J Mar Sci Technol (2016) 21:325–333
DOI 10.1007/s00773-015-0354-x

ORIGINAL ARTICLE

A decomposition method for surrogate models of large scale structures

Serdar A. Koroglu[1] · Ahmet Ergin[1]

Received: 25 February 2015 / Accepted: 2 November 2015 / Published online: 23 November 2015

Abstract Analysis is computationally the most expensive part of optimization. Surrogate models, which are approximate but faster statistical models, can be used in place of more precise but more computer-intensive methods like finite element method to improve efficiency. Unfortunately, the surrogate models are limited by the number of model parameters. So large-scale problems cannot be fully defined by a single surrogate model. Furthermore, current domain decomposition methods cannot be used with black-box models. This study presents a novel approach to design thin-walled structures using surrogate models that overcome the curse of dimensionality by a special decomposition method. A parametric panel structure is defined as a building block. An interface is developed to maintain compatibility across the blocks. Finally, an iterative algorithm finds the displaced state using only local information. Three test structures are used to show the convergence of the algorithm for static analysis. In these sample cases, number of steps required for convergence of the error did not change with the number of panels. This approach offers many benefits including automatic design creation and optimization, effective usage of stream processors and model reuse.

Keywords Surrogate model · Domain decomposition · Design optimization · Curse of dimensionality

1 Introduction

Engineering design optimization is mainly based on two aspects: search and analysis. While there are currently many search approaches with varying degrees of applicability, the analysis methods are generally restricted and they are the most time-consuming part of the search process.

The analysis bottleneck has driven the research community to improve the efficiency of the search methods with different perspectives. For example, parallel computation trend offered some solutions in the search techniques. Evolutionary algorithms (EA) are natively suitable for concurrent search [1]. Sometimes EA's are applied at the global level for multilevel search approaches [11].

Many solutions regarding linear systems of equations have also been offered in literature. Pre-conditioners like incomplete factorization [8] and approximate inverses [9] are used with relative ease. Furthermore, alternative methods like domain decomposition methods are active research topics, where both discretization and the differential problem are being utilized. Schwarz methods [7] decomposes the problem domain with overlapping sub-domains and the coordination of solution is maintained by passing intermediate solutions as Dirichlet boundary condition to the neighbouring sub-domain with a coarse grid usage over the global domain.

Direct substructuring, being one of the earliest attempts of factorization of the problem, was a non-overlapping domain decomposition method [15]. Its main purpose was overcoming the memory limitations for large-scale structures. On the other hand, it was also suitable for parallel implementation. As a top-down approach, the global model has to be decomposed into subdomains. The nodes at the

✉ Serdar A. Koroglu
info@serdarkoroglu.com

Ahmet Ergin
ergina@itu.edu.tr

[1] Faculty of Naval Architecture and Ocean Engineering, Istanbul Technical University, Istanbul, Turkey

interfaces are selected as master degrees of freedom. DOFs of interior nodes are eliminated by static condensation creating a superelement for the subdomain. The resulting Schur complement system is solved and the remaining solution for the internal nodes are recovered by a back-substitution process. Repeating patterns in the structure create an advantage by calculating only a single Schur complement for each part.

Faster solution methods have appeared since then as iterative substructuring algorithms, where distributed Krylov subspace methods have been utilised in the solution procedure of Schur complement system [10, 13].

In recent years, the computer-intensive analyses are replaced with approximate but faster statistical models, referred to as surrogate models [16] (a.k.a. meta-model, response surface). The main idea is based on the construction of an alternative model by sampling the search space with the data provided from computationally expensive simulations increase the search efficiency. Although the computer simulation results have no measurement error, they still have uncertainty with the predictions. From that perspective there have been many efforts to efficiently design computer experiments and have an estimation of uncertainty of predictions [17]. The surrogate models can be used in search directly as in efficient global optimisation [12], as local models [6], as multi-fidelity models in space mapping techniques [2], or in reliability analysis [4].

These models are very powerful but they are limited by the number of parameters the problem has. The required number of samples to build the model increases exponentially with increasing number of parameters. Some improvements have been made by sampling techniques like space filling [14] or sequential sampling [20], but still, the curse of dimensionality problem [3] has not been solved yet. The problem of handling models with large number of variables was also emphasised along with three other challenges in a panel discussion that was held in 9th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimisation in Atlanta [18].

Until now, the attempts to utilise the surrogate models were merely bringing together the common problem definitions or their limited/localised counterparts with common surrogate modelling methods. One of the implications of the “no free lunch theorem” [21] is that there cannot be a single optimum solution for every search problem or, in other words, a generic solution may not be the best approach for a given problem.

This study is based on these motives. The new proposed approach was originated from the problem point of view. Considering the limitations of statistical methods, the speed up has been achieved by taking advantage of the geometrical order and patterns that exist in common structures. Regarding the parallel computing trend, the approach was designed with the least possible interdependence between the components. Besides, to promote the reuse of components and to benefit from using stream processing technologies [19], the models have been created as generic as possible. A stiffened panel is defined as a generic building block of a structure. The behaviour of the global structure is calculated from the outputs of surrogates of those panels and those outputs are used exclusively. This in turn, enables a parallel mechanism that scales easily and confines the parameter limitations to the panel level only. The proposed algorithm to decompose the structure has showed constant time complexity behaviour in the test cases, i.e., number of iterations did not change with increasing number of sub-structures.

