Proceedings of ASME Turbo Expo 2011
GT2011
June 6-10, 2011, Vancouver, British Columbia, Canada

GT2011-46541

OPTIMIZATION OF A U-BEND FOR MINIMAL PRESSURE LOSS IN INTERNAL COOLING CHANNELS – PART I: NUMERICAL METHOD

Tom Verstraete, Filippo Coletti, Jérémy Bulle, Timothée Vanderwielen, Tony Arts

von Karman Institute for Fluid Dynamics
Turbomachinery and Propulsion Department
Chaussée de Waterloo 72
1640 Rhode-Saint-Genèse, Belgium
Email: tom.verstraete@vki.ac.be, coletti@vki.ac.be

ABSTRACT

This two-parts paper addresses the design of a U-bend for serpentine internal cooling channels optimized for minimal pressure loss. The total pressure loss for the flow in a U-bend is a critical design parameter as it augments the pressure required at the inlet of the cooling system, resulting in a lower global efficiency. In this first part of the paper the design methodology of the cooling channel is presented. The minimization of the total pressure loss is achieved by means of a numerical optimization method that uses a metamodel assisted differential evolution algorithm in combination with an incompressible Navier-Stokes solver. The profiles of the internal and external side of the bend are parameterized using piece-wise Bezier curves. This allows for a wide variety of shapes, respecting the manufacturability constraints of the design. The pressure loss is computed by the Navier-Stokes solver, which is based on a two-equation turbulence model and is available from the open source software OpenFOAM. The numerical method predicts an improvement of 36% in total pressure drop with respect to a circular U-bend, mainly due to the reduction of the separated flow region along the internal side of the bend. The resulting design is subjected to experimental validation, presented in Part II of the paper.

INTRODUCTION

The thermal efficiency of gas turbines increases dramatically with the maximum temperature of the cycle. As a result, state-of-the-art gas turbines are designed to operate at turbine inlet temperatures that approach 2000 K. Since the materials commonly employed for the turbine components cannot withstand temperatures above 1350 K, effective cooling must be applied along the hot-gas-path in order to guarantee a safe functioning. In most cases the coolant is air bled from the high pressure compressor, which bypasses the combustor and enters the blade through its root, circulating through serpentine internal passages. The flow in internal cooling channels is fully turbulent and generally free of compressibility effects. The geometrical configurations are complex and the velocity field is highly three-dimensional. More than 20% of the discharge air from the compressor is used to cool the high pressure turbine, leading to a severe penalty on the thermodynamic efficiency. Therefore, an effective design must be able to maintain the metal temperature below acceptable limits with minimal coolant mass flow rates and pressure drop penalties. Reviews of mechanisms and performances of turbine blade cooling techniques were presented, among others, by Han et al. [1] and Weigand et al. [2].

Among the salient features of the cooling passages, the U-bends that connect consecutive passages play a key role, as they represent regions of strong pressure loss, especially for small radius ratio (mean bend radius/duct hydraulic diameter): in this case the bend region can be responsible for up to 25% of the pressure loss in the entire multi-pass cooling system. Consequently this flow configuration has received profound attention from the scientific and technical community. Numerous experiments investigating the turbulent flow in 180° bends have been conducted, both for circular and sharp turns. The contributions of Humphrey et al. [3], Chang et al. [4], Monson and Seegmiller [5] and Cheah et al. [6] using laser Doppler velocimetry (LDV) concern the first type of U-bends. The velocity field in sharp corner bends was investigated by Liou et al. [7] by LDV, Son et al. [8] by two-dimensional particle image velocimetry (PIV) and Schabacker et al. [9] by stereoscopic PIV. All studies highlighted the presence of secondary flows driven by the imbalance between the centrifugal forces and the radial pressure gradient. For sufficiently high curvature (i.e. low radius ratio), separation occurs along the inner wall in the second half of the bend.

U-bend geometries make an excellent test case for turbulence models, as the effects of the streamlines curvature and the associated secondary flows are typically challenging to reproduce in numerical simulations. The broad trends can be captured by two-equation eddy-viscosity models, provided that the boundary layer is resolved without recurring to wall functions, as discussed by Iacovides and Launder [10]. Recently, Lucci et al. [11] and Schueler et al. [12], using eddy-viscosity models, found overall agreement with experiments. However two-equation models cannot predict the effect of streamline curvature on the turbulence structure due to their inability to account for turbulence anisotropy. Calculations based on second moment closure were shown to provide more accurate predictions, e.g. in the two-pass channels studied by Bonhof et al. [13] and Chen et al [14]. Nevertheless, due to their reduced computational cost, two-equation models are still the standard tool employed in the design process in industrial applications.

The high pressure penalty imposed by the U-bend has fostered the interest towards strategies to improve the aerodynamic performance, especially in sharp turn configurations. Metzger et al. [15] varied the width of the passages, the corner radius and the clearance height, finding that the latter parameter had a substantial impact, with the pressure drop increasing for smaller clearances. The influence of the divider wall thickness was explored by Liou and Chen [16]. They found that a thicker wall reduces the turbulence level and shortens the reattachment length of the recirculating cell in the downstream half of the bend. Bonhof et al. [17] showed that inserting turning vanes alleviates the pressure loss. Schueler et al. [18], while confirming that properly, designed vanes which significantly reduce the pressure drop, also underlined that poorly designed vanes can actually deteriorate the aerodynamic performance.

All the above-mentioned studies concerned with the minimization of the U-bend pressure drop follow a classic trial-and-error approach: several configurations are generated varying a number of geometrical parameters, performances are compared and global trends are evaluated. However, given the large number of parameters, this type of design process remains extremely time-consuming. Moreover, as many of the parameters are strongly coupled, the relations between them and their effects are difficult to asses. In order to ease and speed up the process, so called optimization methods can be applied. Most of these techniques exploit natural principles to obtain effective solutions, while minimizing the intervention of the human designer. A recent example is the study of Zehner et al. [19], who optimized the divider wall of a sharp turn U-bend. They used the ice-formation technique to generate a starting profile of minimum energy dissipation, and further improved the performance applying an evolutionary algorithm. Namgoong et al. [20] used Design of Experiment and surrogate design space model for similar purposes.

This two-part paper addresses the design of a smooth U-bend of radius ratio 0.76, optimized for minimal pressure loss. In this first part of the paper the design methodology is presented. The minimization of the total pressure loss is achieved by means of a numerical optimization method that uses a metamodel-assisted differential evolution algorithm in combination with an incompressible Navier-Stokes solver. The profiles of the internal and external side of the bend are parameterized using piece-wise Bezier curves. The pressure loss is computed by the Navier-Stokes solver, which is based on a two-equation turbulence model. The resulting design is subjected to experimental validation, presented in Part II of the paper [21], where the results of the aerodynamic optimization are discussed.

