# Coordination of Coupled Black Box Simulations in the Construction of Metamodels

Husslage, B.G.M.; van Dam, E.R.; den Hertog, D.; Stehouwer, H.P.; Stinstra, E.

*Publication date:*
2003

*Document Version*
Publisher's PDF, also known as Version of record

Link to publication in Tilburg University Research Portal

*Citation for published version (APA):*
Husslage, B. G. M., van Dam, E. R., den Hertog, D., Stehouwer, H. P., & Stinstra, E. (2003). *Coordination of Coupled Black Box Simulations in the Construction of Metamodels*. (CentER Discussion Paper; Vol. 2003-2). Operations research.

CentER

Discussion Paper

No. 2003–02

# COORDINATION OF COUPLED BLACK BOX SIMULATIONS IN THE CONSTRUCTION OF METAMODELS

By Bart Husslage, Edwin van Dam, Dick den Hertog,
Peter Stehouwer, Erwin Stinstra

January 2003

ISSN 0924-7815

TILBURG UNIVERSITY

# Coordination of coupled black box simulations in the construction of metamodels

Bart Husslage[a, c] Edwin van Dam[a] Dick den Hertog[a]
Peter Stehouwer[b] Erwin Stinstra[b]

## Abstract

This paper introduces methods to coordinate black box simulations in the construction of metamodels for situations in which we have to deal with coupled black boxes. We define three coordination methods: parallel simulation, sequential simulation and sequential modeling. To compare these three methods we focus on five aspects: throughput time, flexibility, simulated product designs, coordination complexity and the use of prior information. Special attention is given to the throughput time aspect. For this aspect we derive mathematical formulas and we give relations between the throughput times of the three coordination methods. At the end of this paper we summarize the results and give recommendations on the choice of a suitable coordination method.

**Keywords:** Design optimization; Coupling; Coordination; Simulation; Black box; Metamodel.

**JEL Classification Number:** C00.

## 1 Introduction

Simulation tools are frequently used nowadays in the design process to predict product or process characteristics. By the complexity of many of those simulation tools there are often no explicit input-output formulas known. This is why these tools are referred to as black boxes. Further, simulation runs are often very time-consuming, so the number of simulated designs is limited in practice.

A way to gain insight in a product is to simulate a large number of different product designs. However, this takes a huge amount of time. In the literature it has been proposed to replace the black boxes by approximation models, or metamodels [4]. Equivalent terms that appear in the literature are compact models, surrogate models, and response surface models. With such metamodels we can evaluate product designs relatively fast and thus gain insight in the

[a] Department of Econometrics & Operations Research, Tilburg University, P.O. Box 90153, 5000 LE Tilburg, The Netherlands.

[b] Centre for Quantitative Methods, P.O. Box 414, 5600 AK Eindhoven, The Netherlands.

[c] Corresponding author; *e-mail address:* B.G.M.Husslage@uvt.nl.

product and find optimal and robust designs in an admissible time. In case we are dealing with only a single black box the design optimization process can be divided into four basic steps, e.g., see [8]:

1. **Problem specification:** At this step we examine the most important features of the design problem. This includes the definition of all design (or input) and response (or output) parameters of the black box. Further, we investigate the restrictions on parameter settings or combinations of parameters. All these restrictions together form the design space for the product.

2. **Design of computer experiments:** Within the design space we must choose a (limited) number of designs that we are going to simulate. These designs should be spread out over the design space, in order to yield information about the black box behavior. A set of designs is described by a simulation scheme. After the construction of such a scheme we can simulate the corresponding designs.

3. **Metamodeling:** We then try to fit models to the found simulation results. The obtained metamodels describe the response parameters as function of the design parameters.

4. **Optimization and robust design:** Finally, we make use of the metamodels and apply optimization techniques to compute an optimal and/or robust product design within the product design space.

The manufacturing industry produces a lot of complex products that consist of several coupled components. Due to the complexity of many components their design is often distributed over specialized design teams. Each of those teams uses its own black box(es) to evaluate the component designs. Because of the coupling between the components, the used black boxes are also coupled.