2 The new building block concept

Structural design may be summarised as the determination of a material distribution that will sustain the required amount of load while carrying its duty, constrained by the operating conditions, cost, design life, etc. It is a search process and in practice, this search starts in a confined search space (for example topology is given a priori) because otherwise it would be impossible to consider every possible design. In this study, the search space is divided into two levels, and an algorithm is proposed to coordinate these levels.

The first level is a parametric stiffened panel block and its corresponding approximate statistical model is generic enough to fit any part of the structure and it is capable of representing any operating condition. The block may be considered as a finite element with a design space of its own. By using a building block like this, it is possible to create the statistical model with reduced number of parameters, compared to a whole structural system. A representative panel can be seen in Fig. 1. Although a rectangular panel is given as an example, curved panels can be defined as well, by addition of parameters that define the

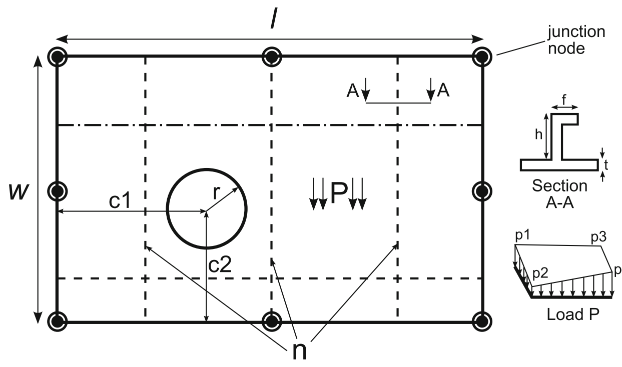

Fig. 1 A representative parametric panel

curvature. The block has information about geometry, material, different load cases and it has junction nodes that define its degrees of freedom (DOF). Those fixed number of nodes are required to cover any type of geometry and to have a compliant interface with other neighbour blocks.

On the second level, the blocks have to be coordinated to obtain the behaviour of the system as a whole. The algorithm does that by iteratively using gathered information from the concurrent solutions of the problems at block level. The details of the algorithm are given in the next section.

This approach, besides solving the main problem of curse of dimensionality, comes with many other side benefits. The building block can be defined generic enough to represent various loading conditions, geometry and material properties. So the construction of the surrogate model is a one-time process and the model can be reused as any sub-component, for many applications. It will be a more demanding task to build one of these models, compared to the surrogate models currently in practice. But in the long term, there is no need to create different models each time when a new design required.

The building block is a fixed model after training. In other words, no matter what input is fed, same operations are applied and that makes it suitable to implement the system in modern stream processors like GPUs, to have a huge advantage in terms of parallel processing. Even dedicated integrated solutions can be created after a well-established surrogate model creation.

If the design space can be defined in terms of number, connection topology and orientation of blocks, complying with producibility and ergonomic constraints and also feasibility of geometries for both block and design level is satisfied; optimum designs can be created automatically even at topological level. Because the problems regarding the model of the building block is solved in the model building stage, encapsulation of the model is accomplished as a result and it is necessary for an automated design process where no user interaction/correction is preferable.

Another advantage of this approach is the direct mapping of the input to the output. For example, statistical model building does not differentiate between input–output data couples from a simple linear analysis with data from a more complex non-linear analysis. In the end the model will generate the results at same speed given a proper model is fitted with the data. Only the one time data generation in the model building stage is adversely affected with the complexity of the simulation.

2.1 Complexity comparison of FEM vs surrogate based decomposition

As will be explained in Section 3.4, no surrogate model has been constructed for the present study. However, even a qualitative comparison can clearly highlight the differences.

First major difference is the variable that defines complexity. It is the number of nodes for FEM and number of design parameters for a surrogate model. For instance, when a stiffener is added to the system, only the parameter that represents the number of stiffeners changes in value for the surrogate model, whereas the number of nodes for each additional stiffener is added to the FEM model node pool.

For many substructuring methods, the boundaries of each subsection have to be determined. But with surrogates, the design can be built parametrically in terms of the panels. So no need for explicit geometrical decomposition.

While for FEM, repeating parts in structure are advantageous, surrogate models are already the single repeating model that defines the whole structure by itself.

Finally, when decomposed, the distributed elements of FEM equations require dedicated CPU time. On the other hand, there is only a single surrogate model so vector processing can be used. This in turn when compared with message passing architectures, greatly improves parallelization. It is only limited by the number of panels that the structure can be defined.