In the present contribution only the aerodynamic performance of the cooling channel geometry is addressed. Including the heat transfer performance would imply to tackle a multi-objective optimization problem, which is beyond the scope of the present work. It is arguable that the accuracy achievable by nowadays computational tools in terms of heat transfer levels is sufficient for the purpose of the present investigation. Higher fidelity simulations are not an option in optimization due to the computational cost. On the other hand RANS solvers prove to yield reliable predictions of the pressure drop. The above considerations support the present choice of focusing first on the aerodynamic performance.

NOMENCLATURE

Latin

ANN	Artificial neural network
C	user defined constant
DE	Differential Evolution
DOE	Design of Experiments
E	error
EA	Evolutionary Algorithm
F	user defined constant
FVM	Finite Volume Method
MSE	Mean Square Error
RSM	Response Surface Models
P	pressure
S	surface area
TF	activation function
a,b,c	design vectors
f	objective function
r	random variable
t	generation index
v	velocity
w	neuron weight
x	design vector
y	trial vector
z	candidate vector

Greek

ε	dissipation rate of turbulent kinetic energy
ρ	density
ω	connection weight

OPTIMIZATION METHOD

The optimization method used is the result of more than one decade of research conducted at the von Karman Institute [22, 23, 24]. The system (Fig. 1) makes use of a Differential Evolution algorithm (DE), a metamodel, a database, and a Finite Volume Method (FVM) CFD solver. The basic idea behind this method is a two-level optimization. A first one uses a rapid but less accurate analysis method (the metamodel) to evaluate the large number of geometries generated by the DE. The optimum geometry, according to the metamodel predictions, is then analyzed by the more accurate but much more computationally expensive FVM calculations to verify the accuracy of the metamodel predictions. The outcome of such an optimization cycle is added to the database. It is expected that, after a new training on the extended database, the metamodel will be more accurate as it is based on more information and the outcome of the next DE optimization will be closer to the real one. The optimization cycle is repeated until the Navier-Stokes results confirm the accuracy of the metamodel predictions.

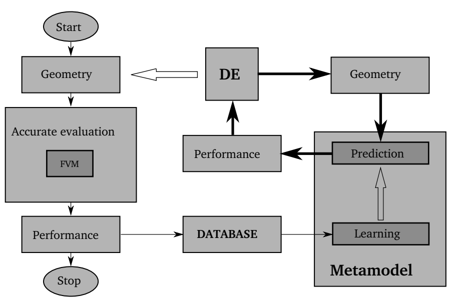

Figure 1 Flow chart of the optimization algorithm.

Geometry

The U-bend under investigation is typical of internal cooling channels. The same scaled version for PIV-measurements in the lab has been considered for the numerical optimization. The baseline geometry is show in Fig. 2. It consists of a circular U-bend with radius ratio of 0.76, a hydraulic diameter of 0.075 meter and an aspect ratio of 1. The Reynolds number is 40.000 and the Mach number of 0.05 allows using an incompressible assumption. The shape of the inner and outer curve is allowed to be changed but needs to remain inside the bounding box shown in the figure, which restricts the height and width of possible changes to account for structural limits. The distance between both cooling channels is not subject to optimization, as well as the hydraulic diameter.

Parameterization

The parameterization of both inner and outer curve is shown in Figs. 3 and 4. Both curves are composed of 4 Bezier curves. The 180 degree turn is split into 2 Bezier curves, while the curves connecting the inlet with the U-bend and the U-bend with the outlet are defined by single Bezier curves. Each Bezier curve is a third order curve defined by a polynomial of 4 control points. By changing the coordinates of these control points the shape of the curve will change. This allows controlling the shape of the U-bend by several well-chosen parameters.

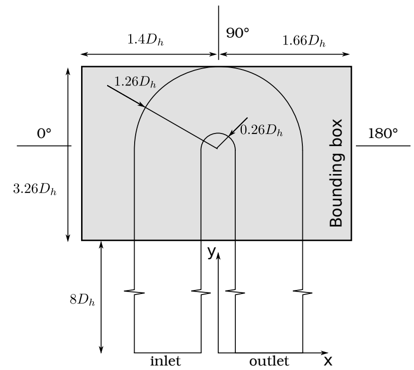

Figure 2: Baseline geometry, definition of area in which the shape is allowed to change.

The 4 curves defining the outer curve of the U-bend are parameterized by a total of 12 degrees of freedom, shown by arrows in Fig. 3. Several control points are only allowed to change in one direction while their other direction is controlled by the position of a neighboring control point to guarantee G1 continuity. The 3rd control point of curve 2 is for instance only allowed to move in the horizontal direction while it follows the vertical movement of the 4th control point of that curve such that the curve remains horizontal in its endpoint. In the last control point of the first curve 3 degrees of freedom are specified: the horizontal and vertical movement of the control point and the curvature in that location. The curvature defines the vertical distance between the last and before last control point of curve 1 and the distance between the first and second control point of curve 2. The use of a curvature parameter guarantees a G2 continuity at the junction between curve 1 and 2, which is considered necessary after some preliminary studies. However, at the symmetrical control point between curve 3 and 4, no curvature specification is needed. As a result, the before last control point of curve 3 and the second control point of curve 4 have a vertical degree of freedom.

The parameterization of the inner curve is similar to the one of the outer curve, although for some control points the definition is based on the position of the control points of the outer curve. The last control point of the first curve is for instance positioned in the horizontal direction by a distance D1 from the same control point of the outer curve. The parameters D2 and D3 define their respective control points in a similar way. This parameterization introduces parameters closely related to the flow physics, compared to a parameterization where the x- or y-coordinates would have been specified. This allows a more linear and direct relation between parameters and objective, as the acceleration of the flow is the result of one single parameter and not the difference between two parameters defined at inner and outer curve. This makes the optimization problem well-posed.

The number of parameters defining the inner curve is 14. The total number of parameters for the entire U-bend is 26, which is a trade-off between a large freedom in shape and the need for an effective optimization. For each parameter suitable ranges are defined.

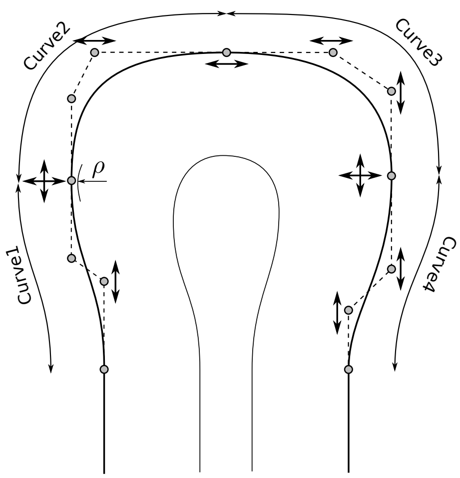

Figure 3: Parameterization of the outer curve.

