad (4)$$

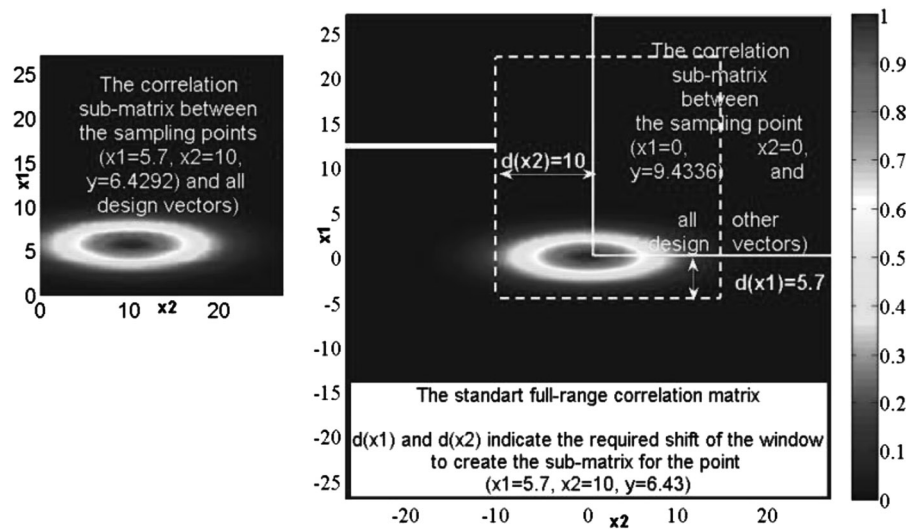


Fig. 4 Building and duplicating sub-matrices

correlation matrix partitioning may ultimately require more computing time.

4 TEAM 22

4.1 TEAM problem 22

The full description of the TEAM benchmark problem 22 (superconducting magnetic energy storage system) may be found in [8]. The target is an arrangement of the two superconducting coils such that the stored energy within the system is $E_{\text{ref}} = 180$ MJ while a minimal stray field B_{stray} is

present. The objective function is defined as

$$\text{OF} = \frac{B_{\text{stray}}^2}{B_{\text{norm}}^2} + \frac{|E - E_{\text{ref}}|}{E_{\text{ref}}} \quad (5)$$

where $B_{\text{norm}} = 3 \mu\text{T}$ and $B_{\text{stray}}^2 = \frac{\sum_{i=1}^{22} |B_{\text{stray},i}|^2}{22}$, subject to geometrical and ‘quench’ constraints.

4.2 3-parameter test results

The full three parameter TEAM 22 problem with standard test settings [8] is potentially a challenge to the kriging method because of the ‘combinatorial explosion’ associated with setting up the correlation model, as explained in Section 3 – thus the savings because of avoidance of the computationally expensive finite element simulations could be compromised by the increased time required by the kriging model should the problem of memory restriction be encountered. The initial sampling points were chosen differently to our previously reported test [9]; in particular the value at the lower bound of the three variable’s test range was selected specifically as the first sampling point to create a standard correlation sub-matrix for duplication, instead of repeatedly calling the correlation function (2), which would be an inefficient and time consuming process. The rest of the sampling points are randomly chosen as summarised in Tables 1 and 2. The efficiency of the algorithm and the memory usage were compared for the three methods M1, M2 and M3 of Fig. 1, where M1 and M2 have already been explained, whereas M3, as a combination of the other two methods, is similar to M2, but with initial duplication of the standard sub-matrix before partitioning is applied. With the prescribed ranges and step sizes the total size of the problem is $81 \times 129 \times 81$, consistent with the standard test size suggested in [8], whereas the size of the sub-matrices needs to be set to be less than the physical memory available in the computer but also depends on the number of sampling points; in our tests this has been set to $n_s \times 5 \times 10^5$, where n_s is the number of existing sampling points.

