; :–
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/

Insight into Fluid Structure Interaction Benchmarking through Multi-Objective Optimization

M. Razzaq[1*], C. Tsotskas[2*], S. Turek[1], J. Hron[3], T. Kipouros[2] and M. Savill[2]

[1] *Institute of Applied Mathematics, TU Dortmund, Germany.*
[2] *Cranfield University,MK430AL, Bedfordshire, Department of Power and Propulsion,School of Engineering, UK.*
[3] *Institute of Mathematics, Charles University Prague, Czech Republic.*

SUMMARY

The integration and application of a new multi-objective tabu search optimization algorithm for Fluid Structure Interaction (FSI) problems are presented. The aim is to enhance the computational design process for real world applications and to achieve higher performance of the whole system for the four considered objectives. The described system combines the optimizer with a well established FSI solver which is based on the fully implicit, monolithic formulation of the problem in the Arbitrary Lagrangian-Eulerian FEM approach. The proposed solver resolves the proposed fluid-structure interaction benchmark which describes the self-induced elastic deformation of a beam attached to a cylinder in laminar channel flow. The optimized flow characteristics of the aforementioned geometrical arrangement illustrate the performance of the system in two dimensions. Special emphasis is given to the analysis of the simulation package, which is of high accuracy and is the core of application. The design process identifies the best combination of flow features for optimal system behavior and the most important objectives. In addition, the presented methodology has the potential to run in parallel, which will significantly speed-up the elapsed time.

Received . . .

KEY WORDS: Finite Element Method (FEM); Fluid-Structure Interaction (FSI); Multi-Ojective Tabu search (MOTS2).

1. MULTI-OBJECTIVE OPTIMIZATION IN FLUID STRUCTURE INTERACTION

By definition, optimization seeks for the best possible performance of a model, which is formulated in a mathematical way as the minimization of a function or a set of functions at the same time. This denotes single- and multi-objective optimization, respectively. The response of the design space to the objective space (set of problem variables and objectives, respectively) could be either linear or non-linear, and continuous or discrete. Thus, exploring effectively the design space and concentrating around the regions where there most optimal values reside is of paramount importance and the whole optimization process was developed in order to tackle this requirement.

The vast majority of real world applications depends on several variables of a given model. The handling of participating variables appropriately is the key for successful optimization. The optimization is applied on a model, which in turn approaches the real behavior of various phenomena found in nature. Defining performance metric(s) is the considered objective(s) and the goal of optimization is to discover the best combination of variables that yields the best performance. Since

*Correspondence to: M. Razzaq[1] Institute of Applied Mathematics, TU Dortmund, Germany. [1]E-mails:mrazzaq@math.uni-dortmund.de, [2]c.tsotskas@cranfield.ac.uk, [1]ture@featflow.de, [3]jaroslav.hron@mff.cuni.cz, [2] t.kipouros@cranfield.ac.uk, [2]mark.savill@cranfield.ac.uk.

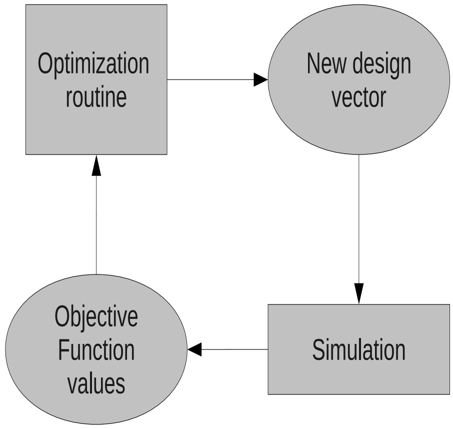


Figure 1. Overview of the Integrated Optimization Process

no global optimum exists when many objectives are defined, focusing on the conflicting objectives is essential.

Computational tools for multi-objective and multi-disciplinary optimization are of paramount importance throughout the design process of real-world applications [**?**], [**?**]. Recently, the increase of computational power favors implementations, which employ these principles. This can considerably reduce the duration of the design cycle and deliver high quality products. The functionality of a new optimizer and its application on a real world problem are presented in this document. The concept of engineering design optimization was conceived and implemented by Cranfield University and TU Dortmund provided the FSI simulation code. The optimization process is applied on FSI [**?**] and the results are analyzed from a multi-objective optimization point of view.

The integration of the FSI solver into an optimization procedure for FSI problems has been reported in [**?**] and fluid structure interaction in the context of shape optimization and computational wind engineering is contributed in [**?**].

The approach presented here treats the problem as a pipeline - a single continuum with the coupling implemented as internal interface, which does not require any special treatment, as depicted in Figure 1. For further details of the underlying numerical aspects of the discretization and solution algorithms for this monolithic approach, see [**?**, **?**, **?**]. The presented optimization process follows the methodology of Multi-Objective Tabu Search (MOTS) [**?**], which stems from the original tabu search [**?**]. Furthermore, a new variant of the former, namely MOTS2 [**?**], has been developed and used in this study. It can be considered as combined extension of numerical analysis tools and artificial intelligence optimization methods and techniques. In addition, MOTS2 includes the improvements discussed in [**?**] and, given any parallel framework, it can operate in parallel mode saving elapsed time. The remaining of this paper is structured as follows. First the mathematical background for the core simulation model, the FSI package, is described, followed by the numerical techniques. The computational approach for FSI is illustrated in section 4. The next part presents and discusses two optimization cases; one single- and one multi- objective.

2. GOVERNING EQUATIONS FOR FSI

The governing equations for fluid and structure are described in the following subsections. We denote by Ω_t^f and Ω_t^s the domains occupied by the fluid and the structure, resp., at the time $t \geq 0$. Let $\Gamma_t^0 = \bar{\Omega}_t^f \cap \bar{\Omega}_t^s$ be the part of the boundary where the elastic structure interacts with the fluid.

()
DOI: 10.1002/

2.1. Fluid

The fluid is considered to be *Newtonian, incompressible* and its state is described by the velocity and pressure fields $\mathbf{v}^f, p^f$. The balance equations are

$$\left.\begin{aligned} \varrho^f \frac{\partial \mathbf{v}^f}{\partial t} + \varrho^f (\nabla \mathbf{v}^f)\mathbf{v}^f &= \operatorname{div} \sigma^f \\ \operatorname{div} \mathbf{v}^f &= 0 \end{aligned}\right. \quad \text{in } \Omega_t^f. \tag{1}$$

The material constitutive equation is

$$\sigma^f = -p^f \mathbf{I} + \varrho^f \nu^f (\nabla \mathbf{v}^f + \nabla {\mathbf{v}^f}^{\mathrm{T}}). \tag{2}$$

The constant density of the fluid is ϱ^f and the viscosity is denoted by ν^f.

2.2. Structure

The structure is assumed to be *elastic* and *compressible*. Its deformation is described by the displacement $\mathbf{u}^s$, with velocity field $\mathbf{v}^s = \frac{\partial \mathbf{u}^s}{\partial t}$. The balance equations are

$$\varrho^s \frac{\partial \mathbf{v}^s}{\partial t} + \varrho^s (\nabla \mathbf{v}^s)\mathbf{v}^s = \operatorname{div}(\sigma^s) \quad \text{in } \Omega_t^s. \tag{3}$$

The material is specified by the Cauchy stress tensor σ^s or by the 2nd Piola-Kirchhoff stress tensor $\mathbf{S}^s = J\mathbf{F}^{-1}\sigma^s\mathbf{F}^{-T}$ via the *St. Venant-Kirchhoff* constitutive law

$$\sigma^s = \frac{1}{J}\mathbf{F}\left(\lambda^s (\operatorname{tr}\mathbf{E})\mathbf{I} + 2\mu^s \mathbf{E}\right)\mathbf{F}^{\mathrm{T}}, \tag{4}$$

$$\mathbf{S}^s = \lambda^s (\operatorname{tr}\mathbf{E})\mathbf{I} + 2\mu^s \mathbf{E}, \tag{5}$$

where $\mathbf{E} = \frac{1}{2}(\mathbf{F}^{\mathrm{T}}\mathbf{F} - \mathbf{I})$ is the Green-St. Venant strain tensor.

The density of the structure in the undeformed configuration is ϱ^s. The elasticity of the material is characterized by the Poisson ratio ν^s ($\nu^s < 0.5$ for a compressible structure) and by the Young modulus E^s. The alternative characterization is described by the Lamé coefficients λ^s and μ^s (the shear modulus):

$$\nu^s = \frac{\lambda^s}{2(\lambda^s + \mu^s)} \qquad E^s = \frac{\mu^s (3\lambda^s + 2\mu^s)}{(\lambda^s + \mu^s)} \tag{6}$$

$$\mu^s = \frac{E^s}{2(1 + \nu^s)} \qquad \lambda^s = \frac{\nu^s E^s}{(1 + \nu^s)(1 - 2\nu^s)} \tag{7}$$

2.3. Complete set of equations for Fluid Structure Interaction

In the case of fluid-structure interaction problems the Lagrangian description for the deformation of the structure part still can be used. The fluid flow now takes place in a domain with boundary given by the deformation of the structure which can change in time and is influenced back by the fluid flow. The mixed ALE description of the fluid has to be used in this case. The fundamental quantity describing the motion of the fluid is still the velocity vector but the description is accompanied by a certain displacement field which describes the change of the fluid domain. This displacement field has no connection to the fluid velocity field and the purpose of its introduction is to provide a transformation of the current fluid domain and corresponding governing equations to some fixed reference domain. This method is sometimes called a **pseudo-solid mapping method** [**?**].

()
DOI: 10.1002/

The complete set of the non-dimensionalized system with the described choice of material relations reads:

$$\frac{\partial \mathbf{u}}{\partial t} = \begin{cases} \mathbf{v} & \text{in } \Omega^s, \\ \Delta \mathbf{u} & \text{in } \Omega^f, \end{cases} \tag{8}$$

$$\frac{\partial \mathbf{v}}{\partial t} = \begin{cases} \frac{1}{\beta}\,\mathrm{Div}\left(-Jp^s\mathbf{F}^{-T}\right) & \text{in } \Omega^s, \\ -(\mathrm{Grad}\,\mathbf{v})\mathbf{F}^{-1}(\mathbf{v} - \frac{\partial \mathbf{u}}{\partial t}) & \\ \quad + \mathrm{Div}\left(-Jp^f\mathbf{F}^{-T} + J\mu\,\mathrm{Grad}\,\mathbf{v}\mathbf{F}^{-1}\mathbf{F}^{-T}\right) & \text{in } \Omega^f, \end{cases} \tag{9}$$

$$0 = \begin{cases} J - 1 & \text{in } \Omega^s \\ \mathrm{Div}(J\mathbf{v}\mathbf{F}^{-T}) & \text{in } \Omega^f \end{cases} \tag{10}$$

where $\beta = \frac{\varrho^s}{\varrho^f}$ is the density ratio. The boundary conditions on the fluid-structure interface are assumed to be

$$\begin{aligned} \sigma^f\mathbf{n} &= \sigma^s\mathbf{n} \\ \mathbf{v}^f &= \mathbf{v}^s \end{aligned} \tag{11}$$

where $\mathbf{n}$ is a unit normal vector to the interface. This implies the no-slip condition for the flow, and that the forces on the interface are in balance.