3 Decomposition algorithm

To determine the global behaviour of the structural system for a static analysis, the displacements of all the junction nodes have to be predicted. If the displaced state of the system is considered, any deviation of the junction nodes' equilibrium position creates a restoring force that points to the undisturbed position. Once all the restoring forces are known, this information can be used to find the equilibrium configuration. The restoring force can be found as follows: for each junction node, each panel that shares the node are isolated and analysed separately. Current estimation of the junction nodes' position/direction is imposed as displacement boundary condition. After the analysis, reaction force/moment can be derived from each panel. The sum of these force/moment values is the reaction force/moment when panels are connected and same deviation has occurred. By using this, the equilibrium of a single DOF for a junction node can be found knowing that all other DOFs are kept constant. However, considering the structure as a whole, a search problem with (number of nodes x DOF) number of parameters has to be solved.

This study presents a mechanism to coordinate these local searches to find the global equilibrium of the system. Analogous to the momentum concept in classical mechanics, the algorithm either increases or decreases the step size of the search depending on the proximity to the solution.

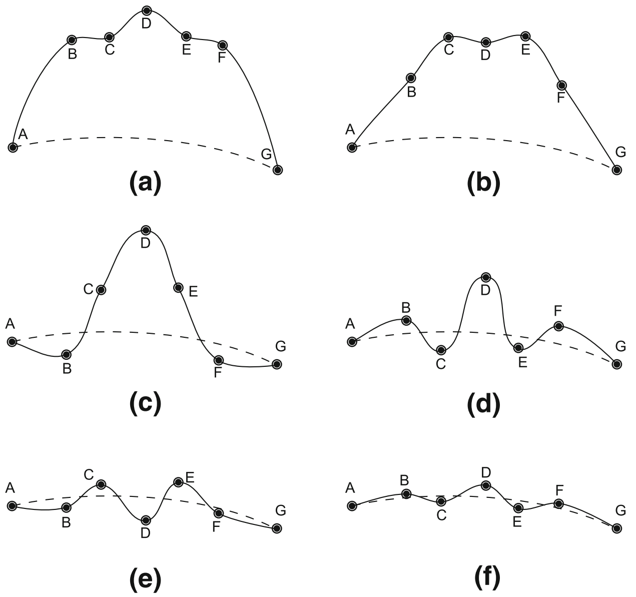

Fig. 2 Representative example for the algorithm

An example, as illustrated in Fig. 2, has been constructed to clarify the logic behind the algorithm. The solid curve shows an estimated displaced state in an intermediate stage of the search. The dashed curve represents the actual equilibrium configuration to be found. First of all, a restoring force which is directed downwards is expected to be generated at node D. As a result of this restoring force, the algorithm updates the position of node D with a lower position. If the new position results in a restoring force with the same direction, the position of the node is updated to an even lower position. Otherwise, the update direction is changed and the amount of the change is decreased because now the solution is trapped between the previous estimate and current estimate. This works for each node and each DOF separately. But globally the change is applied to all nodes and DOFs at the same time and the solution cannot be located as in the single search.

To overcome this problem a momentum like step size is applied separately for each node and associated DOF. The step sizes are increased if the restoring force's/moment's sign does not change; otherwise, it is decreased. So even though the solution is not trapped, the step size is changed. By using this strategy, the nodes which are relatively far away from equilibrium gain momentum and drag the neighbouring nodes with lower momentum. For example, in Fig. 2, after a close estimation of the positions of C, D and E (far from actual equilibrium but locally no noticeable restoring force), they would have low step sizes. But nodes B and F would have a growing step size, and after updating their nodal position, C and E in turn would have a larger step size and finally same would apply to node D. The same principle will continue to drive the search to converge to the correct configuration in a similar manner.

3.1 Formal summary of the algorithm

The equilibrium state of the structure can be found by solving

$$\min_{x_i} \sum_i rr_i(x_i), \tag{1}$$

where i represents every (node, DOF) pair combination, x_i is displacement configuration and rr is the residual reaction force/moment for ith (node, DOF) pair. Algorithm relies on an assumption of monotonicity of residual reactions. In other words, for every x_m, x_n if

$$\operatorname{sgn}(x_m) = \operatorname{sgn}(x_n) \tag{2}$$

and

$$|x_m - x_{eq}| > |x_n - x_{eq}|, \tag{3}$$

then

$$rr_i(x_m) > rr_i(x_n), \tag{4}$$

where x_{eq} is the displacement configuration of the equilibrium. The residual reaction at each (node, DOF) is equal to zero for x_{eq}.

Algorithm starts with the initialization procedure,

$$x_i^0 = \epsilon_i \tag{5}$$

$$u_i^0 = \epsilon_i \tag{6}$$

where ϵ is a small random real number, x_i^t is the displacement of ith (node, DOF) at iteration t and u_i^t is the corresponding update coefficient.

Iteratively displacements of each (node, DOF) are updated according to the following:

$$f_{ip}^t = \text{panelReaction}_{ip}(x_i^t), \tag{7}$$

where panelReaction is the reaction force calculated for (node, DOF) of an isolated panel p represented by a surrogate model. Total residual reaction for a node

$$rr_i^t = \sum_p f_{ip}^t \tag{8}$$

$$u_i^{t+1} = \begin{cases} u_i^t \times (1+\delta) & \mathit{if}\ rr_i^t \times rr_i^{t-1} > 0 \\ u_i^t \times (1-\delta) & \mathit{if}\ rr_i^t \times rr_i^{t-1} \leq 0 \end{cases} \tag{9}$$

$$x_i^{t+1} = x_i^t + u_i^{t+1}, \tag{10}$$

where $\delta \in (0,1)$, p is the set of panels having ith (node, DOF) pair.