Only the calculation times during the prediction stage and the memory usage are different when using these three

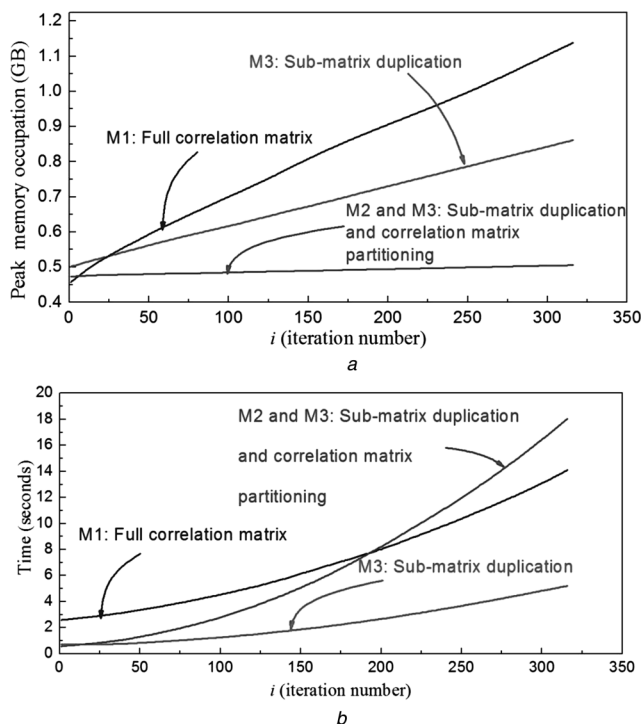


Fig. 5 Memory requirements (a), and computing times (b), for the two-variable test described by (4)

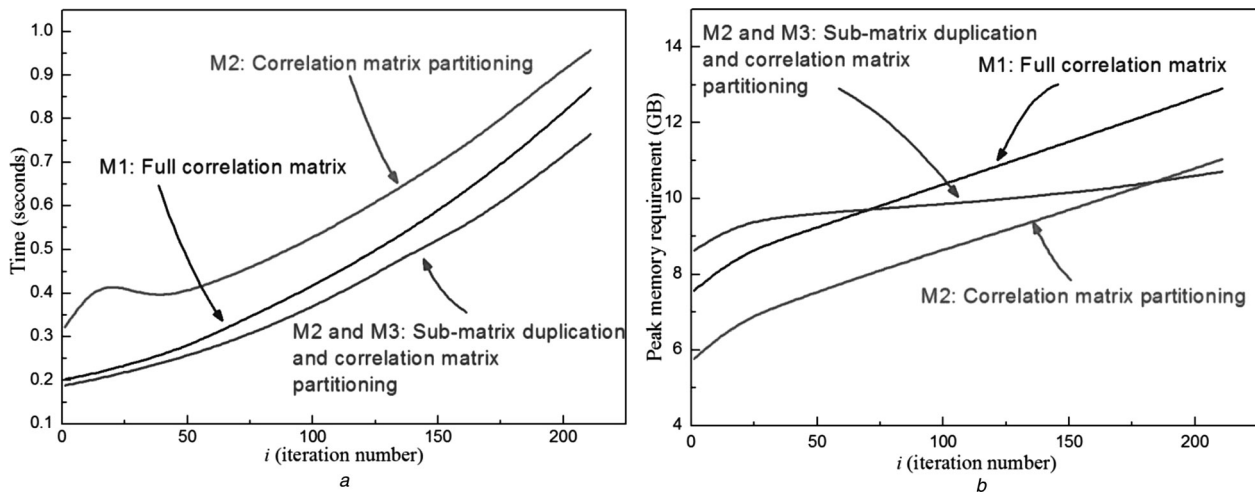


Fig. 6 Computing times (a), and peak memory requirements (b), obtained by different correlation matrix building methods for the TEAM 22 problem

different approaches of correlation matrices construction, since the number of updated sampling points required for converging to the global optimum is the same. A comparison between the results obtained with our kriging assisted algorithm, using the three schemes of building correlation matrices and some published results using other methods is presented in Table 3. The slight difference in the results obtained here as compared with our previous work [9] is because of the different initial settings used.