3. FEM DISCRETIZATION

The discretization in space is done by the standard Galerkin finite element method. Let $I = [0, T]$ denote the time interval of interest. The equations (8)-(10) are multiplied by the test functions ζ, ξ, γ such that $\zeta = \mathbf{0}$ on Γ^2 (external boundary of structure), $\xi = \mathbf{0}$ on Γ^1 (external boundary of fluid), and integrated over the space domain Ω and the time interval I. Using integration by parts on some of the terms and the boundary conditions leads to

$$\int_0^T \int_\Omega \frac{\partial \mathbf{u}}{\partial t} \cdot \zeta dV dt = \int_0^T \int_{\Omega^s} \mathbf{v} \cdot \zeta dV dt - \int_0^T \int_{\Omega^f} \mathrm{Grad}\,\mathbf{u} \cdot \mathrm{Grad}\,\zeta dV dt, \tag{12}$$

$$\begin{aligned} \int_0^T \int_{\Omega^f} J\frac{\partial \mathbf{v}}{\partial t} \cdot \xi dV dt &+ \int_0^T \int_{\Omega^s} \beta J \frac{\partial \mathbf{v}}{\partial t} \cdot \xi dV dt \\ &= - \int_0^T \int_{\Omega^f} J\,\mathrm{Grad}\,\mathbf{v}\mathbf{F}^{-1}(\mathbf{v} - \frac{\partial \mathbf{u}}{\partial t}) \cdot \xi dV dt \\ &\quad + \int_0^T \int_\Omega Jp\mathbf{F}^{-T} \cdot \mathrm{Grad}\,\xi dV dt \\ &\quad - \int_0^T \int_{\Omega^s} \frac{\partial \Psi}{\partial \mathbf{F}} \cdot \mathrm{Grad}\,\xi dV dt \\ &\quad - \int_0^T \int_{\Omega^f} J\mu\,\mathrm{Grad}\,\mathbf{v}\mathbf{F}^{-1}\mathbf{F}^{-T} \cdot \mathrm{Grad}\,\xi dV dt, \end{aligned} \tag{13}$$

$$0 = \int_0^T \int_{\Omega^s} (J-1)\gamma dV dt + \int_0^T \int_{\Omega^f} \mathrm{Div}(J\mathbf{v}\mathbf{F}^{-T})\gamma dV dt. \tag{14}$$

The treatment of the problem as one monolithic system suggests to use the same finite elements on both the structure part and the fluid region. A pair of finite element spaces known to be stable for problems with incompressibility constraint is chosen. The compatibility condition between the velocity space and pressure space is satisfied by the so called inf-sup or LBB condition named after

()
DOI: 10.1002/

Ladyzhenskaya, Babuska and Brezzi [**?**],

$$\sup_{\mathbf{u}\in W_h} \frac{\int_\Omega \operatorname{div} \mathbf{u} q}{\|\mathbf{u}\|_{1,\Omega}} \geq \gamma \|q\|_{0,\Omega} \quad \forall q \in Q_h \tag{15}$$

where γ is a mesh-independent constant. $W_h := U_h \times V_h \subset C(\Omega)$ and $Q_h \subset L^2_0(\Omega)$.

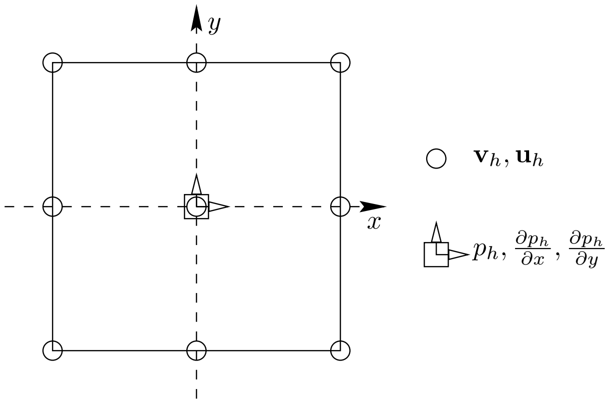


Figure 2. Location of the degrees of freedom for the Q_2P_1 element

The LBB-stable conforming biquadratic, discontinuous linear finite element pair Q_2P_1 is invoked, which is most accurate and robust finite element pairs for highly viscous incompressible flow (see [**?**], [**?**], [**?**]). This choice results in 39 degrees of freedom per element for $2D$, see Figure 2 for the location of the degrees of freedom.

Then, the variational formulation of the fluid-structure interaction problem is to find $(\mathbf{u}_h, \mathbf{v}_h, p_h) \in U_h \times V_h \times P_h$ such that the equations (12), (13) and (14) are satisfied for all $(\zeta_h, \xi_h, \gamma_h) \in U_h \times V_h \times P_h$ including initial conditions.

The spaces U_h, V_h, P_h on an interval $[t^n, t^{n+1}]$ are defined in the case of the Q_2P_1 pair as follows

$$\begin{aligned}
U_h &= \{\mathbf{u}_h \in [C(\Omega_h)]^2, \mathbf{u}_h|_T \in [Q_2(T)]^2 \quad \forall T \in \mathcal{T}_h, \mathbf{u}_h = \mathbf{0} \text{ on } \partial\Omega_h\},\\
V_h &= \{\mathbf{v}_h \in [C(\Omega_h)]^2, \mathbf{v}_h|_T \in [Q_2(T)]^2 \quad \forall T \in \mathcal{T}_h, \mathbf{v}_h = \mathbf{0} \text{ on } \partial\Omega_h\},\\
P_h &= \{p_h \in L^2(\Omega_h), p_h|_T \in P_1(T) \quad \forall T \in \mathcal{T}_h\}.
\end{aligned}$$

After discretization in space by the finite element method (i.e. Q_2P_1), derive the system of nonlinear algebraic equations arising from the governing equations in each time step

$$\begin{pmatrix} \mathbf{S_{uu}} & \mathbf{S_{uv}} & \mathbf{0} \\ \mathbf{S_{vu}} & \mathbf{S_{vv}} & k\mathbf{B} \\ c_{\mathbf{u}}\mathbf{B_s^T} & c_{\mathbf{v}}\mathbf{B}_f^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u}_h \\ \mathbf{v}_h \\ p_h \end{pmatrix} = \begin{pmatrix} \text{rhs}\mathbf{u} \\ \text{rhs}\mathbf{v} \\ \text{rhs}p \end{pmatrix}, \tag{16}$$

where $\mathbf{S}$ describes the reactive, diffusive and convective terms from the governing equations, $\mathbf{B}$ is the discrete gradient operator and $\mathbf{B^T}$ is the discrete divergence operator.

4. SOLVER

The above system of nonlinear saddle point type of the algebraic equations in (16) is solved using the Newton method as basic iteration which can exhibit quadratic convergence. The basic idea of the Newton iteration is to find a root

$$\mathbf{R}(\mathbf{X}) = \mathbf{0}, \tag{17}$$

()
DOI: 10.1002/

using the available known function value and its first derivative. One step of the Newton iteration with damping results in iterations of the form

$$\mathbf{X}^{n+1} = \mathbf{X}^n + \omega^n \left[\frac{\partial \mathbf{R}(\mathbf{X^n})}{\partial \mathbf{X}}\right]^{-1} \mathbf{R}(\mathbf{X}^n) \tag{18}$$

where $\mathbf{X} = (\mathbf{u}_h, \mathbf{v}_h, p_h)$. The Jacobian matrix $\frac{\partial \mathbf{R}(\mathbf{X^n})}{\partial \mathbf{X}}$ can be computed by finite differences from the residual vector $\mathbf{R}(\mathbf{X})$

$$\left[\frac{\partial \mathbf{R}(\mathbf{X^n})}{\partial \mathbf{X}}\right]_{ij} \approx \frac{[\mathbf{R}]_i(\mathbf{X}^n + \alpha_j \mathbf{e}_j) - [\mathbf{R}]_i(\mathbf{X}^n - \alpha_j \mathbf{e}_j)}{2\alpha_j}, \tag{19}$$

where $\mathbf{e}_j$ are the unit basis vectors in $\mathbf{R}^n$ and the coefficients $\alpha_j > 0$ are increments at each iteration step n of the iteration (18), which can be taken adaptively according to the change in the solution in the previous time step or can be fixed. We set this parameter to be fixed, i.e.,

$$\alpha_j = -b_* \epsilon_M$$

where b_* parameter to be assigned at start and $\epsilon_M = \sqrt{\text{DBL\_Machine}}$, see [**?**, **?**].

The damping parameter $\omega^n \in (-1, 0)$ is chosen such that

$$\mathbf{R}(\mathbf{X}^{n+1}) \cdot \mathbf{X}^{n+1} \leq \mathbf{R}(\mathbf{X}^n) \cdot \mathbf{X}^n.$$

The damping greatly improves the robustness of the Newton iteration in the case when the current approximation $\mathbf{X}^n$ is not close enough to the final solution, see [**?**, **?**] for more details.

In this considered $2D$ problem a direct solver for sparse systems like UMFPACK [**?**] is used. This choice provides very robust linear solvers however its memory and CPU time requirements are too high for larger systems (i.e. more than $20,000$ unknowns). In that case the standard geometric multigrid approach is utilized, for details see [**?**, **?**]. As the sparsity pattern of the Jacobian matrix is known in advance, which is given by the used finite element method, this computation can be done in an efficient way so that the linear solver remains the dominant part in terms of the CPU time (see [**?**] for more details).

5. FSI OPTIMIZATION BENCHMARKING

This FSI optimization benchmark is based on the 2D steady FSI problem from the benchmark configuration of Turek and Hron [**?**] with additional altered boundary control flows as shown in Figure **??**.

5.1. Defining the objective functions

The quantities of interest are with respect to the position of the point A (Figure **??**):

1. The displacements $u_x(t)$ and $u_y(t)$ in x- and y-direction of the point A at the end of the beam structure (see Figure **??**).
2. Forces exerted by the fluid on the *whole* submerged body, i.e. lift and drag forces acting on the cylinder and the beam structure together

$$(F_D, F_L)^{\mathrm{T}} = \int_S \sigma^f \mathbf{n}\, dS = \int_{S_1} \sigma^f \mathbf{n}\, dS + \int_{S_2} \sigma^f \mathbf{n}\, dS,$$

where $S = S_1 \cup S_2$ (see Figure 3) denotes the part of the circle being in contact with the fluid and $\mathbf{n}$ is the outer unit normal vector to the integration path with respect to the fluid domain.

Finally, numerical results for this problem involving optimization for a steady fluid-structure interaction are given here to illustrate the capability of the approach considered.

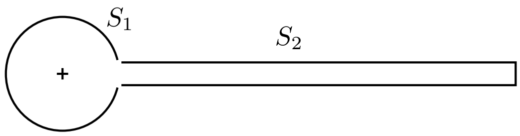


Figure 3. Integration path $S = S_1 \cup S_2$ for the force calculation

5.2. Formulating the Optimization Problem

The idea is to integrate the FSI solver into an optimization procedure for FSI problems. Furthermore, these FSI configurations can be extended towards optimal control of body forces acting on and deformations of the elastic object in which case additional outer in flow/out flow regions control the optimal result.

Two scenarios are presented below: one single and one multi-objective optimization. As single-objective optimizer a derivative-free optimization method for this unconstrained minimization problem is chosen, which is the SIMPLEX algorithm developed by Nelder and Mead [**?**, **?**]. The method is wide spread due to the fact that it makes no assumptions about the objective function except that it is continuous and it is quite numerically robust [**?**, **?**]. Furthermore, MOTS2 will perform the multi-objective optimization and its functionality is described in **??**.

Definition The problem domain, which is based on the 2D version of the well-known FSI benchmark in [**?**], is illustrated in Figure **??**. The thickness of the beam is increased from $0.02m$ to $0.04m$.

An objective function is the minimization of lift/drag forces on the deformable structures through boundary flow control. Mathematically this optimization problem can be written as

$$\underset{V_1,V_2}{\text{minimize}} \left(lift(V_1,V_2)^2 + \alpha V_C^2 \right) \tag{20}$$

where α is the normalization parameter. The control velocity profile from the region a_1 and the region a_2 is prescribed in the following

$$v_C^f(x,0) = V_C = \begin{cases} V_1(x-0.45)(x-0.60), & a_1 \\ V_2(x-0.45)(x-0.60), & a_2. \end{cases} \tag{21}$$

Similarly, the multi-objective optimization problem is formulated as

$$\underset{V_1,V_2}{\text{minimize}} \quad lift(V_1,V_2), drag(V_1,V_2), u_x(V_1,V_2), u_y(V_1,V_2) \tag{22}$$

where V_1 is the magnitude of the parabolic velocity from/to the region a_1 and the region V_2 velocity from/to a_2. Also, u_x, u_y denote the horizontal and vertical displacement of point A respectively. The region a_1 and the region a_2 are specified between the points $(0.45, 0)$ $(0.60, 0)$ and the points $(0.45, 0.41)$ $(0.60, 0.41)$, respectively (see Figure **??**). A parabolic velocity profile is prescribed at the left channel inflow

$$v^f(0,y) = 1.5\bar{U}\frac{y(H-y)}{\left(\frac{H}{2}\right)^2} = 1.5\bar{U}\frac{4.0}{0.1681}y(0.41-y), \tag{23}$$

where $\bar{U} = 0.2m/s$ denotes the mean inflow velocity in x-direction and H denotes the channel height. The outflow condition is chosen as *stress free*. The outflow condition effectively prescribes some reference value for the pressure variable p. In this paper, the reference pressure at the outflow is set to have *zero mean value*. The *no-slip* condition is prescribed for the fluid on the other boundary parts, i.e. top and bottom wall, circle and fluid-structure interface.