In the engineering practice Multidisciplinary Design Optimization (MDO) techniques are often used to deal with coupled problems, e.g., see [3], [6], and [7]. These techniques are based on optimization procedures that iteratively solve several small optimization problems in order to gradually converge to the optimal solution.

In this paper, however, we are not just interested in the optimal solution, but we also want to gain insight in the product behavior in order to design a reliable product. This is accomplished by an efficient construction of metamodels for the product as a whole. Because of the large number of design and response parameters of the product we cannot do this all-at-once, e.g., using the four steps mentioned above. This is why we must exploit the structure of the product in order to be able to model the product efficiently. Therefore, we focus on the construction of metamodels for the black boxes. However, due to the coupling between the black boxes we must also coordinate the modeling process. We do this by defining a coordination method that controls the order of the simulation runs and the construction of metamodels. Further, this coordination method enables us to exploit the coupling between the black boxes and helps to construct metamodels for all black boxes in an efficient way. The collection of these models then implicitly forms the required metamodel for the product.

In the upcoming sections we discuss several aspects of coordination methods.

Extensions of other steps in the design process will be published in forthcoming papers. This paper is organized as follows. Section 2 discusses the coupling between black boxes and gives an example of such a structure. Section 3 introduces the notion of coordination method and defines three different methods. It also defines five important aspects of coordination methods that are used for comparison. Section 4 discusses the throughput time aspect mathematically and Section 5 is devoted to the remaining four aspects. This latter section also compares the coordination methods with respect to all five aspects. Finally, Section 6 gives the conclusions and some topics for further research.

## 2 Black box coupling

Every black box has several design and response parameters. The design parameters can be divided into local design parameters and linking design parameters. Local design parameters are input to just one black box, whereas linking design parameters are input to multiple black boxes. There is also a special type of input parameter, i.e., the response input parameter, which is a black box response that is input to other black boxes. In this paper we assume that the response input parameters do not form cycles between the black boxes. Figure 1 shows the design and response parameters, as well as all possible couplings, for two coupled black boxes.

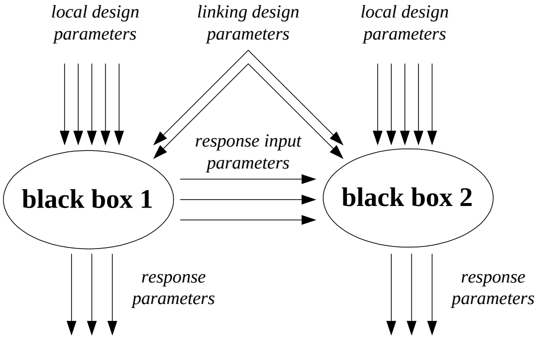

Figure 1: Design and response parameters for two coupled black boxes.

From Figure 1 it can be seen that black boxes can be coupled by linking design parameters or response input parameters. The presence of response input parameters gives rise to the need for a coordination method. This is why this paper focuses on coupling by response input parameters only, the case of coupling by linking design parameters will be discussed in a forthcoming paper. In this latter case we can couple the simulation schemes for the coupled black boxes beforehand, i.e., before the simulations take place.

Under the assumption that the response input parameters do not form cycles, the construction of simulation schemes depends on simulation results of preceding black boxes. Therefore, this type of coupling can be represented by a directed graph in which the nodes represent the black boxes and the arcs represent their relations, or the couplings. The assumption that there exist no cycles in the directed graph is substantiated by problems found in the practice of CQM. The nonexistence of such cycles gives our directed graph a forward structure, i.e., there is an explicit precedence ordering of the black boxes. This is why we refer to the directed graph as a black box chain. Throughout this paper we use the following notation to indicate the black boxes in the chain and their characteristics:

| | |
|---|---|
| $B$ : | set of black boxes, $B = \{1, 2, \ldots\}$. |
| $P_b$ : | set of all black boxes that directly precede black box $b \in B$. |
| $B_e$ : | set of all black boxes with no successors, i.e., $B_e = B \setminus \left(\bigcup_{b \in B} P_b\right)$. |
| $n_b$ : | number of required simulation runs at black box $b \in B$. |
| $s_b$ : | time per simulation run at black box $b \in B$. |

Using this notation we can always number the black boxes in such a way that their numbering reflects the precedence ordering in the chain, i.e., $b \notin P_{b'}$ if $b \geq b^{'}$. Further, note that $P_b = \emptyset$ if black box $b \in B$ is at the beginning of the chain and $b \in B_e$ if black box $b$ is at the end of the chain.