Table 1 Setting of initial sampling points

	R_2 (m)	h_2 (m)	d_2 (m)
sample 1	2.6	0.408	0.1
sample 2	3.4	2.2	0.4
sample 3	3.2	0.744	0.4

Table 2 Specific definition of the test

	R_2 (m)	h_2 (m)	d_2 (m)
test range	[2.6 3.4]	[0.408 2.2]	[0.1 0.4]
step size	0.01	0.014	0.003
number of steps	81	129	101

Table 3 Performance comparison algorithms for TEAM 22 problem

Algorithm	R_2 (m)	d_2 (m)	$h_2/2$ (m)	OF	No. of FEM calls
GA	3.040	0.386	0.240	0.134	2400
HuTS	3.080	0.380	0.246	0.089	3821
ITS	3.100	0.388	0.240	0.098	1824
SA	3.078	0.390	0.237	0.098	5025
NTS	3.080	0.370	0.254	0.089	1800
PBIL	3.110	0.421	0.241	0.101	3278
Kriging	3.09	0.349	0.267	0.08778	217

Genetic algorithm (GA) [10]; Tabu search (HuTS) [11]; improved Tabu search (ITS) [12]; simulated annealing algorithm (SA) [13]; New Tabu Search (NTS) [14]; Population-based Incremental Learning (PBIL) [15]. Results for GA, HuTS, ITS, SA, NTS and PBIL taken from [14].

The kriging assisted by EI took 214 iterations to locate the optimum, including the initial sampling points, thus the total number of calls of the FEM software was 217, which outperforms other algorithms.

But the main purpose of this test was to observe calculation times and peak memory requirements when different schemes were used for building correlation matrices. From Fig. 6 it can be seen that although the memory usage can be maintained below a certain size through applying the correlation matrix partitioning method, computing times are longer during the prediction process because of the way of partitioning the matrices via design vectors. On the other hand, when the correlation matrix partitioning method is combined with the sub-matrix duplication scheme, the kriging model performs the optimisation task efficiently and, as shown in Fig. 6a, outperforms the other two algorithms. Initially the sub-matrix duplication method, which requires storing the benchmark standard sub-matrix for duplicating the other correlation matrices, has a larger memory usage, but after approximately 25 iterations, for this specific TEAM 22 problem, the rate of memory increase reduces quite dramatically, as shown in Fig. 6b. As the iterations continue there is a linear increase of memory requirements for algorithms M1 and M2, which eventually overtake the memory requirements for the M3 approach. Although for this case the size of the problem was not exceptionally large, and was comfortably handled by a PC with a 16 GB of installed RAM, it is obvious that for larger problems using M3 would be advantageous.

4.3 8-parameter TEAM 22

The 8-parameter TEAM 22 problem is a challenging test and the fact that there are very few results published in literature is a clear indication of the difficulties. Several observations can be made before attempting to solve this problem using the proposed algorithm. It is obvious that a large memory will be necessary to store the response surface produced by kriging, as well as the correlation matrices used during the prediction stage. The size of these matrices is directly linked to the number of variables, hence in this case of eight variables and using modest 10 steps for each variable vector, a 10^8 parameter set needs to be stored. Since the kriging predictor produces the response surface at each

Table 4 The specific definition of 8-parameter TEAM 22 test at the initial stage

	R_1 (m)	R_2 (m)	$h_1/2$ (m)	$h_2/2$ (m)	d_1 (m)	d_2 (m)	J_1 (A/mm ²)	J_2 (A/mm ²)
min	1.0	1.8	0.1	0.1	0.1	0.1	10	-30
max	4.0	5.0	1.8	1.8	0.8	0.8	30	-10
no. of steps	6	6	6	6	6	6	3	3
step size	0.6	0.64	0.34	0.34	0.14	0.14	10	10

Table 5 The average value of gradient of 8-parameter TEAM 22 test

		R_1 (m)	R_2 (m)	h_1 (m)	h_2 (m)	d_1 (m)	d_2 (m)	J_1 (A/mm ²)	J_2 (A/mm ²)
first-stage test	$D(x)$	0.6	0.64	0.14	0.14	0.68	0.68	10	10
	AG	0.0046	6.433×10^{-4}	0.0184	0.0446	0.2168	0.2168	3.036×10^{-3}	3.499×10^{-5}
second-stage test	$D(x)$	0.3	0.32	0.17	0.17	0.04	0.04	10	10
	AG	0.0148	0.0541	0.0947	0.1024	0.8546	0.8415	4.232×10^{-3}	4.279×10^{-3}
third-stage test	$D(x)$	0.3	0.32	0.085	0.085	0.01	0.01	10	10
	AG	0.0375	0.0231	2.736	1.743	0.476	0.257	4.572×10^{-3}	2.1755×10^{-3}

D : parameter value, AG: average gradient.