The prototypical parameters for the fluid (glycerine) and rubber-like materials (polypropylene) are: The density and kinematic viscosity of the fluid are $\rho^f = 1000kg/m^3$, $\nu^f = 10^{-3}m^2/s$,

0
DOI: 10.1002/

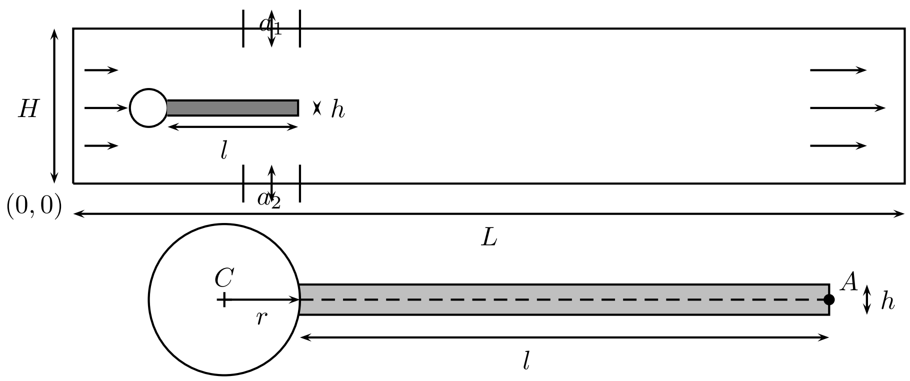
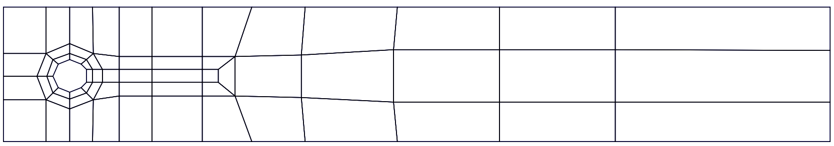


Figure 4. Geometry and computational domain of the configuration and details of the structure part

respectively. Thus the Reynolds number is $Re = 20$ based on the cylinder diameter. The density of the structure is $\rho^s = 1000kg/m^3$, the Young modulus is $E = 178000kg/ms^2$ and the Poisson ratio is $\mu^s = 0.4$.

The domain has length $L = 2.5m$ and height $H = 0.41m$, the circle center is positioned at $C = (0.2, 0.2)$ (measured from the left bottom corner of the channel) and the radius is $r = 0.05m$, the elastic structure beam has length $l = 0.35m$ and height $h = 0.02m$, the right bottom corner is positioned at $(0.6, 0.19)$, and the left end is fully attached to the fixed cylinder, the control point is A, attached to the structure and moving in time with $A(0) = (0.6, 0.2)$. The setting is intentionally non-symmetric (see [**?**]) to prevent the dependence of the onset of any possible oscillation on the precision of the computation. The mesh used for the computations is shown in the the Figure **??**.



level	#el	#dof
0	62	1338
1	248	5032
2	992	19488
3	3968	76672

Figure 5. Coarse mesh with number of degrees of freedom for refined levels

5.2.1. SIMPLEX Results The FSI-Opt computations are done on the same the mesh and its refinement levels, as used for the FSI benchmark in [**?**]. The reference value of lift coefficient in case of stationary FSI calculation is $7.6e - 1$ (see [**?**, **?**] for more details). When the flow is introduced or injected with the constant velocity $V_2 = 10m/s$, from below the lift on the beam obviously increases, see Figure **??**, which shows that it is the wrong direction to inject flow. For the case of suction, the flow with same constant velocity $V_2 = 10m/s$ from below produces negative values of lift in increasing order, see Figure **??**. If the flow is injected from top and extracted from bottom with the same velocities $V_1 = V_2 = 10m/s$ without considering the SIMPLEX method, then the resulting lift coefficient on the beam seems to be quite smeared, irregular and hard to predict or conclude what could be best coordinate/direction which can give minimum lift. The vector magnitude of the flow behavior is shown in Figure **??**.

From this it is clear that the $V_1 = V_2 > 10m/s$ is not a good direction to select coordinates of SIMPLEX. Hence it became clear that for the implementation of SIMPLEX method the coordinates of the triangle should be between $[0, 10]$. For the numerical simulation the coordinates $(0, -3)$, $(3, 3)$

()
DOI: 10.1002/

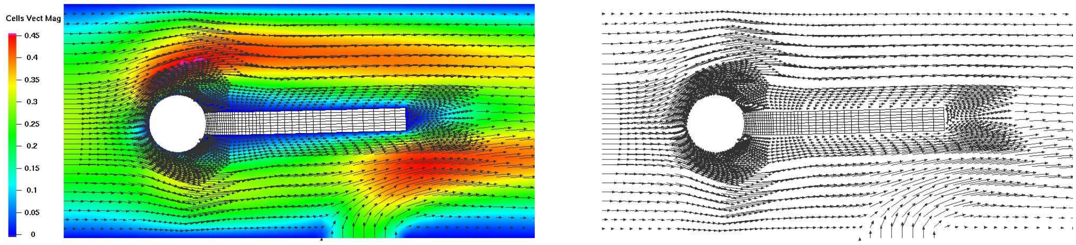


Figure 6. No SIMPLEX: Flow vector magnitude (Injection) level 3

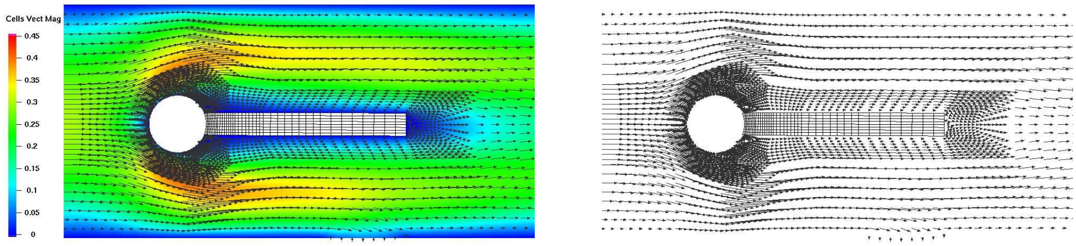


Figure 7. No SIMPLEX: Flow vector magnitude (suction) level 3.

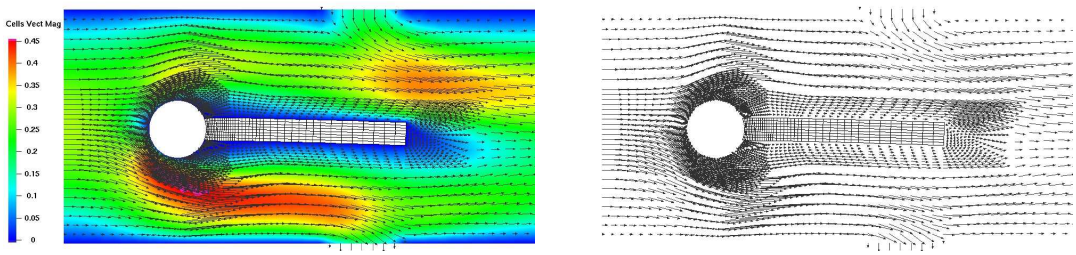


Figure 8. No SIMPLEX: Flow vector magnitude (Injection and suction) level 3

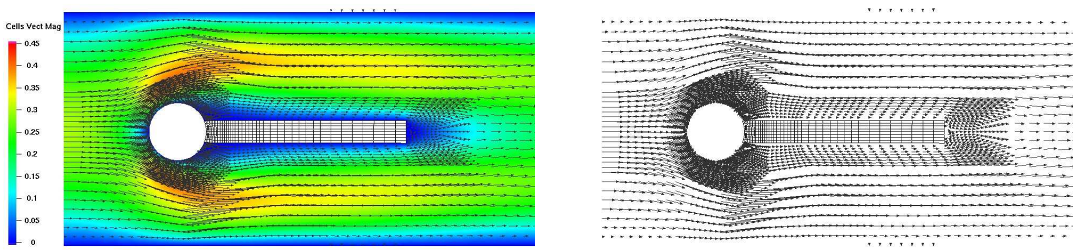
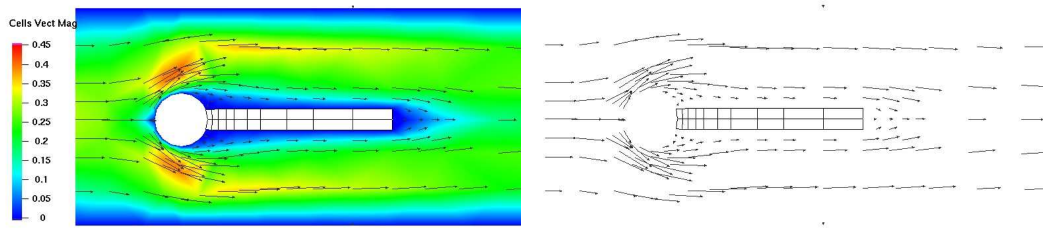


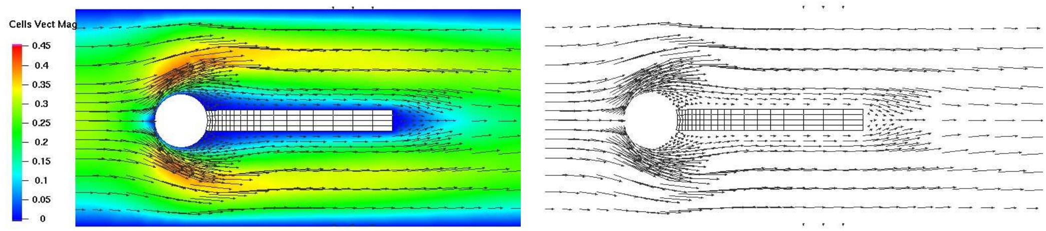
Figure 9. SIMPLEX: Flow vector magnitude (Injection and suction) level 3

and $(-3, 3)$ for a two variable Nelder-Mead algorithm are used. For this case, if the simplex method is in place lift coefficient goes to almost zero, as shown in Figure **??** and in result the beam became almost static. Optimal points are then the (V_1, V_2) values which result in minimum lift on the beam depending on the parameter α. As α decreases the reduction of the lift on the beam is visible and the optimal point $(1.06e+0, 1.08+1)$ is for mesh level 1, $(1.04e+0, 1.05e+01)$ is for mesh level 2 and $(1.04e+0, 1.05e+01)$ is for mesh level 3. Results are shown in Figures **??**, for mesh levels 1, 2 and 3 in respective order, which show the optimal velocity values V_1 and V_2 providing the minimum lift on the beam as compared with the FSI1 benchmark reference lift values which is $7.6e-1$.

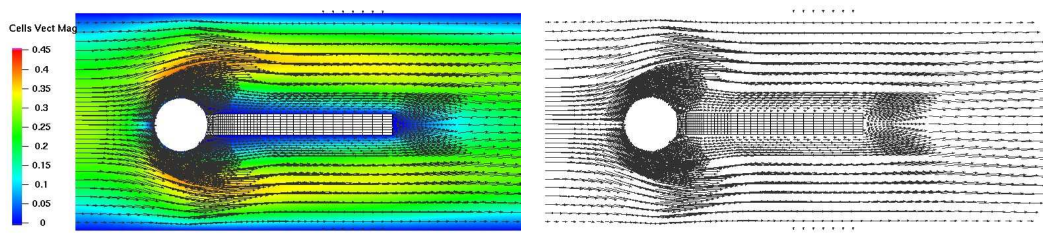
DOI: 10.1002/



α		level1, dof=5032	
	iter	optimal values(V_1, V_2)	lift
$1e+0$	57	$(3.74e-1, 3.88e-1)$	$8.1904e-1$
$1e-2$	60	$(1.04e+0, 1.06e+0)$	$2.2684e-2$
$1e-4$	73	$(1.06e+0, 1.08e+1)$	$2.3092e-4$
$1e-6$	81	$(1.06e+0, 1.08e+1)$	$2.3096e-6$



α		level2, dof=19488	
	iter	optimal values(V_1, V_2)	lift
$1e+0$	59	$(3.66e-1, 3.79e-1)$	$7.8497e-1$
$1e-2$	59	$(1.02e+0, 1.04e+0)$	$2.1755e-2$
$1e-4$	71	$(1.04e+0, 1.05e+01)$	$2.2147e-4$
$1e-6$	86	$(1.04e+0, 1.05e+01)$	$2.2151e-6$



α		level3, dof=76672	
	iter	optimal values(V_1, V_2)	lift
$1e+0$	67	$(3.66e-1, 3.79e-1)$	$7.87e-1$
$1e-2$	77	$(1.02e+0, 1.06e+0)$	$1.97e-2$
$1e-4$	100	$(1.04e+0, 1.06e+0)$	$2.03e-4$
$1e-6$	100	$(1.04e+0, 1.06e+0)$	$1.3372e-6$

Figure 10. No displacement is visible of the beam due to optimal boundary flow control.