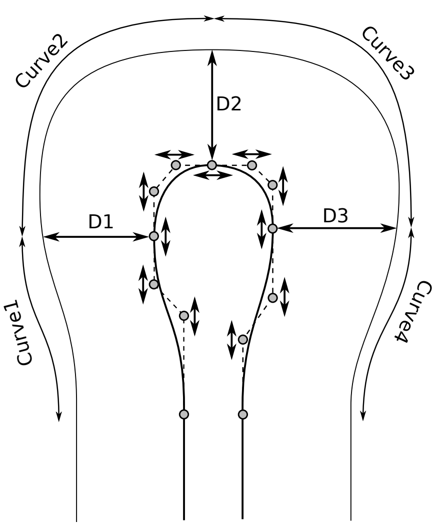

Figure 4: Parameterization of the inner curve.

Grid generation

A structured gird with 342x50x50 (855,000) cells has been used. The mesh is generated by an automated Gambit [25] script and allows for local refinement in regions of high curvature. In Fig. 5 a typical 2D view of the grid is shown. The boundary layer has been refined in accordance to the necessity of the Launder-Sharma low-Reynolds $k-\varepsilon$ turbulence model. The maximum y+ value does not exceed 2.2.

The *k-ε* model "is arguably the simplest complete turbulence model" (Pope [26]), is implemented in most commercial software and is one of the most broadly employed at industrial level. Its performance is reasonably satisfactory in shear flows with small effects of streamwise pressure gradients and streamline curvatures, but far from these assumptions, it can fail badly. However it has been selected for the present application due to its large diffusion: given that the proposed methodology is apt for industrial problems, it was the intention of the authors to demonstrate its potential in conditions that are representative of real-life design practice.

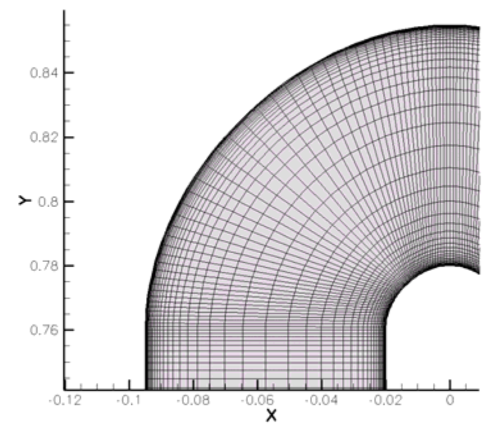

Figure 5: Zoom on the grid in the U-bend.

Performance evaluation

The simpleFoam solver from OpenFoam [27] is used to evaluate the incompressible Navier-Stokes equations. At the inlet a fully developed velocity profile is imposed, together with values of k and ε for the turbulence model. Both are computed based on a turbulence intensity of 5% measured in the lab. At the outlet the static pressure is imposed. A convergence criteria of 5.10^{-6} for the residuals is imposed. The computations are run in parallel on 5 cores, requiring an average of 2 hours per calculation.

The U-bend optimization is driven by the minimization of the pressure drop introduced by the U-bend. The objective function is formulated as:

$$\min\ f(\vec{x}) = P_{total}^{inlet} - P_{total}^{outlet} \tag{1}$$

where $\vec{x}$ is the design vector containing the 26 parameters describing the geometry (see Figs. 2 and 3). The total pressure is computed as the mass flow averaged quantity at the inlet respectively outlet of the domain, positioned 8 hydraulic diameters away from the U-bend.

$$P_{total} = \frac{\int_S p_{total} \cdot \rho v \cdot dS}{\int_S \rho v \cdot dS} \tag{2}$$

Single-objective Differential Evolution

Evolutionary Algorithms (EA) have been developed in the late sixties by J. Holland [28] and I. Rechenberg [29]. 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 need 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 [30]. 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. The nomenclature resembles the one of evolutionary processes. A design vector $\vec{x}$ 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 purpose of the algorithm is to find the individual $\vec{x}$ which minimizes an objective function $f(x)$. To describe one version of the single-objective DE [30], the *t*-th generation containing T individuals is considered. Each individual $\vec{x}_t$ contains *n* parameters.

$$\vec{x}_t = (x_1, x_2, ..., x_n) \tag{3}$$

To evolve the parameter vector $\vec{x}_t$, three other parameter vectors $\vec{a}_t$, $\vec{b}_t$ and $\vec{c}_t$ are randomly picked such that $\vec{a}_t \neq \vec{b}_t \neq \vec{c}_t \neq \vec{x}_t$. A trial vector $\vec{y}_t$ is defined as

$$y_i = a_i + F \cdot (b_i - c_i) \quad , \qquad i = 1..n \tag{4}$$

where F is a user defined constant ($F \in]0,2[$) which controls the amplification of the differential variation $(b_i - c_i)$. This procedure is usually called the mutation. The candidate vector $\vec{z}_t$ is obtained by a recombination operator involving the vectors $\vec{x}_t$ and $\vec{y}_t$, and is defined as

$$z_i = \begin{cases} y_i & if \quad r_i \leq C \\ x_i & if \quad r_i > C \end{cases} \quad i = 1..n \tag{5}$$

where r_i is a uniformly distributed random variable ($0 \leq r_i < 1$) and C is a user defined constant ($C \in]0,1[$). This procedure is usually called the crossover, in analogy with Genetic Algorithms (GA).

The final step in the evolution of $\vec{x}_t$ involves the selection process and, for the minimization of the objective function $f(x_t)$, is given by

$$x_{t+1} = \begin{cases} z_t & if \quad f(z_t) \leq f(x_t) \\ x_t & if \quad f(z_t) > f(x_t) \end{cases} \quad i = 1..n \tag{6}$$

The selection process involves a simple replacement of the original parameter vector with the candidate vector if the objective function decreases by such an action.

Repeating the previous defined operations on each individual of the *t*-th generation will lead to a next generation (the *t*+1 th) with individuals with at least the same performance of the parent population due to Eqn. (6). Generation after generation individuals generated by random mutation will replace their ancestors when they perform better. The closer the population approaches the optimum of the objective function *f*, the smaller the mutations (Eqn. (4)) will become, which results in a more local search towards the optimum.

Metamodel assisted Differential Evolution

The major drawback of evolutionary algorithms such as DE is the total number of evaluations of the objective function 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) by a computationally cheaper method. This could be achieved by using a metamodel, which is a sort of interpolation tool using the already analyzed individuals by the FVM analysis.

The metamodel performs the same task as the high fidelity FVM analysis, 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 used in the present work uses the metamodel as an evaluation tool during the entire evolutionary process [31]. 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 "offline 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.

Two different metamodels, an Artificial Neural Network (ANN) and Kriging, have been used in the present study and their performance has been compared. Both have N input values and M output values.

Artificial Neural Network (ANN)

An ANN, schematically shown in Fig. 6, is composed of several elementary processing units called neurons. These neurons are arranged in layers and joined by connections of different intensity, called connection weights. The network used by the present optimization has three layers: The input layer consists of the *n* design variables, the hidden layer has *K* neurons an the output layer has *m* neurons.