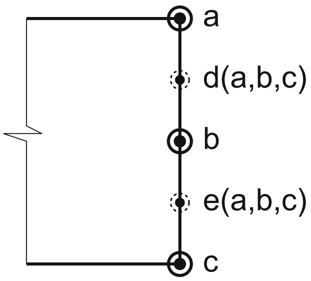

Fig. 3 Edge interpolation

Table 1 Error percentages for various edge interpolations

Panel type	3 Junc.	4 Junc.	5 Junc.
Z - flat panel	0.12	-	-
Z - panel with 5 stiffeners	26.96	13.11	6.51
X - flat panel	1.83	-	-
X - panel with 5 stiffeners	29.56	19.13	11.54
V - flat panel	0.24	-	-
V - panel with 5 stiffeners	8.13	3.096	1.32

3.2 Edge interpolation

Because there is only one surrogate model at hand for any combination of design parameters, all the building blocks' interfaces with other blocks have to be compatible. For that reason, all the surrogate models are built with a fixed number of junction nodes. As seen in Fig. 3, there are three junction nodes and the remaining nodes' DOF are represented in terms of the DOF of the junction nodes by means of an interpolating polynomial. So nodes D and E are represented by a polynomial, which has coefficients that are dependent on the position of nodes a, b and c. In this study, the example applications have three junction nodes per edge and remaining nodes are imposed as constraint equations in the form of quadratic polynomials.

To have an idea of the errors as result of constraining the edges with constraint equations, two example situations are tested. First one is two flat panels loaded with a lateral pressure and the other is two stiffened panels with five stiffeners crossing the interface.

In Table 1, the error percentages for each situation are given for the maximum displacement in the Z direction, the maximum rotation around X axis and the maximum Von Mises equivalent stress. As can be seen in the table, for a flat panel three junction nodes would suffice. But increasing the complexity, while keeping the number of junction nodes the same, generates higher errors. When the number of junction nodes is increased to four, results improve immediately. For that reason, prior to constructing the generic surrogate model, a good compromise must be found between simple and complex panel forms considering the number of junction nodes and subdivision density.

3.3 Power transform

In the course of the algorithm, large variations in nodal displacements are common. Because of the dynamic nature of the search, these fluctuations must be handled quickly. If update coefficients are adjusted directly, both displacements and corresponding update will change wildly and this will cause stability problems. Another solution is transforming the displacements to have the update cause non-uniform corrections. So irrespective of the order of magnitude, the displacements receive updates with proportional magnitude. An analogous situation arises in regression problems. Transform is applied to output variables to have constant variance in errors [5].

To preserve the sign of values, the following modified power transform was used in the algorithm:

$$\text{transform}(x, \lambda) = \begin{cases} x^{\lambda}, & \text{for}\, x > 0 \\ -|x|^{\lambda}, & \text{for}\, x \leq 0 \end{cases} \tag{11}$$

In the example applications of the next section, it has been observed that 0.15 as the λ parameter showed the best performance.

3.4 Finite element method test bench

To construct a surrogate model for this study deserves a deep and an extensive study on its own. To test the decomposition algorithm without the surrogate model itself, ANSYS finite element method software has been used as a test bench. Instead of supplying the surrogate with estimated displacements in the current iteration, the data are sent to FEM and results are harvested back for updating the current estimation. For the applications presented in the next section, a fixed FEM model (fixed geometry, mesh, loading condition, material) has been constructed for each panel block in each of the examples. Models only accept displacements of interface nodes as input and feedback reaction forces in return. The flow chart of the test bench and algorithm is given in Fig. 4.

Application of FEM shows us the upper limit for the proposed method because the surrogate model at most will behave exactly like its data generator.

4 Applications (3, 9, 25 panel)

Three example problems with varying degrees of scale and complexity are constructed to demonstrate the convergent behaviour of the algorithm. The first example consists of 3 panels with no stiffeners, the second panel has 9 panels and