In Figure **??**, it is easily seen that the beam is not displaced i.e. no lift on the beam is observed due to the boundary control, and results are shown for three different mesh refinement levels.The lift coefficient on the beam with changing α parameter is given in the corresponding tables in the Figure **??**. Also, for higher mesh refinement levels more iterations are required and the result (lift$\approx$ 0) is better compare to the result for the level 1 and level 2.

()
DOI: 10.1002/

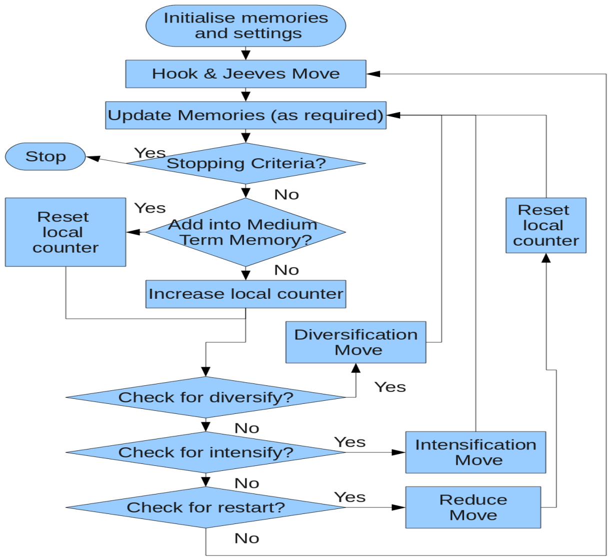


Figure 11. MOTS2 flow diagram

5.3. MOTS2 description

Tabu Search belongs to the category of stochastic search optimizers and is based in the original ([**?**]) and the Multi-Objective ([**?**]) version. The current implementation is based on It searches throughout the design space in a stochastic way and it avoids recently visited design points, so as to guarantee more exploitation of the unknown design space. In fact, local search [**?**] is combined with stochastic elements. Three different hierarchical memories are used to assist critical decisions during the optimization process. It also keeps track of certain statistics during the process, which direct the search according to the discovered landscape of the design space. In addition, the optimizer employs a mechanism for local and global search. The statistics detect design points around the current search point, within relatively short distance, whereas the search mechanisms attempt to discover good design points in the entire design space. Consequently, the functionality of MOTS2, as depicted in Figure **??**, results in better performance throughout the optimization process. The configuration settings are listed in Table **??**.

The search is guided by the current base point and collective memory banks. Around the base point, adjacent candidate design points are investigated and evaluated. Then, the corresponding objective values are sorted according to domination criteria of multi-objective optimization [**?**] and the following base point is resolved. The previous base point and all the recently generated points are inserted into the appropriate memory banks. Aggregated information will be used in future steps, when certain conditions are triggered. This procedure keeps repeating until stopping criteria are met. Depending on the nature of the application these are usually the elapsed time, the number of evaluations, the number of of consecutive failures to find a better point, number of iterations or a combination of them. Herein, the core is the Update Memories, Hooke and Jeeves-, Intensify- and Reduce-Move.

The following parts take place in every iteration as follows:

- The *Hooke and Jeeves Move* is the most important as it occurs on every iteration: Starting from the base point, a couple of valid and non-tabu points are generated by combining the current base point and the current search step. Some of the recently created points are evaluated (by sampling) and added into the appropriate memory banks. These points are within the close vicinity of the base point and this is the local search phase of the optimizer.

()
DOI: 10.1002/

Table I. MOTS2 configuration

performing diversification move after # iterations	20
performing intensification move after # iterations	10
performing reduction move after # iterations	45
initial search step	0.1
search step retain factor	0.65
# of random samples	6
# of variables	2
# of objectives	4
max objective function evaluations	14000
# of regions in Long Term Memory	4
Short Term Memory size	20
maximum improvements	200
maximum duplicates	30

- Then comes the *Pattern Move*. This is just an enhancement of the Hooke and Jeeves Move where the next base point will be quickly resolved. Whenever Hooke and Jeeves Move takes place for second time, the following base point is generated by combining information from the last two base points.
- *Update Memories*: At the end of every iteration the newly resolved base point is inserted into the base memory bank, history bank and pareto front bank (should it fulfil their corresponding conditions).

The aforementioned moves are performed several times until certain conditions are met, which will trigger one of the following moves. During the execution of the algorithm, the memory banks are enriched with information which will be exploited later on. Hence, a zero-knowledge search starts and the optimizer learns through information about the intrinsic features of the design space from the banks iteration-by-iteration. According to the principles of artificial intelligence, this is the best method of a heuristic search [**?**].

The following moves are carried out when specific numbers of iterations occur:

- *Intensify Move*: By definition, contrary to single-objective optimization, during multi-objective optimization several points form the trade-off. However, during every iteration, only one of them might be the base point. Therefore, the rest of the points that dominate the current trade-off, but have not been selected as base points, are stored into the intensification memory. Whenever the search cannot discover any new nor non-tabu point, another point from the back-up bank is selected randomly as the next base point. Hence, the search returns back to the most promising points discovered so far and picks-up the search thereafter. This is the most frequent performed move.
- *Diversify Move*: Instead of finding a better point, within a short range, a new non-tabu point is randomly generated from least explored region of the design space. This is the global search phase of the optimizer and its frequency depends on the problem.
- *Restart Move*: Whenever the search fails to discover a new good point with the current search step, a new base point is randomly resolved and the search step is refined.

Regarding the FSI optimization, the combination of the range of V_1 and V_2 defines the design, which belongs to the design space, $\mathbb{R}^2$. In an analogous way, the objectives belong to a different space, namely objective space, $\mathbb{R}^4$. Every time a single point of the design space maps to a point of the objective space. The aim of the optimizer is to try different combinations of these two variables on the given simulation model and detect which areas express the best performance, defined by the objectives. After successfully iterating through the optimization phase, the best discovered trade-off is presented to the designer to choose the final design. This is known as the decision phase. The time required in order to establish the variables-to-objectives mapping is the evaluation time of the given variables via the simulation. This is the most critical part of the optimization process, as it affects the overall execution time of the whole optimization. In fact, the overall execution time can be expressed as the summation of multiples of the execution time required for a single evaluation and the overhead of the optimizer, which is practically negligible. In this case, each design evaluation can take up to 1 minute.

5.4. MOTS2 Results

Earlier studies, see **??**, attempted to optimize the case described in 5.2 by using one composite objective function **-** a weighted sum of two objectives **-** by employing a genuine single-objective optimizer. Contrary, MOTS2 deals directly with native multi-objective optimization problems. By including more objectives the dimensionality (and hence the complexity) increases considerably, which necessitates the use of a totally new algorithm. MOTS2 has been verified and validated [**?**] and can handle both constrained and unconstrained cases. By introducing more objective functions, the complexity of the system increases. Therefore, a larger variable range will be required in order to explore the design space sufficiently and not to induce any bias.

The main aim is to minimize the motion features of the beam by controlling the top and bottom flow. In particular, this means to minimize lift, drag, horizontal and vertical displacements of the point A at the tail of the beam, at the same time. The application involves 2 control variables that correspond to 4 objectives. Various combinations of these two variables are evaluated through the FSI simulation, as explained above. Internally, the optimizer ranks these objectives for domination, generates new designs and the results are presented below. The set of the 2 variables and 4 respective objectives is called a tuple. The objectives, in the order of appearance in the following Figures are drag, lift, horizontal displacement (u_x) and vertical displacement (u_y) of the beam. The target is to minimize all of the aforementioned objectives, as this brings stability to the system. Both of the variables range between -50 and 50 of $\mathbb{R}$. The optimization process generated and evaluated 14000 different design combinations. Among them, 3600 were feasible. Moreover, 1200 tuples dominate the objective space, with respect to the aforementioned objectives. This is indicative of the complexity of the system where the number of the objectives is larger than the number of variables.

The optimum discovered tuples are depicted in Figures **??** and **??**. These are the scatter plots and the parallel coordinates projections [**?**], respectively. The former informs the user of the pairwise relations between each of the components of the optimization process. The latter is an alternative way to represent multivariate data in 2D. In fact, it is a transformation of an N-dimensional space into an assembly of N mutually and individually scaled parallel axes. Any point of the original N-dimensional space is represented by a set of lines connecting parallel axes and intersecting them in the values of original coordinates. In this projection each line that connects one point from each axis represents one tuple. The top of each axis corresponds to the maximum value. Likewise the minimum value is at the bottom. Both Figures are particularly useful in order to identify relations and interactions between the variables and objectives. The most interesting types or relations are the correlations between objectives, and how variables' variation affects one or more objectives. Moreover, the results form 3 different clusters, which will be explained below.

A scatter plot matrix is a compact way to represent all of the participating components in a pairwise way of a NxN matrix. The user is informed about each component individually and how each component interacts with the remaining ones. It is important to notice that the matrix is symmetric. The matrix could be split into 4 sub-matrices; a 2x2 on the top-left, a 2x4 top-right, a 4x2 bottom-left and a 4x4 bottom-right. These represent the relations between variables vs variables, variables vs objectives, objectives vs variables and objectives vs objectives, respectively. The elements in the main diagonal represent the histogram of each variable and objective. This information will be combined with the search for patterns in the parallel coordinates plane. The two pictures for the top-left part depict the optimum samples of the design space. By combining the histograms, the user is informed about which areas the optimiser focused on. The remaining of the first two columns and first two rows depict the relationship between each variable and the respective objectives. The big sub-matrix bottom-right shows the relationship between the objectives.

()
DOI: 10.1002/

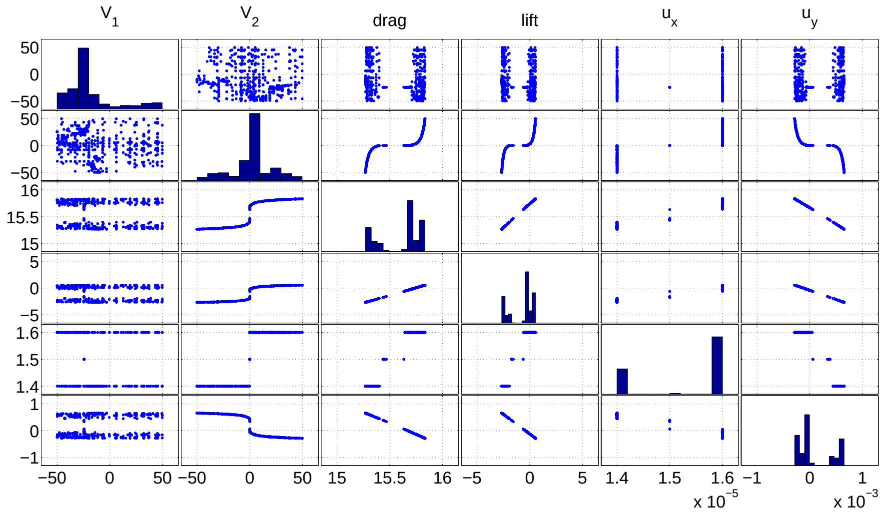


Figure 12. MOTS2 results on 4-objectives optimization scatter plot matrix. The order of lines and columns represents V_1, V_2, drag, lift, u_x, u_y respectively. Diagonal elements represent the corresponding histogram for each component.

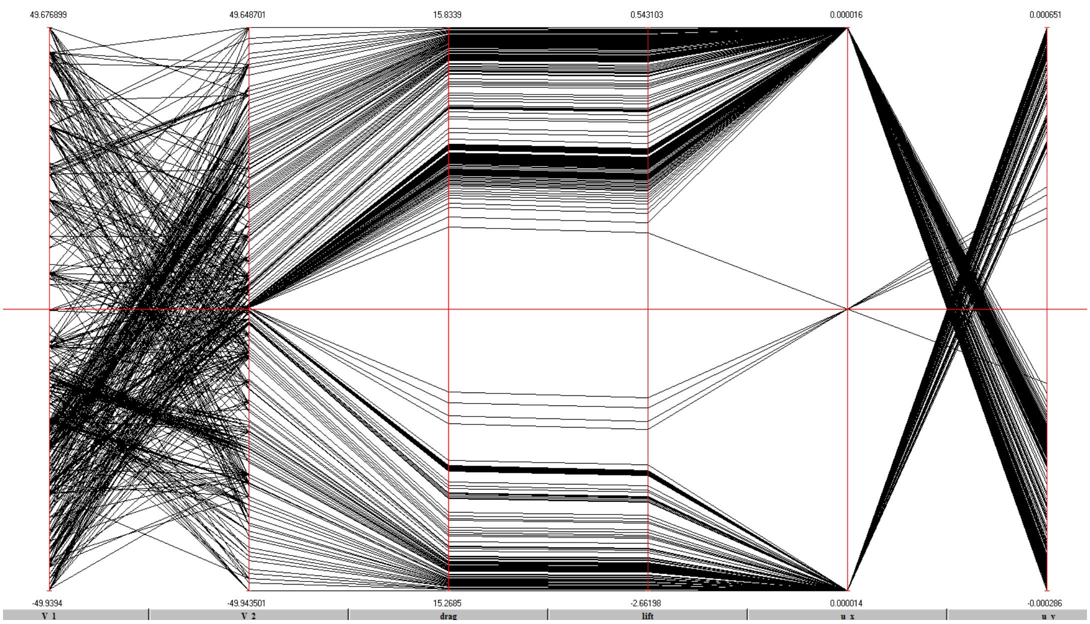


Figure 13. Full Data-Set

Therefore, the optimizer focuses the search on the regions of negative V_1 and positive V_2, near the origin. In addition, there is a clear relationship between V_2 and lift objective. Obviously, this trend extends for the rest of the objectives. Each variable forms 2 distinct sets between drag, lift and u_y, whereas an additional smaller set is formed for u_x. Moreover, near the most sampled area there are two parts, which extent towards the origin. It seems obvious that drag and lift are correlated linearly. In addition u_y is also linearly correlated with the aforementioned objective in a reciprocal way. In terms of multi-objective optimization, these objectives live in harmony and one of the can sufficiently describe the case. This is also proven in parallel coordinates, below. Both between the variables and the remaining of the objectives, u_x has 3 distinct regions. Finally, lift and u_x conflict each other.