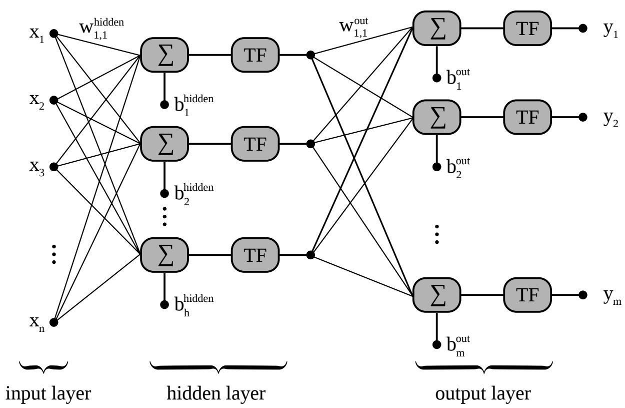

Figure 6: Artificial Neural network (ANN) layout.

The input to output relation for each hidden neuron *j* is given by Eqn. (7), where x_i denotes the input, *n* the number of input connections to the neuron, $w_{i,j}$ the weight given to the connection between the *i*-th input neuron and the *j*-th hidden neuron, b_j is a bias value and out_j the output. A similar relation is used to define the output in function of the hidden layer neurons.

$$out_j = TF\left(\sum_{i=1}^{n} \omega_{i,j}^{hidden} \cdot x_i \quad + b_j^{hidden}\right) \tag{7}$$

The non-linear activation function *TF* is a sigmoid (Eqn. (8)).

$$TF(x) = \frac{1}{1+\exp(-x)} \tag{8}$$

Several techniques exist to train the ANN i.e. to determine the values of the weights $w_{i,j}$ and the bias *bi* for each neuron *i*. In the present work, the standard back propagation technique and training by the use of differential evolution have been used.

Kriging

Kriging was initially developed by geologists to estimate mineral concentrations based only on scarce data available at some places of an area [32]; about the same time it was also introduced in the field of statistics to include the correlations that exist between residuals of a linear estimator [33]. The theory behind interpolation and extrapolation by kriging was developed by the mathematician G. Matheron based on the Master's thesis of D. G. Krige [34] on the use of the statistical techniques to predict the gold grades at the Witwatersrand reef complex in South Africa.

There are many texts in geostatistics [35, 36] and in spatial statistics [37-39] that provide many details on the development and use of kriging models in their respective disciplines. Recently, kriging became of interest to approximate deterministic computer models due to its capability of not only predicting a value, but also to give the uncertainty of the prediction. Several authors can be found that use kriging methods for accelerating an optimization process [40- 45].

Kriging belongs to the family of linear least squares algorithms, such as polynomial response surface models (RSM), however it is reproducing the observed data exactly. The mathematical form of a kriging model has two parts as shown in Eqn. (9). The first part is a linear regression with an arbitrary number *k* of regression functions g_j, that tries to catch the main trend of the response. An RSM model can be used for this purpose, however many authors (e.g. [41,45]) use a constant value for this part and rely on the second part of the model to pull the response surface through the observed data [42].

$$\tilde{f}(\vec{x}) = \sum_{j=1}^{k} \beta_j g_j(\vec{x}) + Z(\vec{x}) \tag{9}$$

The second part, Z(x), is a model of a Gaussian and stationary random process with zero mean. An assumption is made on the mathematical form of the covariance of Z(x), which is usually a Gaussian function [45]. The parameters β_j and the function Z(x) are determined such that $\tilde{f}(\vec{x})$ is the best linear unbiased predictor. A linear estimator means that $\tilde{f}(\vec{x})$ can be written as a linear combination of the observation samples:

$$\tilde{f}(\vec{x}) = \sum_{i=1}^{N} \omega_i(\vec{x}) f_i(\vec{x}) \tag{10}$$

The unbiasedness constraint means that the mean error of the approximation is zero:

$$E\left[f(\vec{x}) - \tilde{f}(\vec{x})\right] = 0 \tag{11}$$

The best linear unbiased predictor is considered the predictor with minimal mean square error (MSE) of the predictions,

$$MSE = E\left[\left(\tilde{f}(\vec{x}) - f(\vec{x})\right)^2\right] \tag{12}$$

One big advantage of kriging above other metamodels is its ability to not only predict the value of the objective function, but also the uncertainty on the prediction (see Fig. 7). The details of constructing kriging models have been thoroughly described in [46, 47, 41, 48, 49, 39].

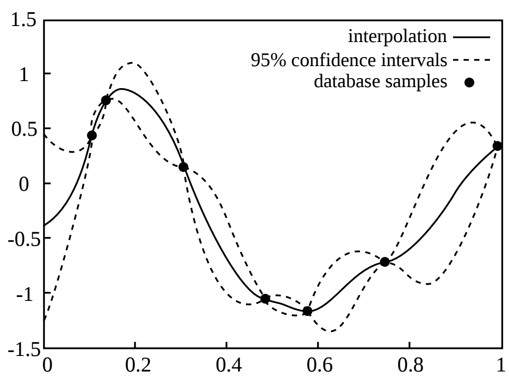

Figure 7: Prediction of mean value and confidence intervals by kriging.

The database

The accuracy of the metamodel predictions strongly depends on the information contained in the database. The Design Of Experiments (DOE) method is used to create the initial database. This maximizes the amount of information contained in it for a limited number of geometries [50].

Each design variable can take two values fixed at 25% and 75% of the maximum design range. A 2^{k-p} factorial design is used. k is the total number of design parameters (26) while p defines the number of lower order parameter combinations that are not analyzed. p is chosen such that in total 2^6=64 samples are generated for the initial database. An additional sample with all parameters at 50% of their range is added, resulting in a total of 65 initial geometries to be analyzed.

Optimization of test functions

The performance of optimization systems is difficult to assess in typical engineering problems requiring FVM computations because the global optimum is not known in most cases. One possibility is to compare the optimum proposed by different optimization algorithms, as is done in the present study. The same evaluation tool must be used (same physical model, discretization and accuracy), because only the difference in optimization performance is of interest. Another method consists of making a systematic sweep over the entire design space and evaluate each design. The global minimum can then be found, and the behavior of the optimizer can be understood.

However, this method becomes unfeasible for more than two design parameters due to the vast number of designs that need to be analyzed. As a consequence, the performance of an optimization system is usually determined by its ability to minimize analytical test functions, for which the solution is known. Moreover, the evaluation of analytical functions requires few computational resources (less than 1 ms as opposed to several hours for FVM), which allows an extensive evaluation of the different settings of the optimization system.

