11th World Congress on Computational Mechanics (WCCM XI)
5th European Conference on Computational Mechanics (ECCM V)
6th European Conference on Computational Fluid Dynamics (ECFD VI)
July 20 - 25, 2014, Barcelona, Spain

MULTIDISCIPLINARY OPTIMIZATION OF TURBOMACHINERY COMPONENTS USING DIFFERENTIAL EVOLUTION

T. Verstraete [1]

[1] Von Karman Institute
Waterloosesteenweg 72,1060 Sint-Genesius-Rode, Belgium
Tom.vertsraete@vki.ac.be www.vki.ac.be

Key Words: *multidisciplinary, optimization, turbomachinery*

A current trend in turbomachinery component design is to rely more and more on predictions from numerical tools in the design process. Where in the past components were designed by using rules of thumb and simple correlations coming from theoretical considerations and experimental experience, nowadays computational tools such as Computational Fluid Dynamics (CFD) and Computational Structural Mechanics (CSM) are well integrated into the design process.
In order to reduce the time-to-market while improving the product quality, optimization techniques are currently very attractive. They tend to replace the time-consuming trial and error design method by an automated design procedure. Moreover, due to the multidisciplinary nature of the design of turbomachinery components, a trend towards concurrent design in which all disciplines are treated simultaneously rather than sequentially is emerging. This allows avoiding time consuming iterations between different disciplines (often represented by different departments in a company), leading to sub-optimal designs.
This paper presents an overview of the lecture given at the Von Karman Institute Lecture Series on Optimization Methods and Tools for Multi physics Design in Aeronautics and Turbomachinery. It presents the use of a multidisciplinary design optimization approach for different turbomachinery components such as axial/radial compressors/turbines using CFD and CSM tools.

Evolutionary Algorithms (EA) have been developed in the late sixties by J. Holland [1] and I. Rechenberg [2]. They are based on Darwinian evolution, whereby populations of individuals evolve over a search space and adapt to the environment by the use of different mechanisms such as mutation, crossover and selection. Individuals with a higher fitness have more chance to survive and/or get reproduced. When applied to design optimization problems, EAs have certain advantages above gradient based methods. They do not require the objective function to be continuous and are noise tolerant. In the presence of local minima, they are capable of finding global optima and avoid to get trapped in a local minimum. Moreover, these methods can efficiently use distributed and parallel computing resources since multiple evaluations can be performed independently. The evaluation itself does not necessary needs to be made parallel. Disadvantages of EAs are mainly related to the large number of function evaluations needed.
Differential Evolution (DE) is a relatively new evolutionary method developed by Price and Storn [3]. It is easily programmable, does only require a few user defined parameters and performs well for a wide variety of these parameters. A determination of optimal user defined

parameters is very often unnecessary. Differential evolution, like all EAs, is population based and requires at each iteration the evaluation of an entire population of designs. It allows an easy extension to handle multiple objectives simultaneously. The nomenclature resembles the one of evolutionary processes. A design vector is called an individual; the collection of individuals at one given iteration is called a population, and the evolution of a population happens within generations, i.e. the children of the current population form the next generation.

The major drawback of evolutionary algorithms such as DE is the total number of evaluations needed. In general, more than thousand evaluations are commonly needed, and depending on the complexity of the optimization problem (both number of parameters and complexity of the objective function), this number can drastically increase.

One way of reducing the unrealistic number of evaluations can be obtained by replacing the expensive evaluations (involving FVM and/or FEM) by a computationally cheaper method such as a metamodel. The metamodel performs the same task as the high fidelity model, but at a very low computational cost. However, it is less accurate, especially for an evaluation far away from the already analyzed points in the design space.

The implementation of the metamodel into the optimization system depends on how the system deals with the inaccuracy of the model. The technique discussed uses the metamodel as an evaluation tool during the entire evolutionary process [4]. After several generations the evolution is stopped and the best individual is analyzed by the expensive analysis tool. This technique is referred to as the off-line trained metamodel. The difference between the predicted value of the metamodel and the high fidelity tool is a direct measure for the accuracy of the metamodel. Usually at the start this difference is rather large. The newly evaluated individual is added to the database used for the interpolation and the metamodel will be more accurate in the region where previously the evolutionary algorithm was predicting a minimum. This feedback is the most essential part of the algorithm as it makes the system self-learning. It mimics the human designer which learns from his mistakes on previous designs.

REFERENCES

[1] J. H. Holland. Adaption in Natural and Artificial Systems. University of Michigan Press, 1975.

[2] I. Rechenberg. Evolutionsstrategie | Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Fommann-Holzboog, Stuttgart, 1973.

[3] K. Price and N. Storn. Di_erential Evolution. Dr. Dobb's Journal, pages 18{24, April 1997.

[4] S. Pierret. Designing Turbomachinery Blades by means of the Function Approximation Concept based on Artificial Neural Network, Genetic Algorithm, and the Navier-Stokes Equations. PhD Thesis, Faculte Polytechnique de Mons, December 1999.