Table 6 Test ranges and steps for the 8-parameter TEAM 22 test

		R_1 (m)	R_2 (m)	h_1 (m)	h_2 (m)	d_1 (m)	d_2 (m)	J_1 (A/mm ²)	J_2 (A/mm ²)
second-stage test	test range	[1.0 2.8]	[1.8 3.72]	[0.2 2.24]	[0.2 2.24]	[0.38 0.66]	[0.1 0.38]	[10 30]	[-30 -10]
	no. of steps	7	7	7	7	8	8	3	3
third-stage test	test range	[1.0 1.6]	[1.8 3.72]	[0.2 3.6]	[0.2 3.6]	[0.1 0.8]	[0.1 0.8]	[10 30]	[-30 -10]
	no. of steps	3	3	21	21	29	29	3	3
fourth-stage test	test range	1.0	1.8	[0.2 3.6]	[0.2 3.6]	[0.4 0.6]	[0.1 0.3]	[10 30]	[-30 -10]
	no. of steps	1	1	41	41	21	21	3	3

Table 7 Optimal solution found in the four tests and the corresponding iterations

	The first-stage test	The second-stage test	The third-stage test	The fourth-stage test
optimum solution	$R_1 = 1, R_2 = 1.8, h_1 = 0.2, h_2 = 0.2, d_1 = 0.52, d_2 = 0.24, J_1 = 30, J_2 = -20, OF = 0.9392$	$R_1 = 1, R_2 = 1.8, h_1 = 0.54, h_2 = 0.54, d_1 = 0.62, d_2 = 0.38, J_1 = 20, J_2 = -20, OF = 0.63732$	$R_1 = 1, R_2 = 1.8, h_1 = 0.37, h_2 = 1.9, d_1 = 0.625, d_2 = 0.1, J_1 = 30, J_2 = -10, OF = 0.7874$	$R_1 = 1, R_2 = 1.8, h_1 = 1.56, h_2 = 1.39, d_1 = 0.4, d_2 = 0.15, J_1 = 30, J_2 = -30, OF = 0.05361$
no. of FEM calls	183	17	29	220

The units of the 8 parameters are the same as in Table 6; OF: the objective function value.

iteration, and it is of the same size as the task itself (10^8), a computer with 9 GB available memory is bound to be challenged. Moreover, background processes are likely to reduce available memory for the optimiser because of dynamic allocation. Adding correlation matrices to this data will only make the problem worse. Thus the correlation matrix partitioning and/or sub-matrix duplication schemes may need to be supplemented by some form of a 'zoom in' strategy, which has been used successfully before. However, the difficulty of this strategy is how to define the optimal step size, as this may determine whether the algorithm will converge to the right answer. To alleviate this difficulty it is proposed that a sensitivity analysis in each dimension is conducted before deciding which of the parameters should be using a finer step size. We have decided to use an average value of the gradient in each dimension to guide the algorithm when choosing the variables that need a finer resolution. The average value of the gradient is thus calculated after each iteration and a variable that needs finer resolution may change in the next

step. Table 4 presents the specific settings at the initial stage (note also that $B_{\text{norm}} = 200 \mu\text{T}$ in this case).

After the first-stage test, in order to determine how to resize the parameters at the next stage, an average value of the gradient with respect to each dimension was calculated. The values obtained from the tests at different stages are listed in Table 5. In this particular case the parameters h_1, h_2, d_1, d_2 seem to have higher values of the average gradient, indicating higher sensitivity and should therefore be focused on in the following tests by refining the increment (increasing the number of steps) in stages two, three and so on. The range of each variable is also monitored and may be decreased or increased as appropriate. All this is illustrated in Tables 6 and 7.