Numerous test functions exist for a variety of specific test purposes. Most of them have been developed to test evolutionary algorithms. In this section only the results for the De Jong test function is presented, although a systematic analysis of different test functions and metamodels have been performed [24].

The De Jong F1 test function is a global optimization test function and is given for a n-dimensional space by:

$$f(\vec{x}) = \sum_{i=1}^{n} x_i^2 \quad , \quad -5.12 \le x_i \le 5.12 \tag{13}$$

The optimum is located at $x_i = 0$. In what follows the results of the optimization of a 20 dimensional De Jong function are described.

In Figs. 8 and 9 the convergence history of the optimization process for respectively a DE with ANN and Kriging are shown. It shows the comparison between the metamodel predictions of the objective function and the validation by the exact formula (Eqn. (13)) after each optimization performed by the DE algorithm. The setting for the optimization are C=0.6, F=0.4, population size T=50, number of generations 1000 and an intitial database counting 65 samples. In the optimization assisted by the kriging metamodel (Fig. 9) a learning behavior is observed. At the first iterations a relatively large difference between the metamodel prediction and the real function value is present. After adding this sample to the database and retraining the kriging metamodel, this difference gradually reduces till the 12th iteration where the difference is minimal. Hereafter the difference increases again but remains smaller than 0.001, which is acceptable. For the ANN assisted optimization, the difference between ANN predictions and reevaluation by the exact function remains large. Moreover, the smallest function value found by the ANN is approximately 9, which is several orders of magnitude larger than the best candidate from the kriging assisted optimization (note the different scales used). One can conclude that for this particular test function the ANN is to be avoided.

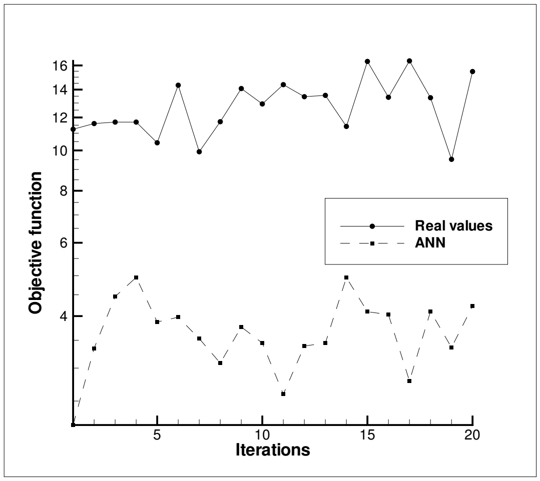

Figure 8: Optimization result of De Jong F1 test function using an Artificial Neural Network (ANN)

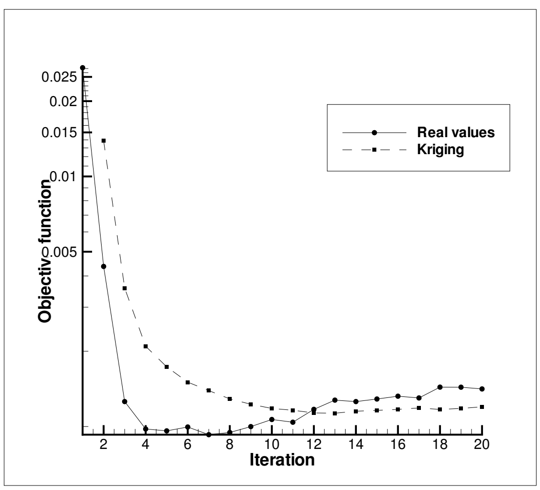

Figure 9: Optimization result of De Jong F1 test function using kriging

In Fig. 10 the same test function has been minimized by the DE algorithm without the assistance of a metamodel. In the abscissa the number of generations is plotted. The number of function evaluations per generation equals the population size (50). After 40 generations, or a total of 2000 evaluations of the objective function, the best individual has an objective function value of 0.1, which is still 2 orders of magnitude higher than the best optimum of the kriging assisted optimization. However, in the kriging assisted optimization, only 20 evaluations of the objective function were performed during the optimization process, and 65 function evaluations were performed prior to the optimization to create the initial database. This shows clearly the advantage of the usage of a metamodel to reduce drastically the number of function evaluations, which is certainly of interest in the case the function evaluation requires an expensive 3D CFD computation. However, not all optimization problems have a simple behavior similar to the De Jong F1 test function with only one well-defined minimum.

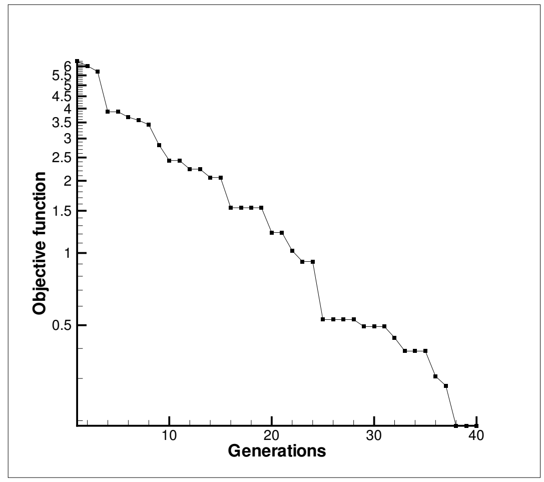

Figure 10: Optimization result using differential evolution without metamodel.

Several other tests performed with other test functions and different dimensions of the design space show similar trends as the case presented here.

RESULTS

In Figs. 11 and 12 the results of the ANN respectively Kriging assisted optimizations are shown. The same behavior of the optimization algorithm as for the De Jong F1 test function is observed, although the difference between the ANN prediction and CFD analysis is less dramatic. The best performing geometry is obtained with the kriging assisted optimization although with the ANN very similar geometries with a slightly higher pressure drop are obtained. The optimal geometry is shown in Fig 13. It has 37.6% reduction in total pressure drop with respect to the standard u-bend. A detailed description for the lower pressure drop within this geometry will be discussed in the second part of this paper, as well as a comparison with measurements.

After the optimization one can analyze the influence of each individual parameter on the objective. In the following discussion the influence of four different parameters on the pressure losses will be investigated.