The numbers of required simulation runs at each black box are determined a priori by the designers, based on the number of design parameters and the expected black box behavior. After the simulation runs have been carried out it may happen that we do not have enough information to construct proper metamodels for some of the black boxes. Then we need to execute an additional set of simulation runs for these black boxes. Section 5 briefly discusses a way to efficiently deal with this problem by means of a two-stage simulation procedure.

Figure 2 gives an example of the coupling between multiple black boxes by response input parameters. This example is used in Section 4 to compute the throughput time for several coordination methods. In the figure an arc represents one or more response input parameters. Note that there may be multiple independent black box chains within the product structure of a certain design problem. All these chains can be dealt with separately in the way described here.

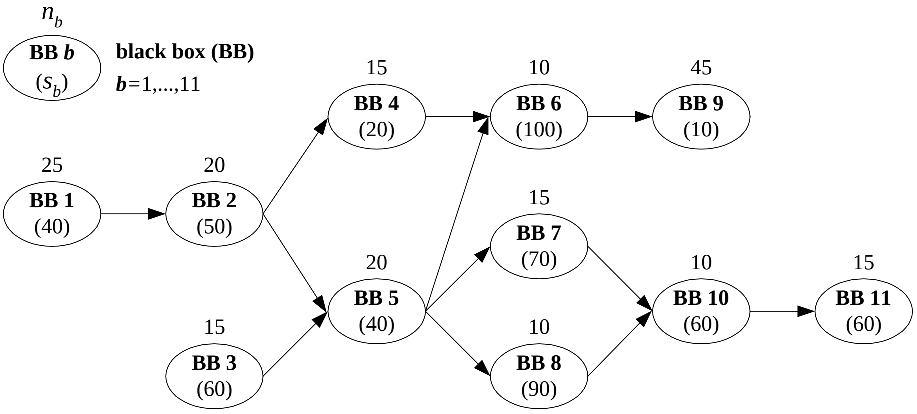

Figure 2: Example of the coupling between multiple black boxes by response input parameters.

# 3 Coordination methods

A coordination method is a rule that determines the order in which simulation runs are carried out and metamodels constructed. In this paper we introduce and analyze the following three coordination methods:

1. **Parallel simulation:** The simulation schemes of all black boxes are run concurrently and independent of each other.

2. **Sequential simulation:** Individual designs in the simulation schemes are sequentially run, following the precedence ordering in the black box chain and using simulation results of predecessors.

3. **Sequential modeling:** Simulation schemes are sequentially run, following the precedence ordering in the black box chain. Every simulation scheme is run completely and all simulation and modeling results are used in the construction of simulation schemes for successors.

When the whole simulation scheme of a black box has been run we can construct a metamodel for that black box, based on the simulation results found, e.g., see [5] and [1]. In order to be able to compare the three coordination methods we look at the following five aspects:

1. **Throughput time:** The total time it takes to carry out all simulation runs needed to construct a metamodel for every black box.

2. **Flexibility:** The sensitivity of constructed metamodels to changes in other black boxes or the product structure. A flexible coordination method is not very sensitive to such changes.

3. **Simulated product designs:** A product design is a particular setting of all design parameters that are input to the product. All of these design parameters are also input to one or more of the black boxes, which are in turn used to evaluate the product. When a certain setting of design parameters is used as input to all these black boxes and the resulting simulation outcomes are again used as inputs for succeeding black boxes, then we say that we have simulated a product design. An advantage of such simulated product designs is that we obtain information about feasible product designs before we start the optimization process, i.e., we already have lower bounds. Further, simulating product designs also increase the credibility of the used optimization and (robust) design approach.