Through analysing the sensitivity of the first-stage rough test, it has been determined that the test range of R_1 and R_2 needs to be contracted in order to save computer memory for the other four more sensitive parameters. Moreover, the sensitivity of the current density settings (J_1 and J_2) is extremely low, so during the three tests the test ranges and

Table 8 Performance comparison between different algorithms

	R_1 (m)	R_2 (m)	h_1 (m)	h_2 (m)	d_1 (m)	d_2 (m)	J_1 (A/mm ²)	J_2 (A/mm ²)	Objective function	No. of FEM calls
PSO	1.0000	2.2647	1.1076	1.7766	0.5225	0.3442	28.1779	-5.4921	2.3363	~6000
Q-PSO	2.2947	2.6126	1.0764	2.2704	0.3967	0.2040	30	-21.293	0.5735	~6000
E-QPSO	1.0000	1.8000	2.0616	3.6000	0.5155	0.2851	19.9975	-6.3571	1.1730	~6000
GSA	1.939	2.823	1.130	1.101	0.399	0.195	22.5	-22.5	0.00517	17 150
ES	1.990	2.931	1.293	0.940	0.290	0.188	26.6	-26.6	0.00489	4200
SAA	1.694	2.907	1.609	0.882	0.323	0.207	20.9	-20.9	0.0110	14000
CGM	1.836	2.762	1.178	1.001	0.395	0.214	22.5	-22.5	0.0248	200
Kriging	1	1.8	1.56	1.39	0.4	0.15	30	-30	0.05361	449
standard answer	1.296	1.8	2.178	3.026	0.583	0.195	16.955	-18.91	0.0018	–

PSO: particle swarm optimisation [16], Q-PSO: quantum-behaved particle swarm optimisation [16–18], E-QPSO: QPSO using the exponential probability distribution [19], GSA: global search algorithm [20], ES: evolution strategy [20], SAA: simulated annealing algorithm [20], CGM: conjugate gradient method [20].

Results for PSO, Q-PSO, E-PSO, GSA, ES, SAA and CGM taken from [16] and [20].

step sizes for currents have not been changed. The correlation matrix partitioning strategy has been applied and the full correlation matrix divided into sub-matrices of the size $n_s \times 3 \times 10^5$ where n_s is the number of sampling points.

The results presented here were acquired applying the following methodology: if available physical memory was larger than requested by the problem the normal method M1 was applied. A switch that allows moving to the correlation matrix partitioning method was implemented; the switching criterion was controlled by the time consumed to construct the correlation matrix. If the time used to build the correlation matrix was larger than a certain threshold, the switching was triggered and the partitioning scheme activated, which then continued throughout the remaining iterations.

In terms of the quality of the answer itself, compared with other methods and the standard answer provided by [8], with the help of the zoom-in strategy based on the evaluation of the predicted average gradient value, the kriging algorithm seems to perform much more efficiently than the particle swarm (PS) and related methods with only 449 iterations required to converge to the optimum, compared with estimated 6,000 needed with PS methods. The other three algorithms (GSA, ES and SA) can find a slightly better solution than the one achieved by kriging but with much higher number of FEM calls. The conjugate gradient method (CGM) performs efficiently and provides an accurate answer, but it is of course a direct method, thus generally cannot guarantee finding a global optimum. Obviously no method can provide such a guarantee, but stochastic methods in general and kriging based methodologies in particular, offer a high probability of success. Although the quality of the solution obtained by kriging was marginally worse than the standard answer, it was achieved using a very small number of FEM calls. Should the optimum in Table 8 obtained by kriging not satisfy the designer, more detailed tests could be added to continue the search, in particular by examining the sensitivity in other directions.

Table 9 Specific definition of TEAM 25

Parameter	Test range (mm)		Step size (mm)	No. of steps
	Min.	Max.		
R_1 (mm)	5	9.4	0.1	45
L_2 (mm)	12.6	18	0.1	55
L_3 (mm)	14	45	1	32
L_4 (mm)	4	19	0.5	31

5 TEAM 25

A model of a die press with an electromagnet for producing anisotropic permanent magnets is chosen as a second example [21]. The shape of the die is to be set up in such a way that magnetic flux density components B_x and B_y should be the same and equal to $0.35\cos(\theta)$ T along a circle line at 10 measurement points for $0^\circ < \theta < 45^\circ$ and $r_0 = 0.01175$ m. The problem has four design parameters R_1 , L_2 , L_3 and L_4 specified in Table 9. The objective function is evaluated at specific points as

$$\text{OF} = \sum_{i=1}^{10} \left((B_{xi,\text{calc}} - B_{xi,\text{requ}})^2 + (B_{yi,\text{calc}} - B_{yi,\text{requ}})^2 \right) \quad (6)$$

where calc means calculated and requ required.