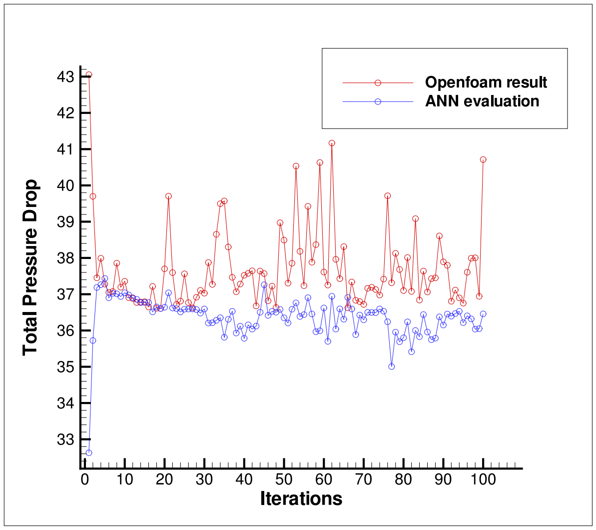

Figure 11: Optimization result using the Artificial Neural Network

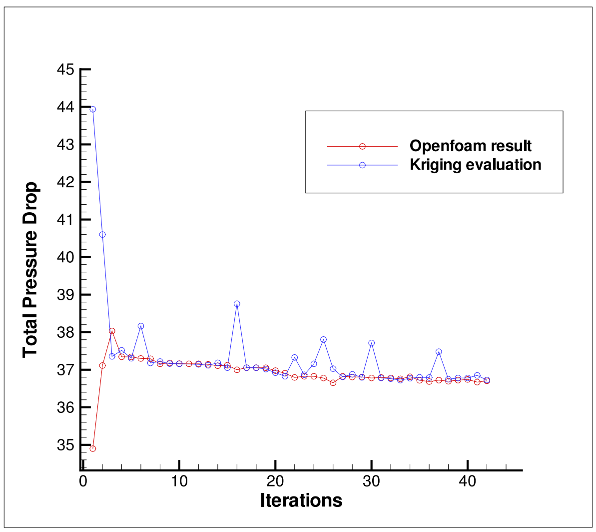

Figure 12: Optimization result using kriging.

In Fig. 14 the influence of the inlet width D1 (see Fig. 4) on the pressure losses is presented. The red circular symbols represent the 65 initial database samples generated prior to the optimization. The squared black symbols represent the samples generated during the ANN optimization, while the diamond black samples represent the samples of the kriging optimization. From the initial database samples alone it is clear that in average a higher D1 results in a lower pressure drop. This trend is followed by the optimizer and most designs have a high D1 value. However, from the samples generated during the optimization it is clear that the maximal value for D1 does not result in the lowest pressure drop. A quadratic regression of all samples is given in Fig. 14. The value of the optimal shape is corresponding closely to the minimum of the quadratic regression, located slightly below the hydraulic diameter of 0.075. This indicates that at the inlet of the 180 turn the flow is not accelerated or decelerated. An acceleration would result in larger centrifugal forces in the bend, while a deceleration would result in a boundary layer that is more sensitive to separation. Both would increase the losses.

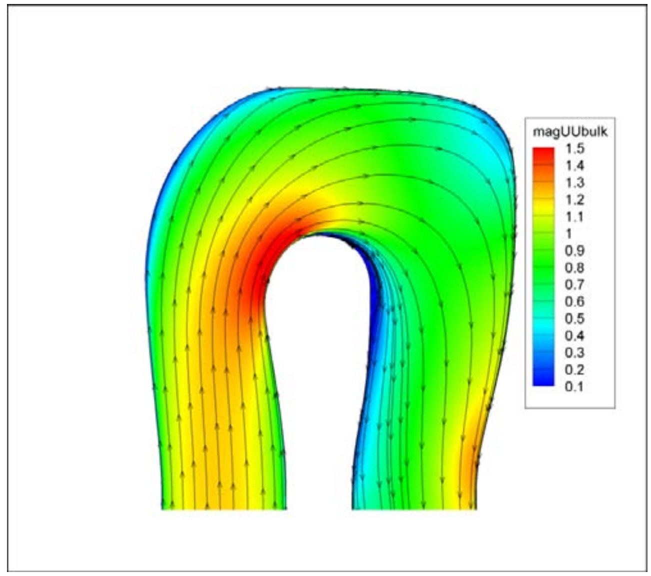

Figure 13: Optimal shape of the U-bend.

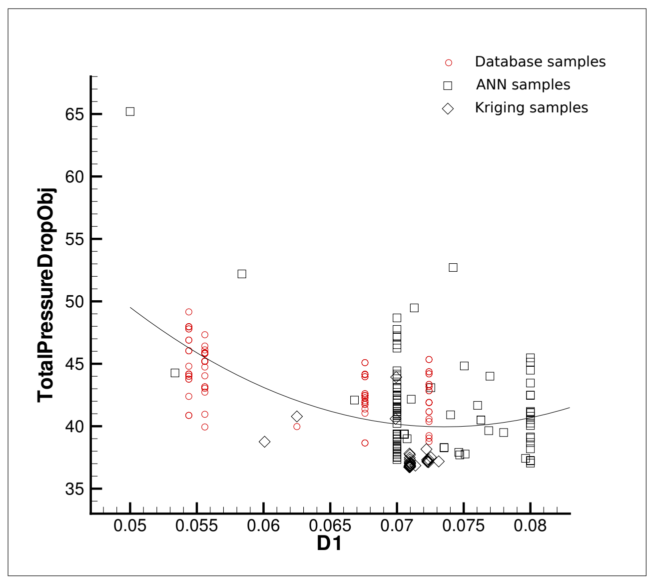

Figure 14: D1 parameter versus pressure drop [Pa] for initial database (circle), ANN (square) and kriging (diamond) samples

In Fig 15 the influence of D2 (see Fig. 4) on the pressure drop is presented. Also here an optimal value is found in between the boundaries of the parameter. An approximately 10% higher value than the hydraulic diameter is the optimal value. It allows for a small diffusion in the middle of the U-bend.

In Fig. 16 the influence of D3 (see Fig. 4) on the pressure drop is shown. A slightly higher value than for D2 is optimal, which includes a small diffusion from 90 degrees towards 180 degrees of the U-bend. For the ANN several geometries are found with a higher value, however resulting in higher pressure losses.

In Fig. 17 the influence of the y-coordinate of the 3rd control point of the 3rd exterior curve (see Fig. 3) on the pressure drop is shown. A clear trend towards the highest possible value is visible. This results in a high curvature in the 3rd exterior curve as seen in Fig. 13. The physical explanation for this large curvature is given in the second part of this paper.