4. **Coordination complexity:** This refers to the amount of communication and time that is needed to implement a coordination method. It also includes extra costs that are incurred by, for example, the need for an automated communication system.

5. **Use of prior information:** This is the use of simulation and/or modeling results from preceding black boxes. These results can help to determine the designs that are expected to yield the most valuable information about the product. Note that in case of response input parameters the use of prior information is a necessity to obtain simulated product designs.

In the next section special attention is given to the throughput time aspect.

# 4 Throughput time

The throughput time for a coordination method is an important aspect. From a time-to-market perspective it is desirable to have short product development times, so the throughput time should preferably be small. Because the construction time of metamodels is negligible, with respect to the simulation run time of designs, we leave it out. Below we discuss the computation of throughput times for the three coordination methods. To clarify the derived formulas we use the black box chain in Figure 2 as a numerical example. The actual time unit is not important for our discussion, we assume it is minutes.

## 4.1 Parallel simulation

With parallel simulation all simulation schemes are run concurrently and independently. The corresponding throughput time, denoted by $TT_{parallel}$, is equal to the maximum of the total simulation times at every black box:

$$TT_{parallel} = \max_{b \in B} n_b s_b \; . \tag{1}$$

When a small increase in the simulation time of a certain black box increases the throughput time, then we call this black box a bottleneck. In parallel simulation the bottlenecks are all black boxes $\tilde{b} \in B$ that satisfy:

$$\tilde{b} = \arg\max_{b \in B} n_b s_b \; . \tag{2}$$

For the black box chain in Figure 2 it can readily be computed that $TT_{parallel}$ is equal to 1050 minutes and that black box 7 (BB 7) forms the bottleneck.

## 4.2 Sequential simulation

Sequential simulation can be viewed as sending one design after another through the entire black box chain, where the designs are simulated at every black box that they encounter. This is why we refer to these designs as globally simulated designs. The maximum number of such designs, say $\hat{n}$, is restricted by the minimum of the number of required simulation runs at every black box, i.e., $\hat{n} = \min_{b \in B} n_b$. Because $\hat{n}$ is a minimum there may be several black boxes that require more simulation runs. The corresponding simulation results are only used locally, i.e., at a certain black box, and therefore referred to as locally simulated designs. Clearly, every black box $b \in B$ needs $\hat{n}$ globally and $n_b - \hat{n}$ locally simulated designs.

Focusing on the globally simulated designs we see that such a design can only be simulated at a particular black box when its simulation has finished at all preceding black boxes. In order to compute the throughput time of the sequential simulation method we first introduce the throughput time function $f_b(n)$. This function gives the minimal time it takes for $n$ globally simulated designs to be finished at black box $b \in B$. Due to this definition we can write $f_b(n)$ as:

$$f_b(n) = \max \left\{ s_b + \max_{b' \in P_b} f_{b'}(n) \; , \; s_b + f_b(n-1) \right\} , \; n \geq 1 \, . \tag{3}$$

The interpretation of this formula is that black box $b$ has to wait until all its predecessors have run $n$ simulations and that it has to wait until it has run $n-1$ simulations itself, before it can start with the $n$-th simulation run. Further, note that $f_b(0) = 0$ implies that:

$$f_b(1) = s_b + \max_{b' \in P_b} f_{b'}(1)\,. \tag{4}$$

This formula gives the longest path up to black box $b$, e.g., see [2], starting at a black box in a beginning of the chain. Because (3) is dynamic in the variables $b$ and $n$ we can rewrite it as:

$$\begin{aligned}
& f_b(n) \\
= \;& \max\left\{ s_b + \max_{b' \in P_b} f_{b'}(n)\ ,\ \max\left\{ 2s_b + \max_{b' \in P_b} f_{b'}(n-1)\ ,\ 2s_b + f_b(n-2) \right\} \right\} \\
= \;& \max\left\{ s_b + \max_{b' \in P_b} f_{b'}(n)\ ,\ 2s_b + \max_{b' \in P_b} f_{b'}(n-1)\ ,\ 2s_b + f_b(n-2) \right\} \\
= \;& \max\left\{ s_b + \max_{b' \in P_b} f_{b'}(n)\ ,\ 2s_b + \max_{b' \in P_b} f_{b'}(n-1)\ ,\ 3s_b + \max_{b' \in P_b} f_{b'}(n-2)\ , \right. \\
& \left. \ldots\ ,\ (n-1)s_b + \max_{b' \in P_b} f_{b'}(2)\ ,\ ns_b + \max_{b' \in P_b} f_{b'}(1) \right\} \\
= \;& \max_{k=1,\ldots,n} \left\{ ks_b + \max_{b' \in P_b} f_{b'}(n+1-k) \right\},\ n \geq 1\,.
\end{aligned} \tag{5}$$

It can be proven that $f_b(n)$ is convex in $n \geq 1$. Therefore, (5) can be simplified to the following maximum function:

$$\begin{aligned}
f_b(n) \;&=\; \max_{k \in \{1,n\}} \left\{ ks_b + \max_{b' \in P_b} f_{b'}(n+1-k) \right\} \\
&=\; \max\left\{ s_b + \max_{b' \in P_b} f_{b'}(n)\ ,\ ns_b + \max_{b' \in P_b} f_{b'}(1) \right\} \\
&\overset{(4)}{=}\; \max\left\{ s_b + \max_{b' \in P_b} f_{b'}(n)\ ,\ (n-1)s_b + f_b(1) \right\},\ n \geq 2\,.
\end{aligned} \tag{6}$$

This reduces (3) to a formula that is only dynamic in the variable $b \in B$.

Next, we define the following two sets:

$C_{b,n}$ : set of possible bottlenecks up to black box $b \in B$ for $n$ globally simulated designs.
$C_n$ : set of bottlenecks for $n$ globally simulated designs.

Note that the black box chain structure can cause the sets $C_{b,n}$ and $C_n$ to differ significantly for distinct values of $n$. For $n = 1$ we use (4) and $C_{b,1} = \{b\} \cup \bigcup_{b' \in \mathcal{I}} C_{b',1}$ , where $\mathcal{I} = \left\{ b^{'} : b^{'} \in P_b\,,\ f_{b'}(1) = f_b(1) - s_b \right\}$, to compute $f_b(1)$ and the corresponding bottlenecks for all black boxes $b \in B$. Combining this information with (6) enables us to compute the throughput time $f(n)$ and the corresponding sets $C_{b,n}$ and $C_n$ for every arbitrary integer $n \geq 2$ as follows:

$$
\begin{aligned}
&f_b(n) = \max\left\{ s_b + \max_{b'\in P_b} f_{b'}(n)\ ,\ (n-1)s_b + f_b(1) \right\} ;\\
&C_{b,n} = \begin{cases} C_{b,1} \cup \bigcup\limits_{b'\in\mathcal{I}} C_{b',n} & \text{if } f_b(n) = (n-1)s_b + f_b(1)\,,\\ \{b\} \cup \bigcup\limits_{b'\in\mathcal{I}} C_{b',n} & \text{otherwise}\,, \end{cases}\\
&\text{where } \mathcal{I} = \left\{ b^{'}\ :\ b^{'} \in P_b\ ,\ f_{b'}(n) = f_b(n) - s_b \right\};\\
&\\
&f(n) = \max_{b\in B_e} f_b(n)\,;\\
&C_n = \bigcup_{b\in\mathcal{J}} C_{b,n}\ ,\ \text{where } \mathcal{J} = \{ b\ :\ b \in B_e\ ,\ f_b(n) = f(n)\,\}.
\end{aligned}
\tag{$TT$}
$$

Using $(TT)$ we can compute the minimal time that is needed to simulate all $\hat{n}$ globally simulated designs, i.e., $f(\hat{n})$. To compute the throughput time we must include the time needed by the locally simulated designs. In this respect, note that a black box may be idle, i.e., not simulating, for several periods of time during the whole simulation process of the globally simulated designs. In case we are dealing with preemptive simulation runs we can start the simulation of locally simulated designs in these idle periods. The throughput time of the sequential simulation method, denoted by $TT_{seqsim,\,pre}$, is then given by:

$$TT_{seqsim,\,pre} = \max\left\{ \max_{b\in B} n_b s_b\ ,\ f(\hat{n}) \right\} \overset{(1)}{=} \max\{\, TT_{parallel}\ ,\ f(\hat{n})\}\ . \tag{7}$$