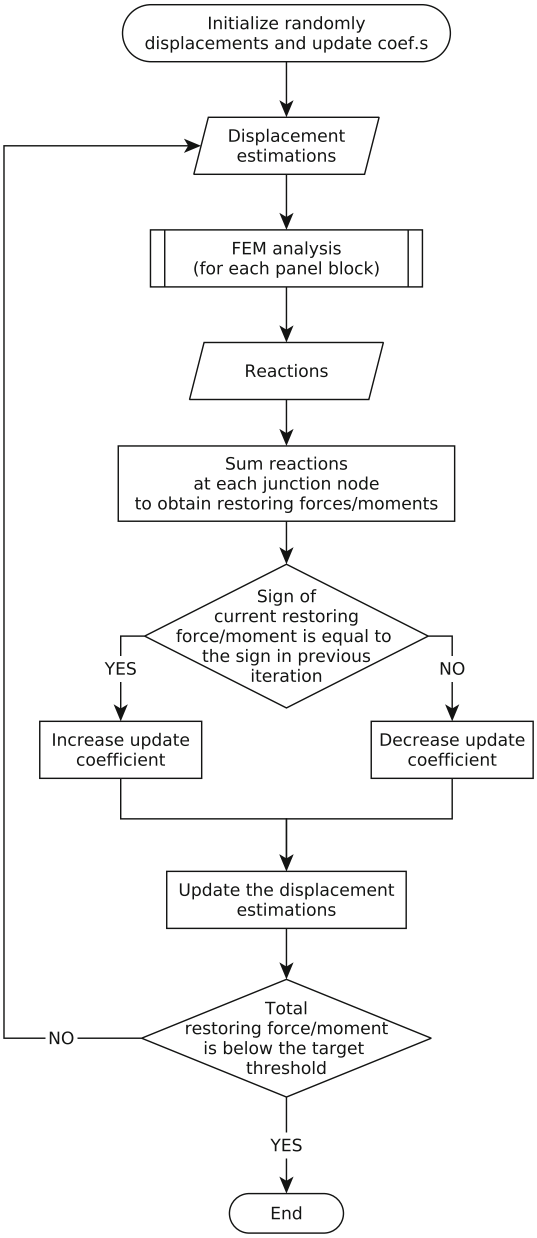
Initialize randomly displacements and update coef.s
Displacement estimations
FEM analysis (for each panel block)
Reactions
Sum reactions at each junction node to obtain restoring forces/moments
Sign of current restoring force/moment is equal to the sign in previous iteration
YES
NO
Increase update coefficient
Decrease update coefficient
Update the displacement estimations
Total restoring force/moment is below the target threshold
NO
YES
End

Fig. 4 Test bench and algorithm flow chart

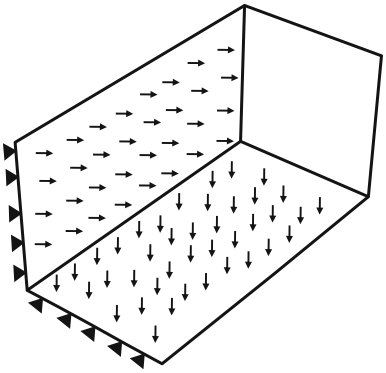

Fig. 5 3-panel structure

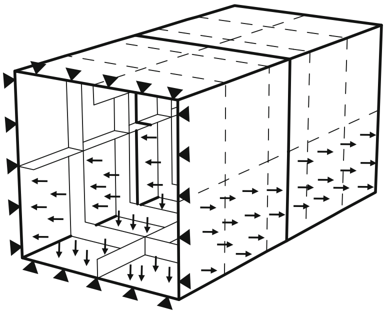

Fig. 6 9-panel structure

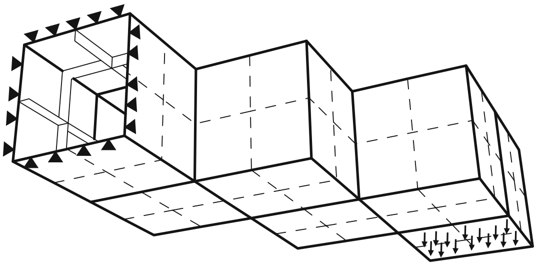

Fig. 7 25-panel structure

finally the last example has 25 panels with the configurations shown in Figs. 5, 6, 7. Problem parameters can be seen in Table 2.

Stiffened panels are analysed using ANSYS' SHELL63 element which is a 4-node shell element with 6 DOF at each node and with bending and membrane capability. 3-panel example had 75 nodes and 48 elements, 9-panel example had 934 nodes and 844 elements and finally 25-panel example had 850 nodes and 600 elements.

All the panels in all examples are modelled separately with the edge interpolation applied to each side. So around each panel there are 8 junction nodes (3 junction nodes per side) each one with 6 DOF. The first and last examples have eccentric loading while the stiffeners of the second example are placed asymmetrically. The algorithm parameters are listed in Table 3.

Starting from the zero displacement configuration, the algorithm was applied for each example separately. The progress of the algorithm has been shown in three different types of plots. Because there are so many number of variables, the monitored performance parameters were shown in a rather compact way. The bold curve represents the average and the lighter curves are +/− 1 standard deviations around the average of all values (all the nodes and DOFs).