Three initial sampling points have been chosen as $(R_1 = 5.0, L_2 = 12.6, L_3 = 14, L_4 = 4.0)$, $(R_1 = 7.0, L_2 = 15.8, L_3 = 20, L_4 = 15)$, $(R_1 = 8.5, L_2 = 17, L_3 = 31, L_4 = 8)$. The first sampling point has been chosen so as to allow for creating the standard matrix for duplicating correlation matrices.

As before, during the first stage of the test (as shown in Fig. 7) the normal approach M1 has been applied until the switching criteria (when the peak memory requirement exceeds 9 GB) is triggered and then Method M3 is applied from the 61st iteration onwards. Although the theoretical physical memory size is 16 GB, with the consideration of the impact from other background processes, the upper limit for the predictor was set to 9 GB. Interestingly the computing time at the instant of switching shows a slight

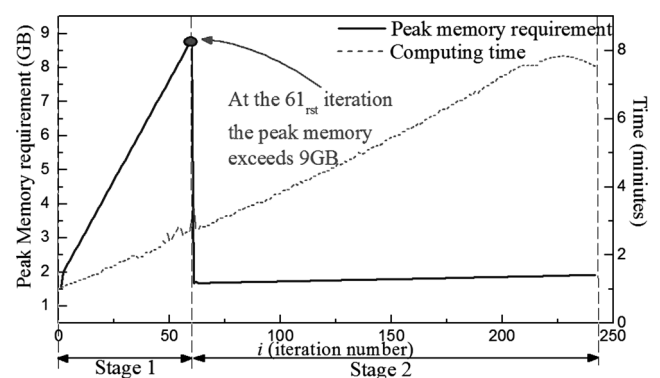
**Fig. 7** Time consumed and the peak memory requirement at each iteration

Table 10 Performance comparison of algorithms for TEAM 25 problem

Algorithm	OF ($\times 10^{-4} T^2$)	R_1 (mm)	L_2 (mm)	L_3 (mm)	L_4 (mm)	No. of FEM calls
GA	2.6861	7.2996	14.174	14.001	14.326	3421
SA	1.6223	7.2252	14.322	14.110	14.306	2145
HuTS	0.5009	7.3780	14.613	14.371	14.201	1580
UTS	1.0501	7.5487	14.908	14.506	14.416	931
NTS	0.6482	7.4337	14.732	14.428	14.237	575
Kriging	0.4527	7.2	14.1	14	14.5	241

Universal Tabu Search (UTS) [14], otherwise as in Table 3. Results for GA, SA, HuTS, UTS, NTS taken from [14].

perturbation but only a small one, indicating that matrix duplication and partitioning has little effect on computation times. However, it would probably be ill-advisable to draw a general conclusion from this observation as the case considered only necessitated one partitioning to be applied; for larger problems requiring further partitioning the extra time requirements may be more notable. Finally, the kriging model converges to the optimum after 241 iterations, which is much more efficient than with other methods reported in Table 10, while the quality of the answer is also better.

6 Conclusion

Kriging assisted optimisation has once again been demonstrated to perform efficiently, both in terms of a better quality of the final answer but also – and most importantly – the much smaller number of necessary function calls involving computationally expensive numerical modelling, such as finite elements in electromagnetic designs. However, some of the time gains could be compromised because of the need to construct memory hungry correlation matrices required in the kriging approach. This will not matter when using large computers, but as design tasks are increasingly solved on small computers or laptops the issue of efficient management of computer memory needs to be addressed and this has been the focus of this paper.

The structure of the correlation matrices associated with the kriging model may be exploited to avoid uncontrolled growth of such matrices for large problems to avoid the need for page swapping which will inevitably lead to slowing down of the process. In particular, it is possible to duplicate and shift data while storing only a small subset of necessary information. For large-scale multi-variable tasks, a zoom-in strategy based on sensitivity evaluation of each variable may be implemented in association with the proposed novel strategies for creating correlation matrices. With the help of these methods, the issue of memory storage can be mitigated to significant extent.