In case $TT_{seqsim,\,pre} = TT_{parallel}$ the black boxes $\tilde{b} \in B$ that satisfy (2) form bottlenecks. If $TT_{seqsim,\,pre} = f(\hat{n})$ the black boxes $b \in C_{\hat{n}}$ are bottlenecks. Note that in this latter case not all bottlenecks may have the same impact on the throughput time. The exact impact of every bottleneck can easily be computed once the bottlenecks are known.

Unfortunately, it may not be possible to stop a simulation run at some point in time and later on proceed from that point, i.e., the simulation runs may be non-preemptive. Switching between different designs may also need a lot of communication between the divisions and may therefore not be very practical. Further, waiting until all globally simulated designs are finished gives us the maximal amount of information for the determination of the designs that we simulate locally. For these reasons we suggest to use non-preemptive simulation runs and to run all locally simulated designs after the globally simulated designs are finished. The throughput time then becomes equal to:

$$TT_{seqsim} = \max_{b\in B}\{\, f_b(\hat{n}) + (n_b - \hat{n})s_b\,\}\ . \tag{8}$$

In this case the bottlenecks are given by the set $C^*_{\hat{n}} = \bigcup_{b\in\mathcal{I}} C_{b,\hat{n}}$ where $\mathcal{I} = \{b\ :\ f_b(\hat{n}) + (n_b - \hat{n})s_b = TT_{seqsim}\,\}$. As above, the impact of these bottlenecks may vary.

For the example in Figure 2 we can use $(TT)$ to compute that:

$$f(n) = \begin{cases} 250 + 90n & \text{if } n \leq 11 \\ 140 + 100n & \text{if } n \geq 11 \end{cases} . \tag{9}$$

From (9) and the fact that $\hat{n} = 10$ we get that $f(\hat{n}) = 1150$. Recall that $TT_{parallel} = 1050$ minutes, so (7) results in $TT_{seqsim,\,pre} = 1150$ minutes. Further, because $f_{11}(\hat{n}) = f(\hat{n})$ it follows from $(TT)$ that the bottlenecks are given by the set $C_{\hat{n}} = C_{11,\hat{n}} = \{1, 2, 5, 8, 10, 11\}$. Using (8) we find that $TT_{seqsim}$ is equal to 1490 minutes. Because $f_9(\hat{n}) + (n_9 - \hat{n})s_9 = TT_{seqsim}$, all black boxes in the set $C^*_{\hat{n}} = C_{9,\hat{n}} = \{1, 2, 5, 6, 9\}$ are bottlenecks.

### 4.3 Sequential modeling

With sequential modeling all simulation schemes are run sequentially, following the precedence ordering in the black box chain. Therefore, the throughput time, denoted by $TT_{seqmod}$, is equal to the longest path in the black box chain, when we take the total simulation time per black box, i.e., $n_b s_b$, on the nodes of the directed graph:

$$TT_{seqmod} = \max_{\hat{B} \subset B} \sum_{b \in \hat{B}} n_b s_b \,, \text{ where } \hat{B} \text{ is a path in the chain.} \tag{10}$$

In this equation a path is defined as a sequence of black boxes starting at a beginning of the chain, so at a black box $b$ for which $P_b = \emptyset$, and ending at an end of the chain, so at a black box $b \in B_e$. All black boxes on a longest path form bottlenecks. In Figure 2, with $n_b s_b$ on the nodes, the black boxes 1, 2, 5, 7, 10 and 11 form the longest path and, therefore, the bottlenecks. The corresponding throughput time is equal to $TT_{seqmod} = 5350$ minutes.