The rest of the analysis is based on the parallel coordinates projection with some references to the scatter plot matrix. For ease of analysis, the Full Data-Set, presented in Figure **??**, breaks down into Figures **??-??** by combining positive and negative values of V_1 and V_2, and excluding the designs for $u_x = 1.5e - 5$. For completeness a snapshot of the optimal designs that include the middle values of u_x are depicted in Figure **??**, where V_2 is almost zero and V_1 takes almost the same value. Fixing V_2 and searching the design space for V_1 will be part of the future work. Since three objectives are linearly correlated and the last one takes 3 distinct values, the scatter plot between each variable and only one objective is present in these Figures. First of all, for the variables axes (first two axes), it is obvious that certain vertices gather more edges, which means that these points are more important and the optimization process discovered the best objectives around them. It is also useful to know that the design space was searched equally well between the range of -50 and 50, without any bias. The search step started from 0.1 and was subject to successive step reductions up to $4.97745e - 06$ step size. Starting from a big step the search narrows down to the most promising areas, which present the best performance and where the search is refined. The remaining 4 axes represent the values of the 4 separate objectives.As shown in Figure **??**, even searching throughout the design space does not achieve objective values close to zero. Clearly, u_x presents discrete clusters of values, which means that certain performance lies within certain regions of the design space. This means that there are 3 different operating modes and the number of edges on each level of the axis indicates the preferable areas; for the middle value only a few designs exist which means that under specific settings the behavior of the FSI model changes. Drag, lift and u_y present a wide range with a few thicker areas, which present areas of high robustness. Thus, it is more interesting to analyse the interactions of the wider objectives as the trade-off changes.

Regarding the bottom velocity (V_2), three clusters are formed, based on u_x axis, depicted in Figure **??**. The two big clusters split very close to 0.0 and they do not mix. By observing the patterns of the sign of V_2, it it is positive, then $u_x = 1.6e - 5$ (Figures **??** and **??**). Similarly, when V_2 is positive the lower cluster of u_x is activated (Figures **??** and **??**). In conjunction with Figure **??**, V_2 acts like a switch for the system, irrespectively of the values of V_1. Moreover, Figure **??** depicts a clear relationship between V_2 and drag objective, which extends of course with the other linearly correlated objectives. By selecting a small region while both variables are positive (Figure **??**), it seems like lift values overlap/mix with the corresponding lift of negative V_1. In fact the value of V_1 does not have a great impact, while V_2 remains constant. Therefore, V_2 is the most important variable, whereas V_1 could be considered as a performance offset for controlling the exact values of the objectives.

The first two objectives concern drag and lift, respectively. Unlike aerodynamic cases, the objectives of lift and drag increase and decrease at the same time. This means that the objectives live in harmony. This is confirmed, for example, in Figure **??** for the plot at the position $4, 3$. In other words, this is an indication that one of them could be omitted to reduce the complexity of the optimization. Analyzing drag is equivalent to lift. The only difference is the range (length in the picture) of lift and drag. This is from 15.269 to 15.834 and -2.662 to 0.543, respectively. One could reduce the dimensionality by excluding one of these objectives. The areas where the lines are thicker are areas of more robust designs; The variation of the designs that map to the objective space presents stable behavior. Ideally the variation should be zero, but any quantity close to zero is satisfactory. Since lift has larger range, this leaves more options for further improvement and gives better control.

The third objective, u_x, has 3 distinct values $1.4e - 5$, $1.5e - 5$ and $1.6e - 5$. On one hand these could be treated as 3 different operating modes/levels for the physical application. However, their relative difference is extremely low. The population of edges for the middle range is significantly low, compared to the other two ends. The latter means that the designs which achieve those specific objectives should not be selected at the (final) decision making phase. Hence, it seems that this objective is less significant and makes the optimization search more complex, for no particular