For the update coefficients we can see that (Figs. 8, 9, 10), at the initial stages, their magnitude increases because the

Table 2 Problem parameters

3 panel		9 panel		25 panel	
Parameter	Value	Parameter	Value	Parameter	Value
Moulded breadth	1 m	Moulded breadth	1 m	Moulded breadth	1 m
Moulded length	2 m	Moulded length	2 m	Moulded length	2 m
Moulded depth	1 m	Moulded depth	1 m	Moulded depth	1 m
Plate thickness	5 mm	Plate thickness	5 mm	Plate thickness	5 mm
Horizontal pressure	2000 N/m^2	Stiffener height	0.1 m	Stiffener height	0.1 m
Vertical pressure	5000 N/m^2	Pressure	10 kN/m^2	Pressure	10 kN/m^2

Table 3 Algorithm parameters

Parameter	Value
Maximum initial displacements	1e-5
Maximum initial update coef.	1e-2
Update coef. increment rate	1.1
Update coef. decrement rate	0.9
Power transform coef. (λ)	0.15

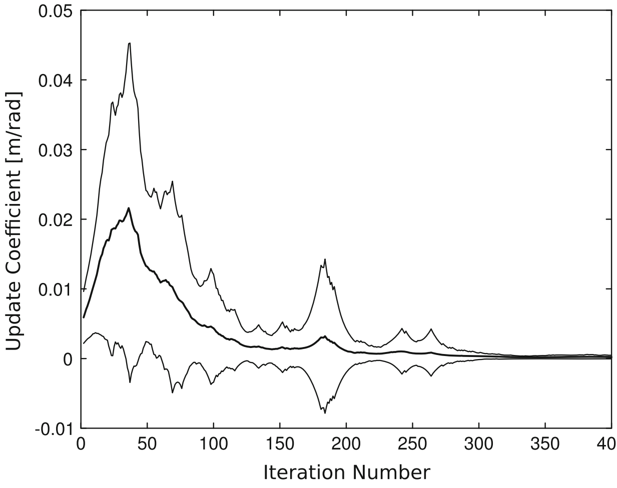

Fig. 8 Update coefficient progress for 1st example

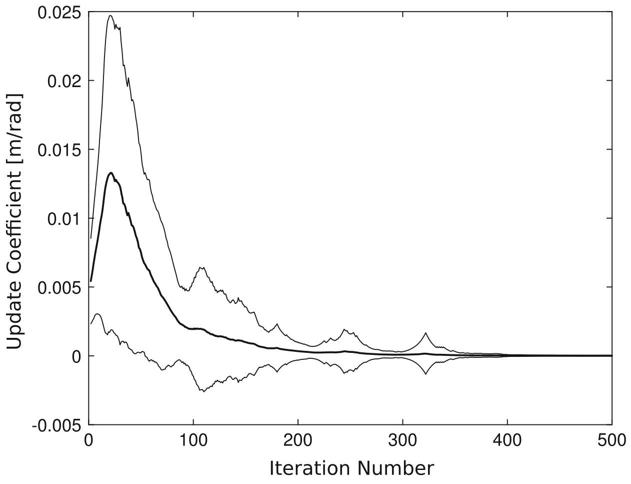

Fig. 9 Update coefficient progress for 2nd example

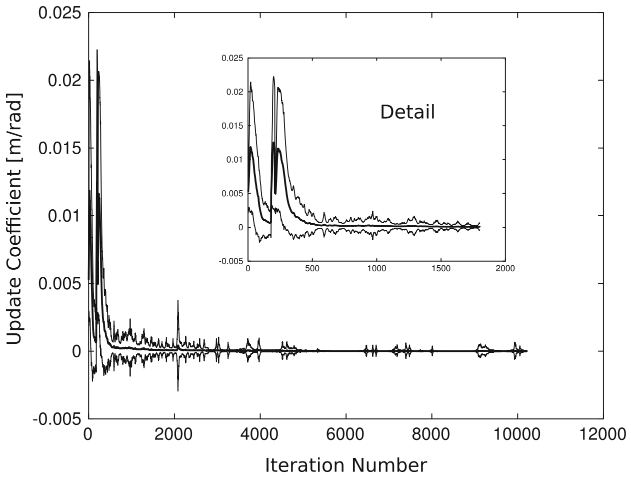

Fig. 10 Update coefficient progress for 3rd example

nodes are far from the equilibrium positions. Update coefficients form a peak, which means the nodes crossed their respective equilibrium configurations. After the peak, the coefficients start to decrease to further locate the precise displacement configuration.

Although a local oscillatory behaviour is observed, the error in estimation for the displacements decreases with increasing the number of iterations globally (Figs. 11, 12, 13). Error in estimation converges within 2 % of average displacement in 5385 iterations for the 3-panel example, 292 iterations for 9-panel and 9618 iterations for the 25-panel example. So there seems to be no direct correlation with the scale of the system. Relatively slow convergence is observed in the first and third examples and load eccentricity and asymmetrical geometry of the structure, which is specific for these examples, may be the reason.