### 4.4 Throughput time relations

From above observations we can derive general relations between the throughput times of the different coordination methods, i.e.:

$$TT_{parallel} \overset{(7)}{\leq} TT_{seqsim,\,pre} \overset{(8)}{\leq} TT_{seqsim} \leq TT_{seqmod} \,. \tag{11}$$

The first two inequalities readily follow from observations (7) and (8). It can easily be proven that the last inequality also holds.

## 5 Comparison of aspects

Besides the throughput time we defined four other aspects in Section 3. This section discusses these aspects for the three coordination methods and summarizes the results.

### 5.1 Parallel simulation

Due to the concurrent running of the simulation schemes at all black boxes the corresponding metamodels are also constructed independently. Therefore, changes in a particular black box do only affect the accompanying metamodel and not the metamodels for other black boxes. This is why parallel simulation is not very sensitive to changes and thus very flexible.

By the independence of the black box simulations it is very unlikely that the same design settings are simulated, so there will in general be no simulated product designs. However, when the coupling of black boxes is caused by linking design parameters only, we can simulate the same settings for these linking design parameters at all black boxes and still obtain simulated product designs.

Because every design team merely operates independently there is no need for a complex organizational structure. Communication is only needed when all simulations have taken place. However, these simulation results are not used to determine new simulation schemes, so there is no prior information used.

## 5.2 Sequential simulation

At sequential simulation we use the precedence ordering in the black box chain and simulation results at the simulation process. Therefore, the globally simulated designs, as well as the locally simulated designs, use prior information. Changes in black boxes affect the simulation outcomes and thus the input to other black boxes. This clearly affects the metamodels for these black boxes, so the method is not very flexible with respect to changes in simulation tools.

When the same linking design parameter settings are simulated at all black boxes then the globally simulated designs result in simulated product designs. The settings for the linking design parameters can be determined beforehand, as will be explained in a forthcoming paper.

Unfortunately, using simulation results of preceding black boxes asks for a lot of communication between the design teams. After every globally simulated design has been run at a certain black box, the results have to be sent to all its successors. This results in a complex coordination process that needs sophisticated communication methods, which have to be supported by the design tools.

## 5.3 Sequential modeling

Using all simulation outcomes and metamodels of preceding black boxes as input implicitly affects the metamodels that are constructed. Therefore, changes in simulation tools also affect other metamodels and make this coordination method not very flexible.

Further, the use of prior information makes it possible to obtain simulated product designs, as long as the same linking design parameter settings are used. Note that the number of required simulation runs varies per black box, so we have to be careful which results to use, i.e., only the designs that have been simulated at all preceding black boxes, in order to obtain simulated product designs.

There is communication needed between the various divisions, but only after each simulation scheme has been run completely and the corresponding metamodel is constructed. The coordination process is therefore relatively simple.

## 5.4 Summary of results

In this section we summarize the results found above. Table 1 gives the five aspects mentioned in Section 3, as well as the scores for each of the three coordination methods at these aspects. Two pluses (++) indicate that the coordination method has a positive effect on a particular aspect. One plus (+) indicates a moderately positive effect. With one minus (−) the effect of the coordination method on a particular aspect is slightly negative and with two minuses (−−) this effect is negative.

Recall that in Section 4 we choose to use sequential simulation with non-preemptive simulation runs, i.e., the locally simulated designs are determined and simulated after all globally simulated designs have finished. Further, note that in Table 1 a positive effect (+ or ++) at the coordination complexity means that the coordination process is not complex.

| Coordination method →<br>Aspect ↓ | Parallel simulation | Sequential simulation | Sequential modeling |
|---|---|---|---|
| Throughput time | ++ | + | −− |
| Flexibility | ++ | −− | −− |
| Simulated product designs | − | ++ | ++ |
| Coordination complexity | ++ | −− | + |
| Use of prior information | −− | + | ++ |

Table 1: Comparison of three coordination methods with respect to five aspects.

A small throughput time, flexibility and a lack of complexity at the coordination process are the main advantages of parallel simulation. However, for the designers it is important that they obtain an accurate metamodel for the product. Parallel simulation may need several extra simulation runs, besides the designs already simulated, to include the coupling between the black boxes properly in the metamodels. Further, it is often required that there should be simulated product designs, for reasons mentioned in Section 3. This requirement makes the sequential coordination methods more favorable than the parallel coordination method, at least, in cases where there are response input parameters present.