DOI: 10.1002/

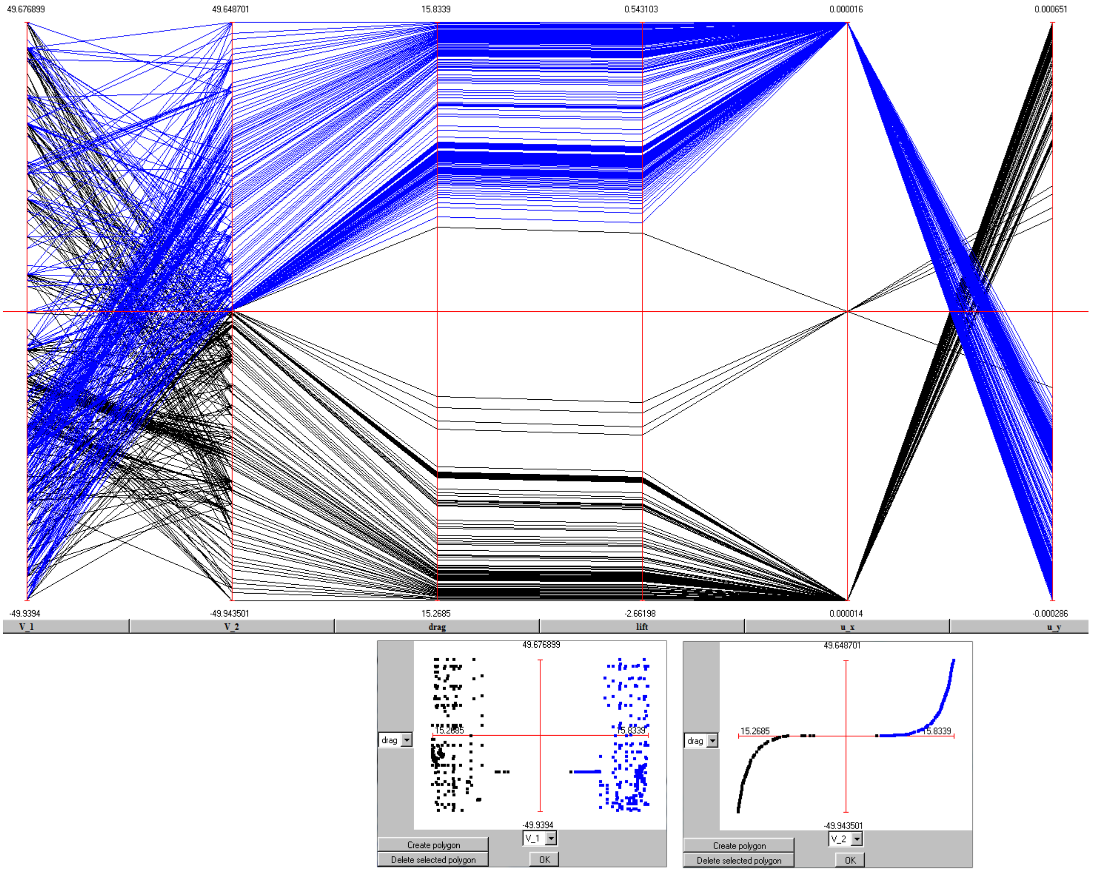


Figure 14. Demonstrating linear correlation

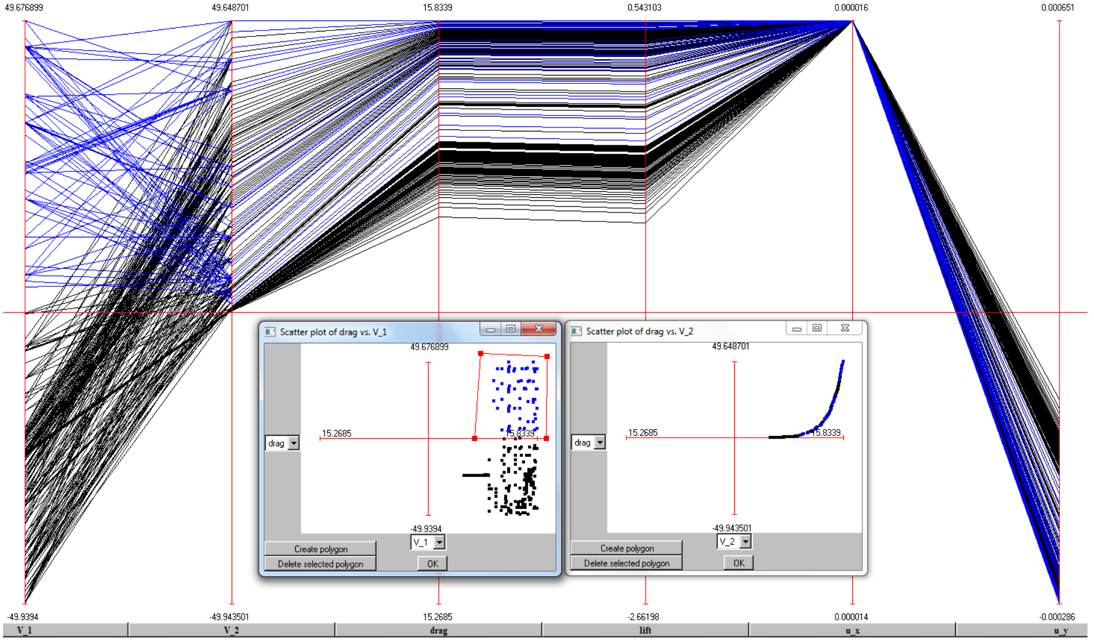


Figure 15. Investigating the behavior of V_1

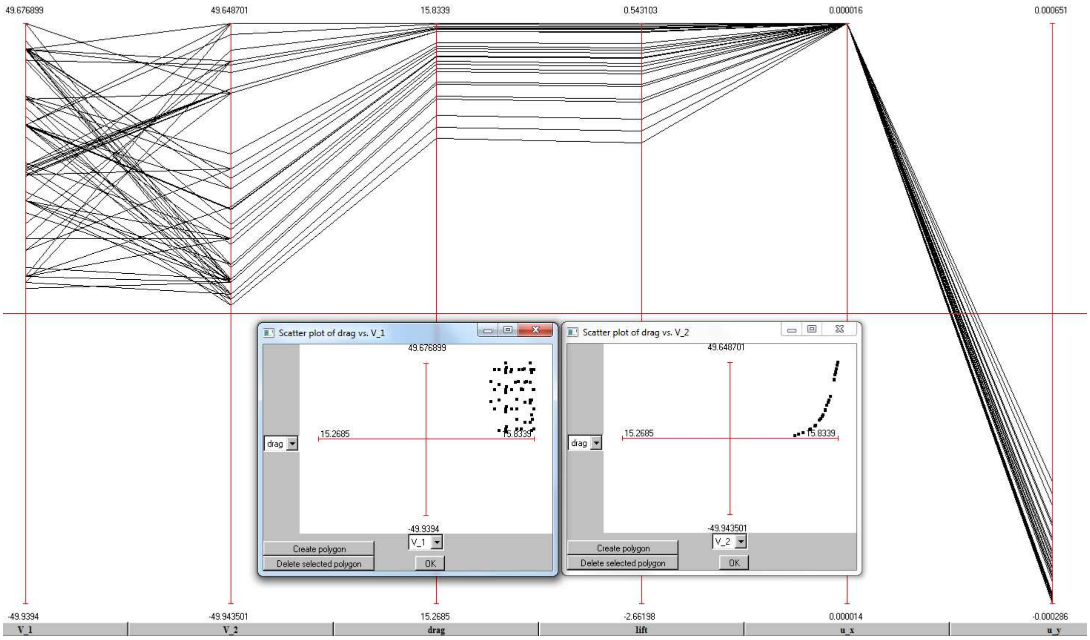


Figure 16. Selecting positive V_1 and V_2

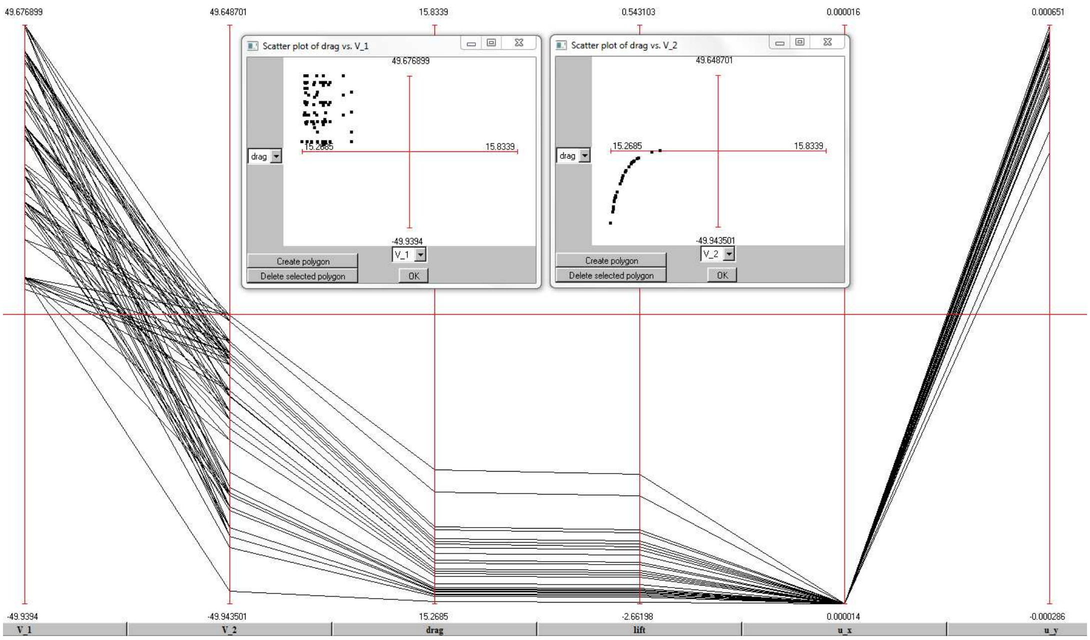


Figure 17. Selecting positive V_1 and negative V_2

information gain. In other words, the complexity of adding a third objective for the amount of information the user could exploit does not pay off. Therefore it could also be excluded.

The last axis represents u_y. Obviously, the designs gather around the extrema of the axes and form three distinct sets. The middle region contains values which correspond to specific designs and can be disregarded, as discussed above. The remaining two clusters reveal the conflicting nature of the problem. It can be shown that increasing lift results in decrease of u_y and vice-versa. Again,

()
DOI: 10.1002/

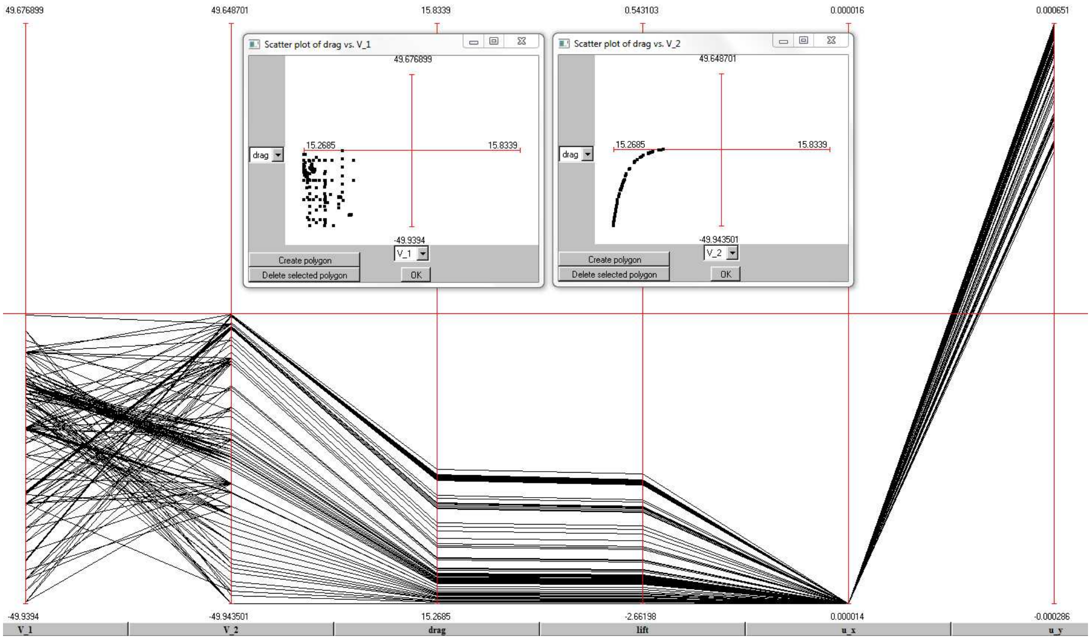


Figure 18. Selecting negative V_1 and negative V_2

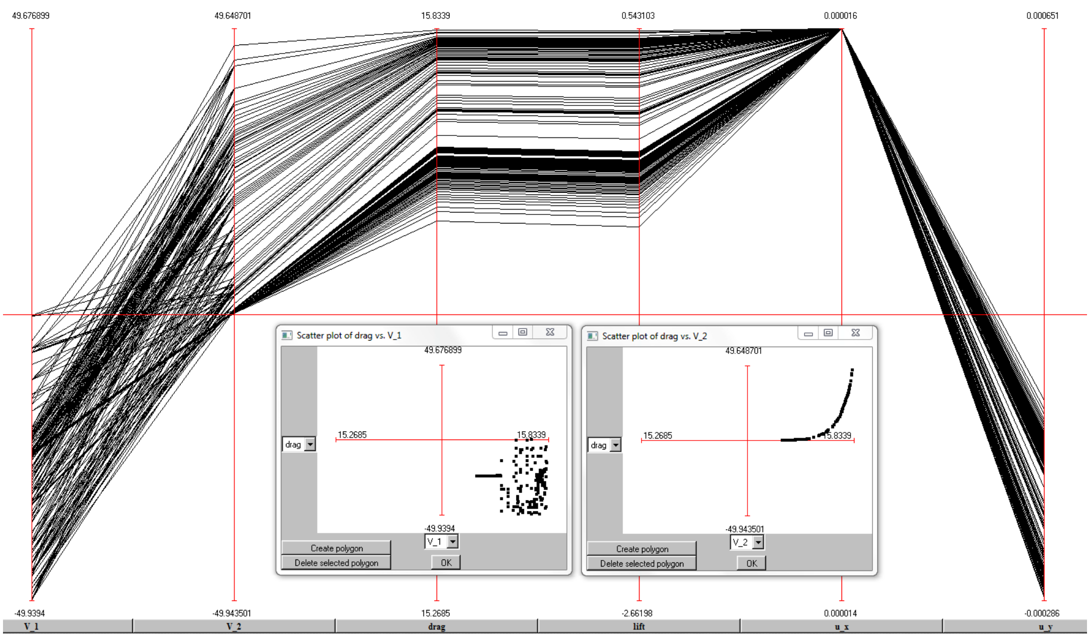


Figure 19. Selecting negative V_1 and positive V_2

certain areas contain more edges (higher line concentration) which are related to the robustness of the simulated system.

By observing the element 1,3 of Figure **??**, it seems interesting to investigate further the prominent points that direct towards zero. The linking with parallel coordinates plane is depicted in Figure **??**. For a certain range of drag values V_1 and V_2 take almost the same values forming a subset of size 155, which corresponds to 18% of the whole trade-off. However, some of the designs correspond to

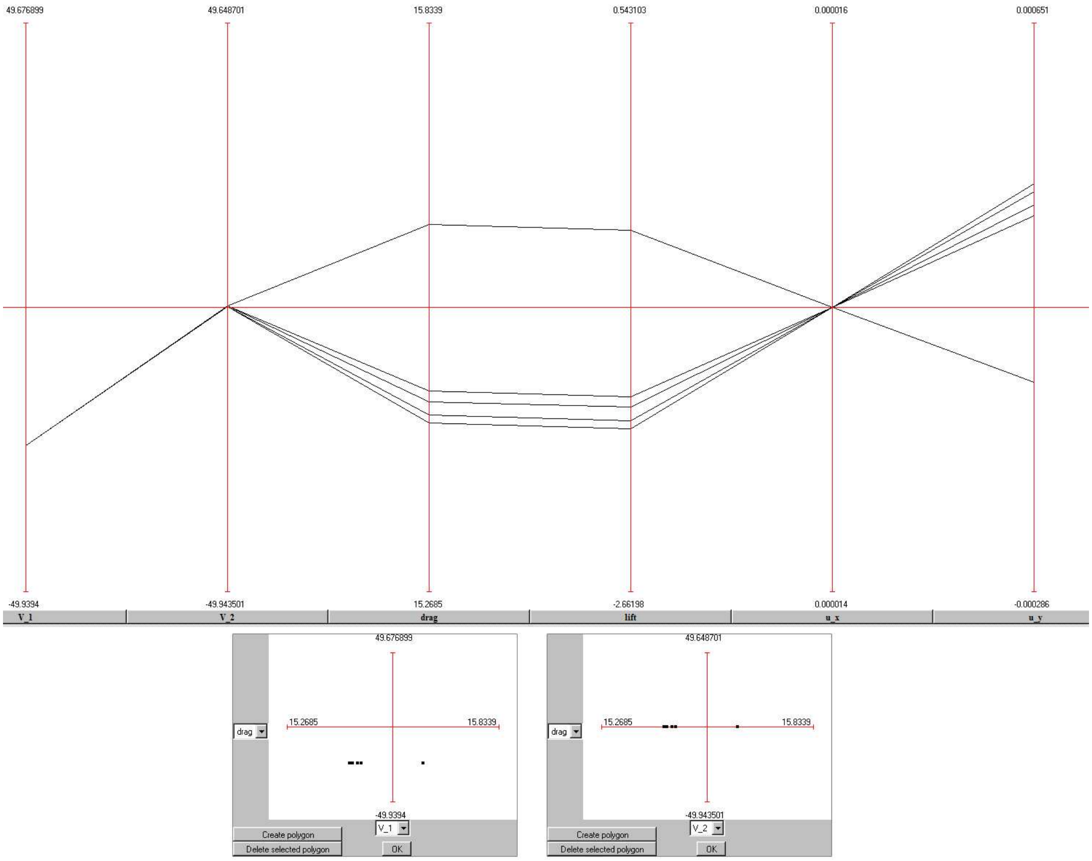


Figure 20. Selecting the middle value of u_x

$u_x = 1.5e - 5$. This lead to the discovery of the most dense region of the design space, depicted in Figure **??**, which was initially pointed out by Figure **??** in the elements 1,1 and 2,2. These design configurations will be investigated in future studies.

In the end, it seems that the considered case could only involve lift and u_x. The drag objective is redundant for the optimization process as it directly follows the same trends as lift, whereas u_x is reciprocal to lift. Possibly, these could be used for future secondary-deductions. This would also reduce the complexity and speed-up the design cycle. Coming up with more conflicting objectives will be part of the future work. At any case, adding an extra objective in any optimization problem, might result in totally different results compared to fewer or more objectives. Finally, for the decision making phase, a design from the isolated regions in Figure **??**, seems very promising because they present nearly similar performance within a close distance.