The final plots (Figs. 14, 15, 16) represent the residual force/moment. Again an oscillating pattern can be seen

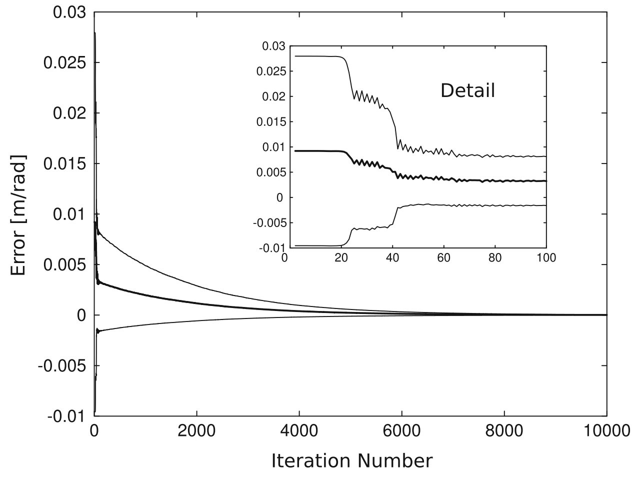

Fig. 11 Error in displacements for 1st example

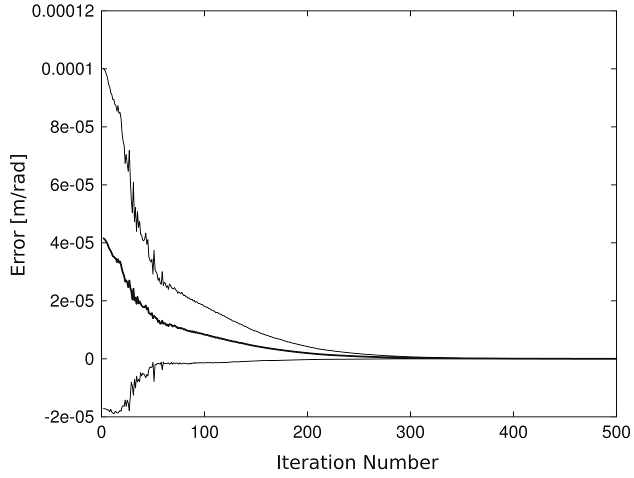

Fig. 12 Error in displacements for 2nd example

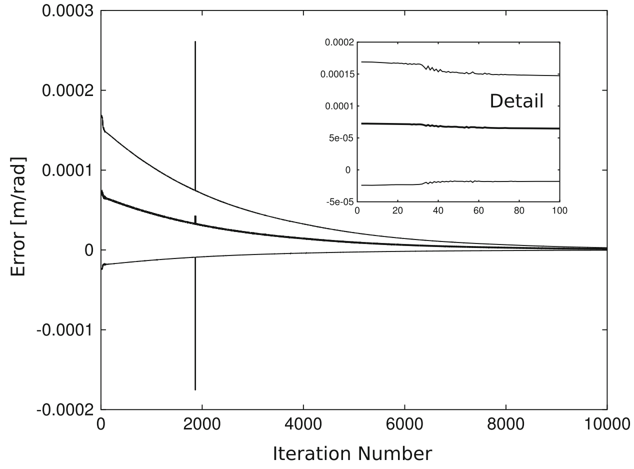

Fig. 13 Error in displacements for 3rd example

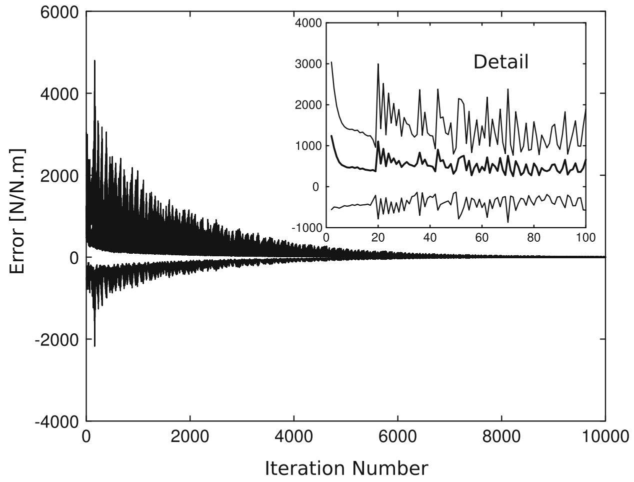

Fig. 14 Residual force/moment for 1st example

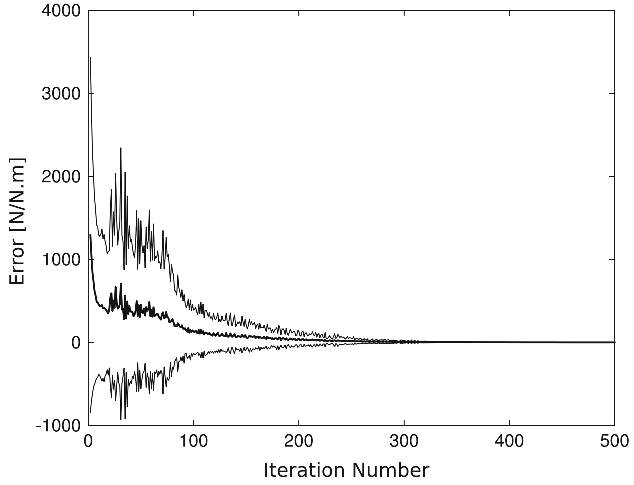

Fig. 15 Residual force/moment for 2nd example

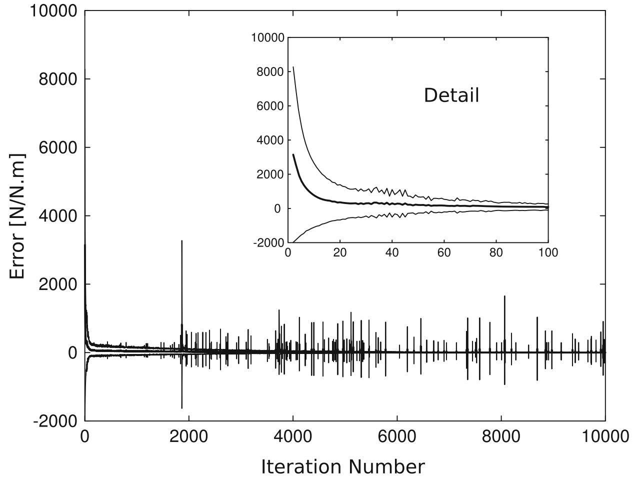

Fig. 16 Residual force/moment for 3rd example

here and overall behaviour is still convergent. Spikes can be seen in especially the third example. These situations may arise from time to time, but the algorithm is robust for these discrepancies and it can make the appropriate correction itself.

5 Discussions and future work

In three example cases, the decomposition algorithm showed convergent behaviour. Furthermore, there was no correlation between scale of the test cases and the number of iterations required to limit the average error below a certain limit.

This study does not show directly whether the new approach performs better when compared against FEM, but current findings point to the potential improvement over conventional techniques. Still it is a future challenge to test the method with additional case studies of even larger scales and complexity.

Although one of the biggest elements of the problem is solved, there are many tasks to be accomplished to use this design approach in a more realistic setting. First of all, the algorithm is in a rough state. Improvements should be made to decrease the number of iterations required for search to converge.

Because only a finite element test bench was used to show that the algorithm works, an actual surrogate model construction is needed to be done. An effective parametrization of the building block should be arranged so that local problem is represented sufficiently without being too complicated to be built.

Another immediate need arises when the global structure is generated automatically to be used in an optimization. The generated structure has to obey the constraints, cover the maximum range of design space without being infeasible.

Last but not least, because the decomposition method presented here is only suitable for static problems, a solution for dynamic problems is still not available.

References

1. Alba E, Tomassini M (2002) Parallelism and evolutionary algorithms. Evolut Comput IEEE Trans 6(5):443–462