Choosing between the two sequential coordination methods mainly depends on the throughput time and the availability of good means of communication between the divisions. Using sequential simulation results in a more complex coordination process, whereas sequential modeling yields a larger throughput time. Therefore, when dealing with large simulation times and an automatized communication system, the sequential simulation method is preferable. Sequential modeling is a good choice when communication between the divisions is hard and the simulation times are not too large.

Of course, the determination of the best coordination method is not that strict and depends on the kind of design problem we are dealing with. This is why a good examination of all aspects for each of the three coordination methods is extremely important. This can be done by using Table 1 and the formulas derived in Section 4.

Section 2 mentions that an initial set of simulation runs per black box may not suffice to construct proper metamodels. To deal with this problem there is often a two-stage simulation procedure used. For the first stage we advise to use parallel simulation for the simulation of the initial set of designs, or initial simulation scheme. This gives a global idea about the black box behavior and the most important parts of the product design space. At the second stage we can combine the simulation results found with a sequential method and simulate extra sets of designs. These extra simulation runs will give more insight in the most important parts of the product design space. Therefore, the resulting metamodels will give good representations for the whole product design space, but still emphasize the most important parts of it.

## 6 Conclusions and further research

In order to gain insight in product and component behavior we replace time-consuming simulation tools by approximation models, or metamodels. The construction of such models is based on sets of simulated designs. Products that consist of several components have, normally speaking, several time-consuming simulation tools, or black boxes, that are used at the component design processes. Coupling of the components causes coupling of the black boxes and results in a need for communication between the different design teams. These coupled design processes can be controlled by a coordination method.

We introduced three coordination methods, derived formulas for their throughput times and gave four other aspects to compare the methods. The results of this comparison can be found in Table 1. This table, along with the throughput time formulas, can be used to determine the best coordination method for a specific design problem. After this, the chosen method is used to construct simulation schemes for all black boxes, following a procedure which will be discussed in a forthcoming paper. Based on the simulation results metamodels can be constructed for all black boxes. These metamodels can then be used for finding an optimal and/or robust design. The construction of metamodels and the optimization process will also be discussed in forthcoming papers.

## Acknowledgements

The authors would like to thank Pascal Etman of Eindhoven University of Technology for sharing his ideas on the concept of coordination in design optimization problems.

## References

[1] Barton R.R. (1998). “Simulation Metamodels.” *Proceedings of the 1998 Winter Simulation Conference*, 167-176, Washington, USA.

[2] Bazaara, M.S., J.J. Jarvis and H.D. Sherali (1990). *Linear Programming and Network Flows.* John Wiley & Sons, New York.

[3] Cramer, E.J., J.E. Dennis Jr., P.D. Frank, R.M. Lewis and G.R. Shubin (1994). "Problem Formulation for Multidisciplinary Optimization." *SIAM Journal on Optimization* 4, 754-776.

[4] Kleijnen, J.P.C. (1987). "Statistical Tools for Simulation Practitioners." Marcel Dekker, Inc., New York.

[5] Kleijnen, J.P.C. and R.G. Sargent (2000). "A Methodology for Fitting and Validating Metamodels in Simulation." *European Journal of Operational Research* 120, 14-29.

[6] Kodiyalam, S. and J. Sobieszczanski-Sobieski (2001). "Multidisciplinary Design Optimization - Some Formal Methods, Framework Requirements, and Application to Vehicle Design." *International Journal of Vehicle Design*, Special Issue, 3-22.

[7] Sobieszczanski-Sobieski, J. and R.T. Hafka (1996). "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments." Presented at the 34th AIAA Aerospace Sciences Meeting and Exhibit.

[8] Stehouwer, H.P. and D. den Hertog (1999). "Simulation-Based Design Optimisation: Methodology and Applications." *Proceedings of the First ASMO UK/ISSMO Conference on Engineering Design Optimization*, Ilkley, UK.