5.5. Selecting the Robust Design

The Multi-Objective Design optimization aims to assist the user to select the best design(s) for the decision making phase. Following the principles of robustness analysis, the performance of the best designs should lie within a close region. This means that most of the designs should not deviate significantly from the target values. Furthermore, the performance should be as robust as possible. So, in this study, the selected designs present no more than 5% variability for the defined objectives. As identified earlier, only one objective is sufficient for assessing the performance. Moreover, one of them (u_x) has 3 distinct values, which expand in 3 wider sets of the lift and u_y axes. By examining the population of the PF of u_x with respect to lift and u_y, the ratio of the range of the values of the objectives over the number of individuals within that range indicates that it is highly probable to find

()
DOI: 10.1002/

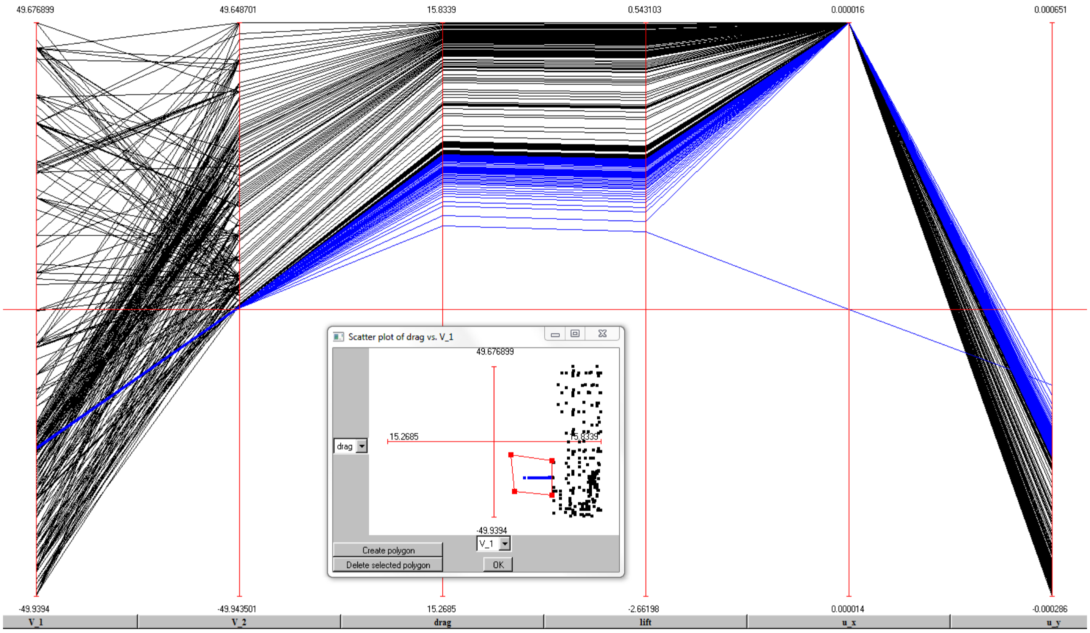


Figure 21. Exploring the prominent points of the scatter plot drag vs V_1

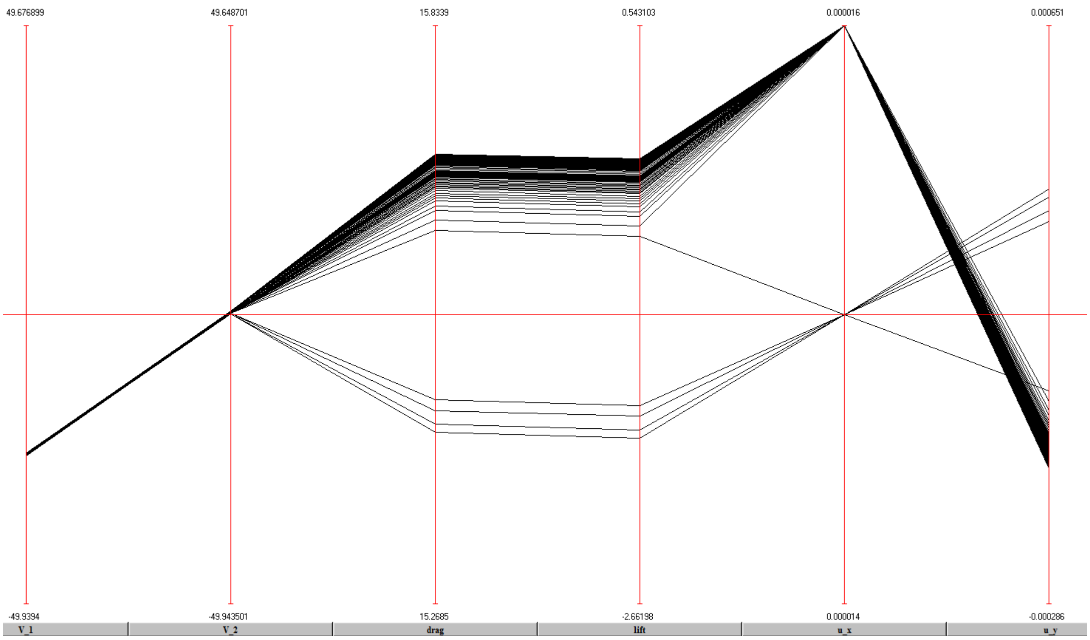
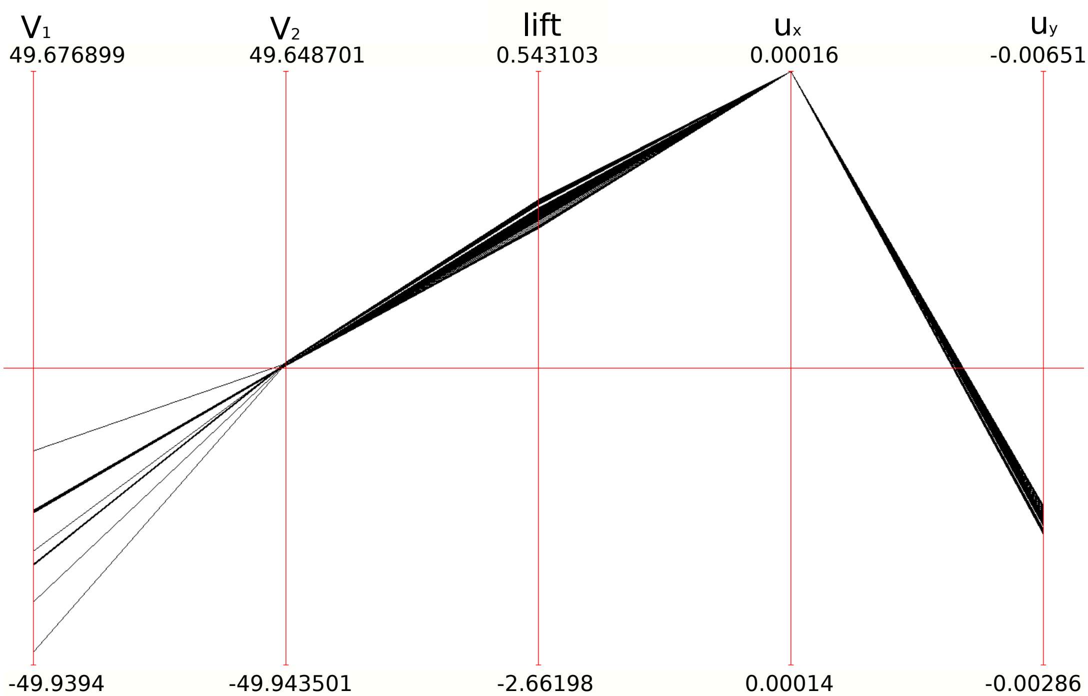


Figure 22. The most dense region for almost fixed values of V_1 and V_2

several designs within the target performance for u_x=$1.6e-5$. This holds both for lift and u_y and was also confirmed computationally. So, the search should focus on the intervals of the objective space where the behavior is robust with respect to lift and u_y. This is the cluster which includes lower part of u_y and is presented in Figure **??**b.

Since the target area of performance has been identified, the corresponding set of designs should be resolved. Following the same procedure, designs whose performance is nearly stable should be

()
DOI: 10.1002/



(a) Robust Design within PF

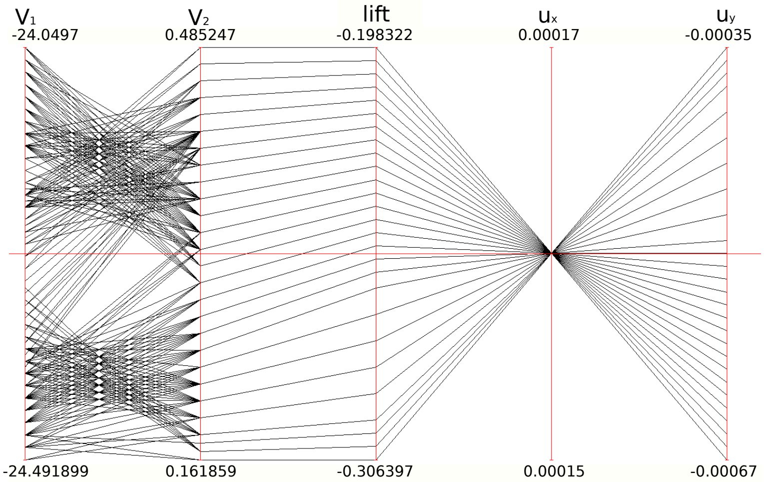
Figure 23. Selection of Robust Designs

selected. Isolating 5% of the range of u_y, the highest concentration of designs was found between -0.83 and -0.34. Within that range V_1 forms two small clusters $[-24.0, -24.5]$, $[-33.0, -33.3]$ and V_2 spans over $[0.15, 0.82]$. Likewise, lift's 5% most populated region is $[-0.31, -0.14]$, where V_1 spans over $[-24.5, -24.0]$, $[-33.27, -33.0]$. By classifying the designs, the region where V_2 belongs to $[0.16, 0.85]$ includes 29% of the designs, as depicted in Figure **??**b. This confirms that V_2 is the most important variable. However, there is a second area of robust designs where the corresponding performance is slightly different, as depicted in the lower part of Figure **??**. This cluster emerged by investigating the 3rd picture of Figure **??**. More specifically, the search focused on the part of the right cluster that extends towards the origin, which contains more than 10% of the population of designs. These are proposed as robust designs and a compromise for the application.

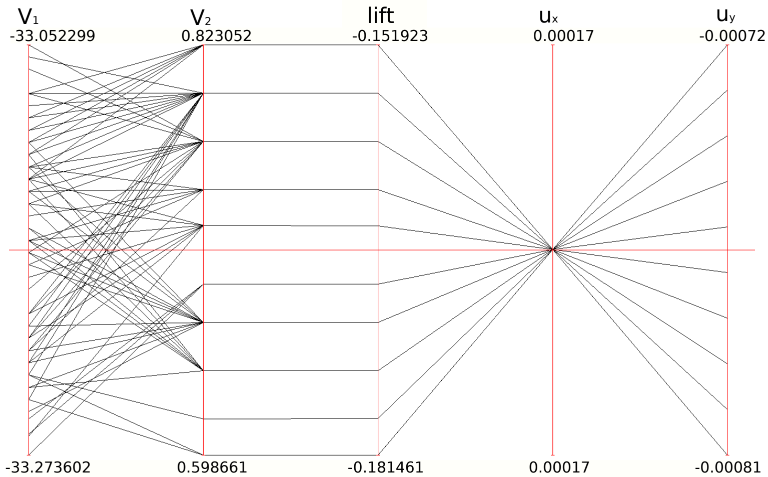
It is important to mention the inherent feature of line-to-point projection between N-dimensional planes and parallel coordinates, which is presented in Figure **??**. The correlation of bottom robust designs against the objectives is a linear curve. However, this is not true for the top robust designs, which follow a parabolic trend. In fact, the generated scatter plots between the most significant variable and the objectives demonstrate that V_2 behaves as a switch.

Ultimately, the comparison between the results of multi- and single-objective optimization is presented in Figure **??**. One design from the top robust designs, which corresponds to $(-33.1711, 0.743855)$ design point, represents the performance of MOTS2 against the optimum design described in **??**. Level 2 comparisons seem nearly identical. The only difference is at the front side of S_1, where the region of the highest vector magnitude is larger and the produced wake expands along the beam. However, by employing higher resolution at Level 3, the flow behavior is similar to Figure **??**. This is expected and justifies the choice of employing MOTS2.