2. Bandler J, Biernacki R, Chen SH, Grobelny P, Hemmers R (1994) Space mapping technique for electromagnetic optimization. Microw Theory Tech IEEE Trans 42(12):2536–2544
3. Bellman R (1961) Adaptive control processes: a guided tour. Princeton University Press, Princeton
4. Bichon BJ, Eldred MS, Mahadevan S, McFarland JM (2013) Efficient global surrogate modeling for reliability-based design optimization. J Mech Des 135(1):011009
5. Box GEP, Cox DR (1964) An analysis of transformations. J R Stat Soc Ser B Methodol 26(2):211–252
6. Breitkopf P, Naceur H, Rassineux A, Villon P (2005) Moving least squares response surface approximation: formulation and metal forming applications. Comput Struct 83(17):1411–1428
7. Chan TF, Smith BF, Zou J (1996) Overlapping Schwarz methods on unstructured meshes using non-matching coarse grids. Numer Math 73(2):149–167
8. Chan TF, Van-der Vorst HA (1997) Approximate and incomplete factorizations. Springer, Heidelberg
9. Cosgrove J, Diaz J, Griewank A (1992) Approximate inverse preconditionings for sparse linear systems. Int J Comput Math 44(1–4):91–110
10. Farhat C, Roux FX (1991) A method of finite element tearing and interconnecting and its parallel solution algorithm. Int J Numer Methods Eng 32(6):1205–1227
11. Hansen LU, Horst P (2008) Multilevel optimization in aircraft structural design evaluation. Comput Struct 86(1):104–118
12. Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. J Glob Optim 13(4):455–492
13. Mandel J, Brezina M (1993) Balancing domain decomposition: theory and performance in two and three dimensions. University of Colorado, Denver
14. Pronzato L, Muller WG (2012) Design of computer experiments: space filling and beyond. Stat Comput 22(3):681–701
15. Przemieniecki JS (1963) Matrix structural analysis of substructures. AIAA J 1(1):138–147
16. Queipo NV, Haftka RT, Shyy W, Goel T, Vaidyanathan R, Kevin Tucker P (2005) Surrogate-based analysis and optimization. Prog Aerosp Sci 41(1):1–28
17. Sacks J, Welch WJ, Mitchell TJ, Wynn HP (1989) Design and analysis of computer experiments. Stat Sci 4(4):409–423
18. Simpson TW, Booker AJ, Ghosh D, Giunta AA, Koch PN, Yang RJ (2004) Approximation methods in multidisciplinary analysis and optimization: a panel discussion. Struct Multidiscip Optim 27(5):302–313
19. Stephens R (1997) A survey of stream processing. Acta Inf 34(7):491–541
20. Wald A (1945) Sequential tests of statistical hypotheses. Ann Math Stat 16(2):117–186
21. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evolut Comput 1(1):67–82