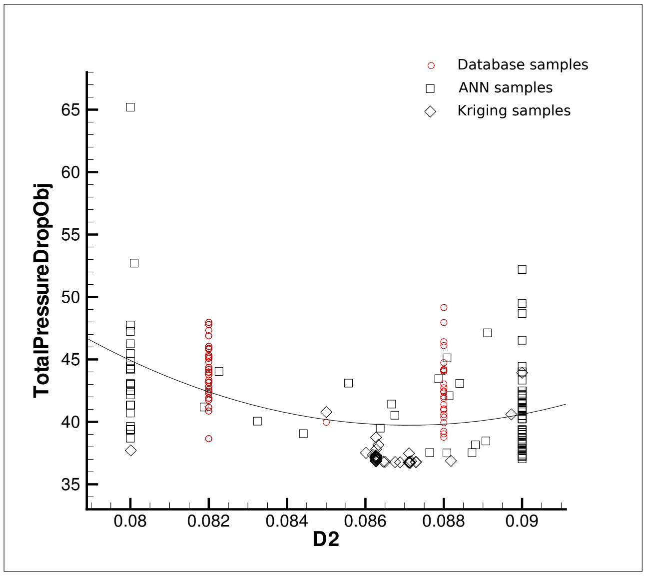

Figure 15: D2 parameter versus pressure drop [Pa] for initial database (circle), ANN (square) and kriging (diamond) samples

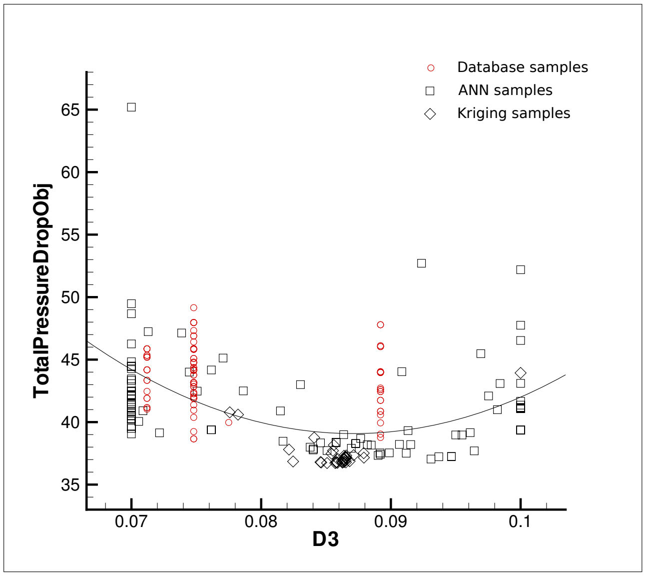

Figure 16: D3 parameter versus pressure drop [Pa] for initial database (circle), ANN (square) and kriging (diamond) samples

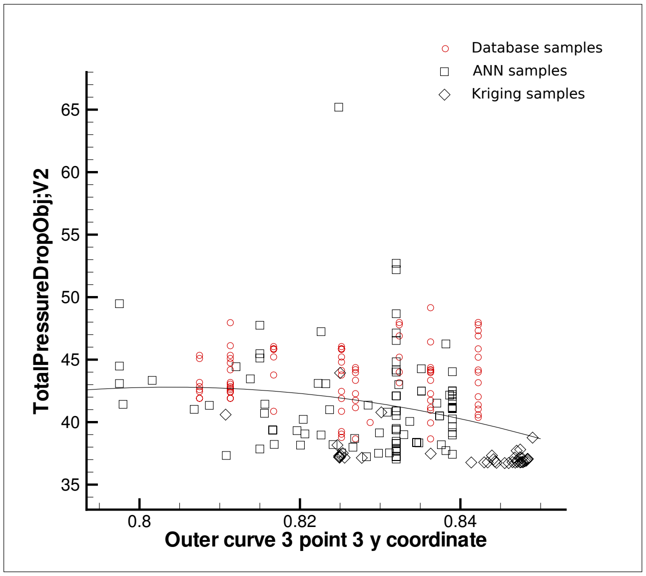

Figure 17: y coordinate of the third control point of the third outer curve versus pressure drop [Pa] for initial database (circle), ANN (square) and kriging (diamond) samples

CONCLUSIONS

This first part of the paper describes an optimization method applied to the design of a U-bend. It uses an automated design procedure involving a Navier-Stokes solver, a metamodel, a database and an evolutionary algorithm. The optimization method is validated by its ability to minimize typical analytical test functions. It is subsequently used to reduce the pressure losses within a U-bend.

Two different metamodels have been used. It is concluded that for the present application with 26 degrees of freedom the kriging metamodel obtains a better performance.

A reduction of 37.6% of the total pressure loss is obtained with respect to the original circular bend. Analysis of the geometries generated during the optimization process allows to reveal important parameters that play a key role in the reduction of the pressure losses. In the second part of this paper the physical interpretation of the flow physics confirms these results.

It shows that the applied method is cost effective and allows to obtain significant improvements within a limited timeframe.

REFERENCES

[1] Han J.-C., Dutta S., Ekkad S., 2000. Gas Turbine Heat Transfer and Cooling Technology. Taylor and Francis, New York.

[2] Weigand B., Semmler K., von Wolfersdorf J., 2006. Heat Transfer Technology for Internal Passages of Air-Cooled Blades for Heavy-Duty Gas Turbines. Annals of the New York Academy of Sciences 934, pp. 179-193