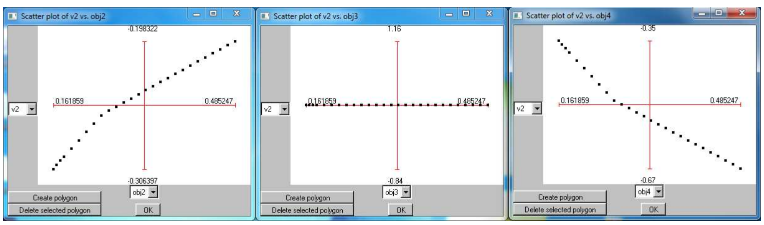
()
DOI: 10.1002/



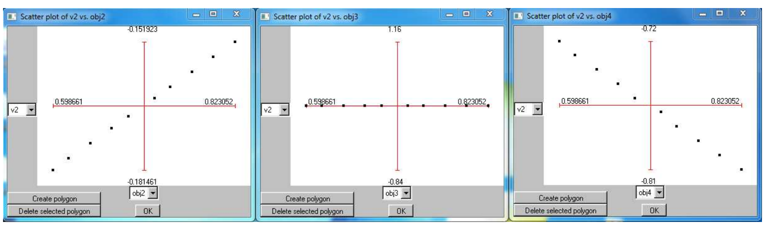
(a) Top robust designs parallel coordinates projection



(b) Bottom robust designs parallel coordinates projection



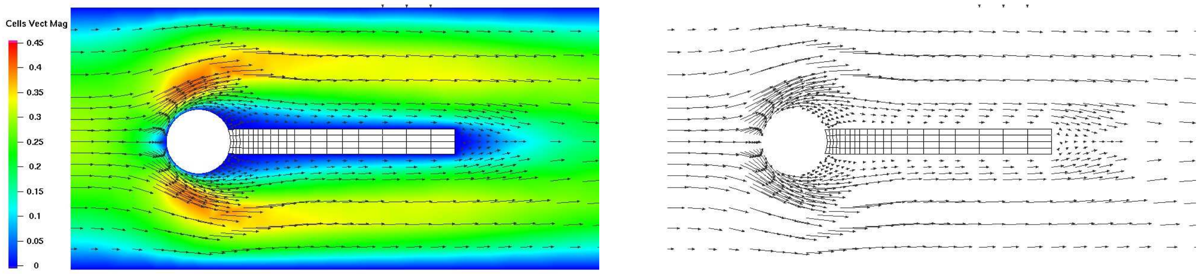
(c) Top robust designs scatter plot



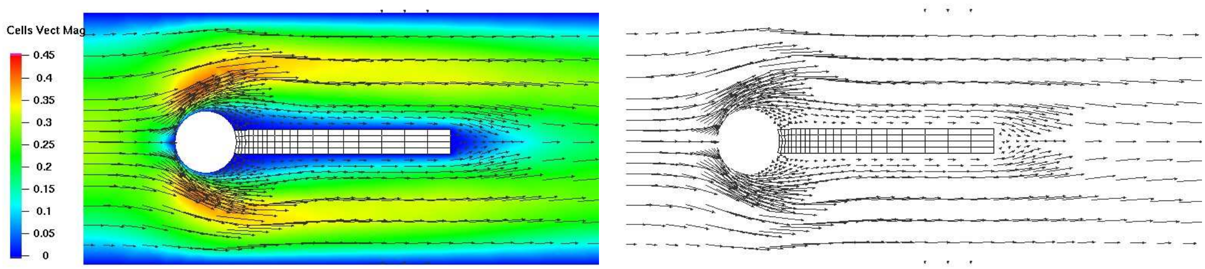
(d) Bottom robust designs scatter plot

Figure 24. Focusing on robust designs

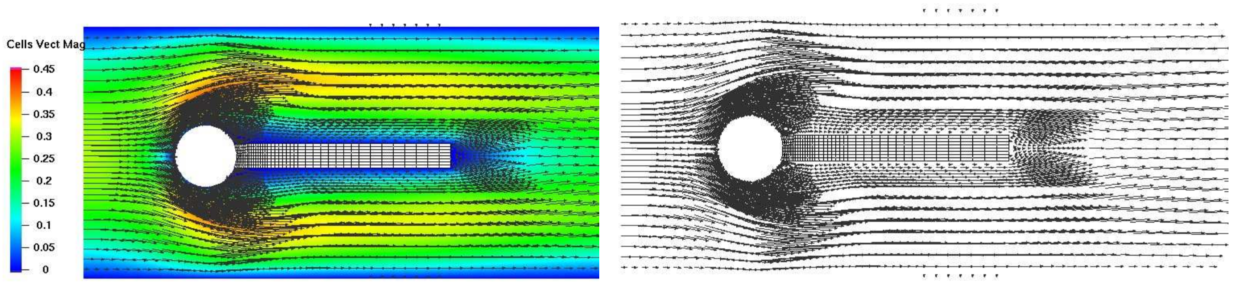
()
DOI: 10.1002/



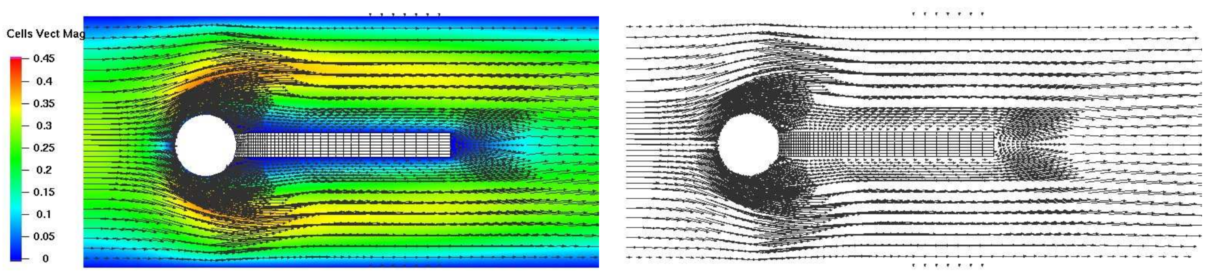
(a) MOTS2 Level 2



(b) SIMPLEX Level 2



(c) MOTS2 Level 3



(d) SIMPLEX Level 3

Figure 25. Comparing the flow between MOTS2 Robust Design and SIMPLEX optimum design

()
DOI: 10.1002/

6. CONCLUSION

This study introduced a new methodology for tackling CFD problems of engineering interest. More specifically, the integration and the application of a multi-objective optimizer (MOTS2) along with a FSI package for 2D problems were described. Further details for both parts were presented, followed by the results. Optimal and robust designs were identified for single and multi-objective optimization cases, respectively. For the latter, the originally formulated 4-objectives problem turns out to be a single-objective case. It was shown that the lift and drag objectives live in harmony, while u_x is reciprocal to them. So, one of them should only be considered so as to reduce the complexity of the problem. In addition, two sets of compromise and robust designs were suggested with less than 5% variation of the overall performance. Nonetheless, the performance of robust design is not the same, compared to the optimal design identified by SIMPLEX. Since u_x corresponds to near zero value for V_2 for the middle-range cluster, it is worthwhile to further investigate this region of the design space. Finally, it seems promising to explore the design space near the areas where the robust designs reside.

Future work will investigate the identified regions of interest. Also the simulation model and the optimization process will be expanded in 3D and more search strategies, respectively. The former necessitates the porting of the FSI package in 3D and more performance metrics should be defined. More variables and objectives should involved for the latter. This compilation has the potential to be applied on several real world problems such as cardiovascular diseases (in health sciences) and aero-elasticity (aerodynamics).

ACKNOWLEDGEMENTS

This work was supported by the German Research Association (DFG) through the collaborative research center SFB/TRR 30. The third author would like to thank DAAD Research Grants for Doctoral Candidates and Young Academics and Scientists, CleanSky and Bodossaki Foundation.

REFERENCES

1. Deb, K. Multiobjective optimization using evolutionary algorithms. 2001; Wiley, New York Chichester.
2. Alexandrov, N. Editorial - multidisciplinary design optimization. Optimization and Engineering 2005; 6(1):5–7.
3. Turek, S. Hron, J. Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow, Fluid-Structure Interaction: Modelling, Simulation, Optimisation, Springer, 2006; Ed: Bungartz, H.-J. and Schäfer, M., Lecture Notes in Computational Science and Engineering, 53: 371-385.
4. Schäfer, M. Sternel, D. C. Becker, G. Pironkov, P. Efficient numerical simulation and optimization of fluid-structure interaction, Fluid-Structure Interaction II: Modelling, Simulation, Optimisation, Springer, 2010; Ed: Bungartz, H.-J. and Mehl, M. and Schäfer, M., Lecture Notes in Computational Science and Engineering, 73:131-158.
5. Hojjat, M. Stavropoulou, E. Gallinger, T. Israel, U. Wüchner, R. Bletzinger, K.-U., fluid-structure interaction in the context of shape optimization and computational wind engineering, Fluid-Structure Interaction II: Modelling, Simulation, Optimisation, Springer, 2010; Ed: Bungartz, H.-J. and Mehl, M. and Schäfer, M., Lecture Notes in Computational Science and Engineering, 73: 351-381.
6. Hron, J. Turek, S. A monolithic FEM/Multigrid solver for ALE formulation of fluid-structure interaction with application in biomechanics. Lecture Notes in Computational Science and Engineering, 2006; 53: 146–170.
7. Razzaq, M. Finite element simulation techniques for incompressible fluid–structure interaction with applications to bio–engineering and optimization, 2011; PhD Thesis, TU Dortmund.
8. Razzaq, M. Damanik, H. Hron, J. Ouazzi, A. Turek, S. FEM multigrid techniques for fluid-structure interaction with application to hemodynamics, Appl. Num. Math., 2011; 62, 1156-1170, http://dx.doi.org/10.1016/j.apnum.2010.12.010.
9. Turek, S. Hron, J. Mádlík, M. Razzaq, M. Wobker, H. Acker, J. F., Numerical simulation and benchmarking of a monolithic multigrid solver for fluid-structure interaction problems with application to hemodynamics, Fluid-Structure Interaction II: Modelling, Simulation, Optimisation, Springer, 2010; Ed: Bungartz, H.-J. and Mehl, M. and Schäfer, M., Lecture Notes in Computational Science and Engineering, 73:193-220.
10. Jaeggi, D. M. Parks, G. T. Kipouros, T. Clarkson, P. J. The development of a multi-objective tabu search algorithm for continuous optimization problems. European Journal of Operational Research, 2008; 185(3):1192–1212.
11. Glover, F. Laguna, M. Tabu search. 1999; Kluwer Academic Publishers, 3rd printing edition.
12. Tsotskas, C. Enhancing The Computational Engineering Design Process: An Environmental-friendly Aerocraft-trajectories perspective, Cranfield University, 2012; UK, PhD interim review.
13. Kipouros, T. Jaeggi, D. M. Dawes, W. N. Parks, G. T. Clarkson, P. J. Biobjective design optimization for axial compressors using tabu search. AIAA Journal 2008; 46(3):701–711.

14. Sackinger, P. A. Schunk, P. R. Rao, R. R. A Newton-Raphson pseudo-solid domain mapping technique for free and moving boundary problems: a finite element implementation, J. Comput. Phys 1996; 1(125): 83–103.
15. Girault, V. Raviart, P. A. finite element methods for Navier-Stokes equations, Springer, 1986; Berlin-Heidelberg.
16. Arnold, D. N. Boffi, D. Falk, R. S. Approximation by quadrilateral finite element. Math. Comput. 2002; 71(239): 909–922.
17. Turek, S. Hron, J. Razzaq, M. Wobker, H. Schäfer, M. Numerical Benchmarking of Fluid-Structure Interaction: A comparison of different discretization and solution approaches, Fluid-Structure Interaction II: Modelling, Simulation, Optimisation, Springer, 2010; Ed: Bungartz, H.-J. and Mehl, M. and Schäfer, M., Lecture Notes in Computational Science and Engineering, 73: 413-424.
18. Kelley, C. Iterative Methods for Optimization, 1999; 18, SIAM, Philadelphia.
19. Turek, S. Efficient solvers for incompressible flow problems, An Algorithmic and Computational Approach, Lecture Notes in Computational Science and Engineering, Springer-Verlag, 1999; 6.
20. Davis, T. A. Duff, I. S. A combined unifrontal/multifrontal method for unsymmetric sparse matrices. SACM Trans. Math. Software, 1999; 25: 1-19.
21. Lagarias, J. C. Reeds, J. A. Wright, M. H. Wright, P. E. Convergence properties of the Nelder-Mead simplex method in low dimensions. SIAM J. Optim. 1998; 9:112–147.
22. Nelder, J. A. Mead, R., A simplex method for function minimization. Computer Journal, 1965; 4(7): 308-313.
23. Becker, G. Siegmann, J. Michaelis, J. Schäfer, M., Comparison of a derivative-free and derivative-based shape optimization method. Proceeding of the 8th World Congress on Structural and Multidisciplinary Optimization, Springer, 2009; Ed: Rodrigues, J. M. and Guedes, H. C. and Fernandes, P. R. and Neves, M. M.
24. Mathews, J. H. Fink, K. K. Numerical Methods Using Matlab 2004; Prentice-Hall Inc.
25. Geilen, M. Basten, T. Theelen, B. Otten, R. An algebra of pareto points. Fundamenta Informaticae 2007; 78(1):35–74.
26. Inselberg, A. Parallel coordinates: visual multidimensional geometry and its applications. Springer, 2009; Dordrecht, New York.
27. Hooke, R. and Jeeves, T. A. Direct search solution of numerical and statistical problems. *J.ACM*, 8(2):212–229, 1961.
28. Russell, S. Artificial intelligence : a modern approach. 2010; Prentice Hall, Upper Saddle River, 3rd ed. edition.

DOI: 10.1002/