Advances in Engineering Software 67 (2014) 186–197

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

The use of partially converged simulations in building surrogate models

N. Courrier *, P.-A. Boucard, B. Soulier

LMT-Cachan (ENS Cachan/CNRS/UPMC/PRES UniverSud Paris), 61 avenue du Président Wilson, F-94230 Cachan, France

ARTICLE INFO

Article history:
Received 1 March 2013
Received in revised form 19 July 2013
Accepted 22 September 2013
Available online 18 October 2013

Keywords:
Partially converged data
Evofusion
Metamodel
Global optimization
LATIN method

ABSTRACT

The main objective of this paper is to propose an optimization strategy which uses partially converged data to minimize the computational effort associated with an optimization procedure. The framework of this work is the optimization of assemblies involving contact and friction.

Several tools have been developed in order to use a surrogate model as an alternative to the actual mechanical model. Then, the global optimization can be carried out using this surrogate model, which is much less expensive. This approach has two drawbacks: the CPU time required to generate the surrogate model and the inaccuracy of this model.

In order to alleviate these drawbacks, we propose to minimize the CPU time by using partially converged data and then to apply a correction strategy. Two methods are tested in this paper. The first one consists in updating a partially converged metamodel using global enrichment. The second one consists in seeking the global minimum using the weighted expected improvement. One can achieve a time saving of about 10 when seeking the global minimum.

1. Introduction

The development of numerical tools and mechanical models over the past few years has helped improve the design of complex systems. The ability of modeling and simulation softwares to handle nonlinear problems, which are sources of numerical difficulties, has increased continuously. Aside from resolution difficulties, the computation costs associated with systematic calls to the simulation routines make direct structural optimization unthinkable in an industrial context. In this work, we focus on parameterized assemblies. Let $\mathcal{D}$ be the design space and $\mathbf{x}$ a vector of dimension p which contains the p design parameters: $\mathbf{x} = [x_1 \dots x_p]$.

As recalled in [1], the most efficient optimization methods rely on multilevel optimization concepts, which can be divided into three categories: parallel model optimization based on domain decomposition methods [2,3], multilevel parameter optimization, which consists in replacing an optimization problem by several subproblems, each with a reduced set of parameters [4–7], and multilevel model optimization, which introduces several modeling levels [8–10].

In the context of multilevel optimization, the use of a tool called "metamodel" has become predominant. A metamodel (or surrogate model [11]) is an approximate model of the objective function $F(\mathbf{x})$. This reduced model, denoted $\widehat{F}(\mathbf{x})$, defined over the whole domain being studied, requires one to simulate a large number of points (which can be very expensive), then associate a mathematical interpolation or regression model with these points [11,12].

In more general case, surrogate models can be divided according to [13] in three categories:

- Response surfaces: a response surface is a functional mapping of several input parameters to a single output feature. It could be of polynomial form where regression coefficients are determined by the method of least squares. Typically model properties of interest are those that characterize model fit quality, contribution of an individual variable to total model variance, model resolution, etc. [11] provides a taxonomy of the different processes used to create a surface.
- Reduced models derived from the Proper Orthogonal Decomposition (POD) or Proper Generalized Decomposition (PGD). Readers can find details in [14,15]. More recently PGD was used on parametric models and this approach is initiated the use of the PGD to deal with parametric problems such as parametric optimization problems, inverse identification problems or real time simulation. Some examples of the parametric modelling can be found in [16,17]. PGD was also carried out in the context of Multiparametric Strategy [18].
- Hierarchical models, also called multifidelity, variable-fidelity or variable-complexity. In the context of two levels of fidelity, corrections between the two models can be lead, for example [19] uses a kriging model to correct the low-fidelity model on the high-fideliy model. This correction is called scaling model or correction response surface [20,21].

* Corresponding author. Tel.: +33 147405335.
E-mail addresses: courrier@lmt.ens-cachan.fr (N. Courrier), boucard@lmt.ens-cachan.fr (P.-A. Boucard), soulier@lmt.ens-cachan.fr (B. Soulier).

0965-9978/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.advengsoft.2013.09.008

In this general case, the main stumbling block is the computation time of the points of the design domain. In order to overcome this problem, we undertake to develop strategies, inspired by previous works in fluid mechanics, based on partially converged data.

This paper focuses on the construction of metamodels based on partially converged data. Two strategies have been implemented: one to localize the zone of interest (the zone in which the optimum is to be found), and the other to build the complete metamodel with a given level of accuracy.

Classically, optimization techniques begin with a metamodel-based optimization (which is much less time-consuming) in order to locate the zone of interest. Then, the mathematical model associated with the metamodel plays an important role. There are several possible construction approaches, such as kriging [22,12], co-kriging [23–25], the RBF method, etc. In this paper, we use a universal kriging built using the DACE toolbox [26,27]. Kriging was introduced in 1951 by D.G. Krige and later developed, among others, by Matheron [12], who laid out its mathematical foundations.