[3] Humphrey J. A. C , Whitelaw J. H., Yee G., 1981. Turbulent Flow in a Square Duct with Strong Curvature. J of Fluid Mech 103, pp. 443-463.
[4] Chang S. M., Humphrey J. A. C, Modavi A., 1983. Turbulent Flow in a Strongly Curved U-Bend and Downstream Tangent of Square Cross-Sections. Physico-Chemical Hydrodynamics 4, pp. 243-269.
[5] Monson D.J., Seegmiller H. L., 1992. An Experimental Investigation of Subsonic Flow in a Two Dimensional U-Duct. NASA report TM-103931.
[6] Cheah S.C., Iacovides H., Jackson D.C., Ji H.H., Launder B.E., 1996. LDA Investigation of the Flow Development Through Rotating U-ducts. J. Turbomachinery 118, 590-596.
[7] Liou T.-M., Chen C.-C., 1999. LDV Study of Developing Flow Through a Smooth Duct With a 180 deg Straight-Corner Turn. J Turbomach, 121(1), pp. 167-174
[8] Son S.Y., Kihm K.D., Han J.-C., 2002. PIV Flow Measurements for Heat Transfer Characterization in Two-Pass Square Channels With Smooth and 90° Ribbed Walls. Int J Heat and Mass Trans 45, pp. 4809-4822
[9] Schabacker J., Boelcs A., Johnson B. V., 1998. PIV Investigation of the Flow Characteristics in an Internal Coolant Passage with Two Ducts Connected by a Sharp 180deg Bend. ASME paper 98-GT-544.
[10] Iacovides H., Launder B. E., 1996. Computational Fluid Dynamics Applied to Internal Gas-Turbine Blade Cooling: A Review. Int J Heat and Fluid Flow 16, pp. 454-470.
[11] Lucci J. M., Amano R. S., Guntur K., 2007. Turbulent Flow and Heat Transfer in Variable Geometry U-Bend Blade Cooling Passage, ASME paper GT2007-27120.
[12] Schüler M., Neumann S. O., Weigand B., 2010. Numerical Investigations of Pressure Loss and Heat Transfer in a 180° Bend of a Ribbed Two-Pass Internal Cooling Channel with Engine-Similar Cross-Sections, J of Power and Energy 224, pp. 349-361
[13] Bonhoff B., Tomm U., Johnson B. V., 1996. Heat Transfer Predictions for U-Shaped Coolant Channels with Skewed Ribs and With Smooth Walls. ASME paper 96-TA-7.
[14] Chen H.-C., Jang Y.-J., Han J.-C., 2003. Computation of Heat Transfer in Rotating Two-Pass Square Channels by a Second-Moment Closure Model. Int J Heat Mass Transfer 43, pp. 1603-1616.
[15] Metzger D. E., Plevich C. W., Fan C. S., 1984. Pressure Loss Through Sharp 180◦ Turns in Smooth Rectangular Channels. J of Engineering for Gas Turbine and Power 106, pp. 677-681.
[16] Liou T.-M., Tzeng Y.-Y, Chen, C.-C., 1999. Fluid Flow in a 180◦ Sharp Turning Duct With Different Divider Thicknesses. J Turbomach 121, pp. 569-576.
[17] Bonhoff B., Leusch J., Johnson B. V., 1999. Predictions of Flow and Heat Transfer in Sharp 180-deg Turns of Gas Turbine Coolant Channels with and without Turning Vanes, 33rd National Heat Transfer Conference, Albuquerque, NM, USA.
[18] Schüler M., Zehnder F., Weigand B., von Wolfersdorf J., Neumann S.O., 2011. The Effect of Turning Vanes on Pressure Loss and Heat Transfer of a Ribbed Rectangular Two-Pass Internal Cooling Channel. J Turbomach 133, 021017 (10 pages).
[19] Zehner S., Steinbrück H., Neumann S. O., Weigand B., 2009. The Ice Formation Method: A Natural Approach to Optimize Turbomachinery Components. Int J of Design & Nature 3, pp.259–272
[20] Namgoong H., Son C., Ireland P., 2008. U-bend Shaped Turbine Blade Cooling Passage Optimization. AIAA paper ISSMO 2008-5926
[21] Coletti F., Verstraete T., Vanderwielen T., Bulle J., Arts T., 2011. Optimization of a U-bend for Minimal Pressure Loss in Internal Cooling Channels – Part I: Experimental Validation. ASME paper GT2011-46555
[22] Pierret, S., 1999, “Designing Turbomachinery Blades by Means of the Function Approximation Concept Based on Artificial Neural Network, Genetic Algorithm, and the Navier-Stokes Equations”, Ph.D. Thesis, Faculté Polytechnique de Mons-Von Karman Institute for Fluid Dynamics.
[23] Pierret, S., and Van den Braembussche, R.A., 1999, “Turbomachinery Blade Design Using a Navier Stokes Solver and Artificial Neural Networks”, ASME journal of turbomachinery, Vol. 121, pp.326-332.
[24] Verstraete, T., “Multidisciplinary Turbomachinery Component Optimization Considering Performance, Stress, and Internal Heat Transfer”,Ph.D. Thesis, Universty of Ghent-Von Karman Institute.
[25] Gambit Academic Research, Release 2.4.6, URL http::// www.ansys.com
[26] Pope S. B., 2000. Turbulent Flows, Cambridge University Press.
[27] Open Foam, URL http://www.openfoam.com.
[28] Holland, J.H., 1975, “Adaption in Natural and Artificial Systems”, University of Michigan Press.
[29] Rechenberg, I., 1973, “Evolutionsstrategie — Optimierung technischer Systeme nach Prinzipien der biologischen Evolution”, Fommann-Holzboog, Stuttgart.
[30] Price, K., and Storn, N., 1997, “Differential Evolution”, Dr. Dobb’s Journal, pages 18–24, April.
[31] Verstraete, T., 2010, “CADO: a Computer Aided Design and Optimization Tool for Turbomachinery Applications”, Presented at 2nd Int. Conf. on Engineering Optimization, Lisbon, Portugal, September 6-9, 2010.

[32] Matheron, G., 1963, “Principles of Geostatistics”, Economic Geology, 58:1246–1266.
[33] Goldberger, A.S., 1962, ”Best Linear Unbiased Prediction in the Generalized Linear Regression Model”, Journal of the American Statistical Association, 57(298):369–375.
[34] Krige, D.G., 1951, “A statistical approach to some mine valuations and allied problems at the Witwatersrand”,. PhD thesis, University of Witwatersrand.
[35] Goovaerts, P., 1997, ”Geostatistics for Natural Resources Evaluation”, Oxford Univ. Press, New York.
[36] Journel, A.G., and Huijbregts, C.J., 1978, ”Mining Geostatistics”, Academic Press, New York.
[37] Cressie, N. A. C., 1991,”Statistics for Spatial Data”, Wiley, New York.
[38] Ripley, B.D., 1981, “Spatial Statistics”, Wiley, New York.
[39] Stein, M.L., 1999, “Interpolation of Spatial Data: Some theory of Kriging”, Springer-Verlag, New York.
[40] S. E. Gano, S. S., and Renaud, J.E., 2005, ”Hybrid Variable Fidelity Optimization by Using a Kriging-Based Scaling Function”, AIAA Journal, 43(11):2422–2430.
[41] Jeong, S, Murayama, M, and Yamamoto, K., 2005, “Efficient Optimization Design Method Using Kriging Model”, Journal of Aircraft, 42(2):413–420.
[42] Martin, J. D., and Simpson, T.W., 2005, “Use of Kriging Models to Approximate Deterministic Computer Models”, AIAA Journal, 43(4):853–863.
[43] Martin, J. D., and T. W. Simpson, T.W., 2004, “A Monte Carlo Simulation of the Kriging Model”, AIAA Paper No. 2004-4483.
[44] Simpson, T W., 1998, “Comparison of Response Surface and Kriging Models in the Multidisciplinary Design of an Aerospike Nozzle”, ICASE Report No.98-16.
[45] Simpson, T. W., Peplinski, J.,Koch, P. N., and Allen, J.K., 200, “Metamodels for Computer-Based Engineering Design: Survey and Recommendations”, Engineering with Computers, 17(2):129–150.
[46] Denison, D. G. T., Holmes, C. C., Mallick, B. K., and Smith, A. F. M., 2002, “Bayesian Methods for Nonlinear Classification and Regression”, Willey, New York.
[47] Guinta, A. A., and Watson, L. T.,1998, “A Comparison of Approximation Modeling Techniques: Polynomial Versus Interpolating Models”, AIAA Paper No. 98-4758.
[48] Jones, D. R., Schonlau, M., and Welch, W. J., 1998,”Efficient Global Optimization of Expensive Black-Box Functions”, Journal of Global Optimization, 13(4):455–492.
[49] Simpson, T. W., Mauery, M. T., Korte, J. J., and Mistree, F., 2001,” Kriging Metamodels for Global Approximation in Simulation-Based Multi-Disciplinary Design Optimization”, AIAA Journal, 39(12):2233–2241.
[50] Montgomery, D.C., 1997, “Design and Analysis of Experiments”, John Wiley and Sons, New York.