7 References

- Lebensztajn, L., Marreto, C.A.R., Costa, M.C., Coulomb, J.-L.: 'Kriging: a useful tool for electromagnetic device optimisation', *IEEE Trans. Magn.*, 2004, **40**, (2), pp. 1196–99
- Sobester, A., Leary, S.J., Keane, A.J.: 'On the design of optimisation strategies based on global response surface approximation models', *J. Global Opt.*, 2005, **33**, pp. 31–59
- Jones, D.R., Schonlau, M., Welch, W.J.: 'Efficient global optimisation of expensive black-box functions', *J. Global Opt.*, 1998, **13**, pp. 455–492
- Sykulski, J.K.: 'New trends in optimisation in electromagnetics'. IET Seventh Int. Conf. on Computation in Electromagnetics CEM., 2008, pp. 44–49, ISSN: 0537–9989
- Xiao, S., Rotaru, M., Sykulski, J.K.: 'Robust global optimisation of electromagnetic designs utilizing gradient indices and Kriging'. Compumag Budapest, PC4-6, 2013
- Xiao, S., Rotaru, M., Sykulski, J.K.: 'Adaptive weighted expected improvement with rewards approach in kriging assisted electromagnetic design', *IEEE Trans. Magn.*, 2013, **49**, (5), pp. 2057–2060
- Schweifel, H.P.: 'Numerical optimisation of computer models' (John Wiley and Sons, 1981)
- Alotto, P.G., Baumgartner, U., Freschi, F., *et al.*: 'TEAM workshop problem 22', <http://www.compumag.org/jsite/images/stories/TEAM/problem22.pdf>
- Xiao, S., Rotaru, M., Sykulski, J.K.: 'Adaptive weighted expected improvement with rewards approach in kriging assisted electromagnetic design', *IEEE Trans. Magn.*, 2013, **49**, (5), pp. 2057–2060
- Holland, J.H.: 'Adaption in natural and artificial system' (MIT Press, Cambridge, 1975)
- Hu, N.: 'Tabu search method with random moves for globally optimal design', *Int. J. Num. Meth. Eng.*, 1992, **35**, pp. 1055–1071
- Ho, S.L., Yang, S.Y., Ni, P.H., Wong, K.F.: 'An adaptive interpolating MLS based response surface model applied to design optimisations of electromagnetic devices', *IEEE Trans. Magn.*, 2007, **43**, (4), pp. 1593–1596
- Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: 'Optimisation by simulated annealing', *Science*, 1983, **220**, pp. 671–680
- Hajji, O., Brisset, S., Brochet, P.: 'A new tabu search method for optimisation with continuous parameters', *IEEE Trans. Magn.*, 2004, **40**, (2), pp. 1184–1187
- Yang, S.Y., Bai, Y.N., Zhang, G.H., Wu, L.: 'An improved population-based incremental learning method for inverse problems'. Proc. Automation Congress, 2008, pp. 1–4
- Coelho, L.S., Alotto, P.: 'Global optimisation of electromagnetic devices using an exponential quantum-behaved particle swarm optimizer', *IEEE Trans. Magn.*, 2008, **44**, (6), pp. 1074–1077
- Hogg, T., Portnov, D.S.: 'Quantum optimisation', *Inform. Sci.*, 2000, **128**, (3–4), pp. 181–197
- Sun, J., Feng, B., Xu, W.: 'Particle swarm optimisation with particles having quantum behavior'. Proc. Congress Evolution of Computation, Portland, OR, 2004, pp. 325–331
- Krohling, R.A., Coelho, L.S.: 'PSO-E: Particle swarm with exponential distribution'. Proc. IEEE Congress Evolution Computation, Vancouver, BC, Canada, 2006, pp. 5577–5582
- Kuntsevitch, A.V., Magele, C., Molinari, G.: 'Multiobjective optimization in magnetostatics: a proposal for benchmark problems', *IEEE Trans. Magn.*, 2002, **32**, (3), pp. 1238–1241
- Takahashi, N.: 'Optimisation of die press model'. Proc. TEAM Workshop Sixth Round, Okayama, Japan, March 1996, pp. 20–21