The kriging technique enables one to determine an approximate function using statistical reasoning. It consists in considering the objective function in the approximate form:

$$\widehat{F}(\mathbf{x}) = \mu(\mathbf{x}) + z(\mathbf{x}) \tag{1}$$

where $\mu(\mathbf{x})$ is a regression function of the known data, and $z(\mathbf{x})$ is a stochastic process which represents the prediction error of the regression model. This function z has zero expectancy and constant variance:

$$\begin{aligned} &\mathrm{E}(z(\mathbf{x})) = 0 \\ &\mathrm{E}(z(\mathbf{x})^2) = \sigma^2_{krige} \end{aligned} \tag{2}$$

In the case of universal kriging, $\mu(\mathbf{x})$ is an approximation by regression over a basis of functions which, usually, are polynomials. It is also important to define a correlation function, which is required for the evaluation of $z(\mathbf{x})$. We use a generalized exponential correlation function similar to that defined in the following equation:

$$\mathrm{Corr}[(\mathbf{x}_i), \mathbf{x}_j)] = \exp\left(-\sum_{l=1}^{p} \theta_l |\mathbf{x}_{il} - \mathbf{x}_{jl}|^{t_l}\right) \tag{3}$$

The various parameters are obtained through the calculation of a maximum likelihood. θ_l is a parameter which is associated with the correlation drop-off rate in the lth dimension, and t is a smoothing parameter of the metamodel.

After solving a problem of minimization of the mean square error (MSE), one finds, for any $\mathbf{x}$, both $\widehat{F}(\mathbf{x})$ and an estimate of the associated MSE, denoted $\sigma(\mathbf{x})^2$ and defined by:

$$\sigma(\mathbf{x})^2 = \mathrm{E}\left[\left(\widehat{F}(\mathbf{x}) - F(\mathbf{x})\right)^2\right] \tag{4}$$

Thus, the optimization, being carried out using this metamodel, is relatively inexpensive. However, such a metamodel, besides being costly to generate, can be inaccurate, which leads to problems in terms of its resolution. In order to alleviate this drawback, there exist metamodel enrichment strategies [28,29], especially the "expected improvement" method [30], which enables one to target the zones of interest more accurately (and, thus, accelerate and improve the optimization). The expected improvement is a probabilistic criterion which defines the improvement which can be achieved in terms of seeking an optimum. Then one carries out the enrichment at a point to the highest expected improvement in order to seek the minimum of the objective function (see Section 3.3).

Strategies based on the calculation of partially converged data have been implemented in order to accelerate the optimization even further. These strategies, which do not necessarily rely on the metamodel tool, can be divided into three main categories.

(1) The first category uses different levels of fidelity of the objective function $F(\mathbf{x})$ ([31,19,32]). The techniques developed in the first two references use two levels of fidelity of the objective function whose optimum is being sought. Local optimizations are carried out on the low-fidelity model (LF) using a confidence-region-based algorithm. Then, the high-fidelity model (HF) is evaluated for each resulting optimum in order to guarantee the convergence of the method. This type of algorithm is illustrated in Fig. 1, where f_{hf} and f_{lf} denote respectively the HF and LF representations of the objective function. We can use a correction between the two models to be more accurate.

(2) Another category of methods using metamodels is comparable to the processing of noisy data [33–35]. Indeed, to a certain extent, partially converged data can be viewed as noisy data. One possible approach consists in transforming the universal kriging method into a modified regressive kriging ([35]). One can also use enrichment criteria which are different from those which are generally applied. [33] proposed a modification of the expected improvement criterion [30]. However, we did not use such methods in our work because the results obtained with noisy data and with partially converged data are not comparable.

(3) The last category of methods, which includes our work, makes full use of partially converged data (*i.e.* points in the design space) [36–39]. The earliest works in that group ([38]) were based on a progressive algorithm, *i.e.* a local optimization method based on a sequential refinement of the mesh. The main idea is to use an adjoint problem whose solution is the gradient of the objective function, leading to the directions of descent. Two solvers are used: one for the direct problem and the other for the adjoint problem. Partially converged solutions of these problems are sought. Then, the optimization variables are updated along the direction of the gradient and the process is repeated until convergence, *i.e.* until the estimated gradient of the function reaches a certain threshold. Then, the following step consists in refining the mesh and repeating the various operations down to the finest mesh level (see Algorithm 1).

Algorithm 1. The progressive algorithm

1: Set up the design variables
2: Begin the resolution using a coarse mesh
3: Proceed with the direct resolution of the problem
4: Solve the adjoint problem
5: Evaluate the function of interest F and its gradient ∇F
6: Update the design variables according to the relation

$$x_j^{l+1} = x_j^l - a_j \frac{\partial F}{\partial x_j} \tag{5}$$

7: Repeat Steps 3 to 6 until the gradient of the objective function has decreased sufficiently
8: Refine the mesh by doubling the number of intervals in each direction, and interpolate the previous solution over this refined mesh
9: Repeat 3 to 6
10: Repeat 8 and 9 down to the most refined mesh
11: Repeat 3 to 6 until $\frac{\partial F}{\partial x_j} \simeq 0$