

Surrogate-based optimization for the worst-case prediction regarding a flexible structure impacted by wind gusts

Guillaume de Nayer, Michael Breuer

► To cite this version:

Guillaume de Nayer, Michael Breuer. Surrogate-based optimization for the worst-case prediction regarding a flexible structure impacted by wind gusts. Journal of Wind Engineering and Industrial Aerodynamics, 2023, 243, pp.105610. 10.1016/j.jweia.2023.105610. hal-04299137

HAL Id: hal-04299137 https://hal.science/hal-04299137

Submitted on 22 Nov 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés. Contents lists available at ScienceDirect



Journal of Wind Engineering & Industrial Aerodynamics

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



Surrogate-based optimization for the worst-case prediction regarding a flexible structure impacted by wind gusts

G. De Nayer, M. Breuer*

Professur für Strömungsmechanik, Helmut-Schmidt-Universität Hamburg, D-22043 Hamburg, Germany

ARTICLE INFO

Keywords: Worst-case scenario Meta-model Stochastic response surface Wind gust Fluid–structure interaction (FSI)

Large-eddy simulation (LES)

ABSTRACT

The paper is a follow-up of the recent study on the assessment of discrete wind gust parameters impacting a flexible lightweight structure as a first step towards the evaluation of the worst-case scenario caused by strong wind gusts (JWEIA 231, 105207, 2022). The present study goes beyond by suggesting an optimization framework which allows to determine the worst-case scenario automatically. For this purpose, a stochastic response surface algorithm with a surrogate model based on radial basis functions is chosen. The algorithm relies on costly evaluations of the objective function, which consist of CPU-time intensive fully coupled fluidstructure interaction (FSI) high-fidelity simulations including the pre- and post-processing of the results. Besides the parallelization of the coupled FSI solver, a parallel version of the optimization algorithm allows to carry out several costly evaluations simultaneously. The Metric Stochastic Response Surface algorithm determines the worst case fast. Then, it continues to explore the optimization space to ensure that the global extremum is reached. A sensitivity study on relevant parameters of the optimization algorithm is conducted. Typically, for the present FSI setup, an optimization run takes one week with 6 evaluations in parallel to compute 100 different configurations. The worst case is found after about one third of the evaluations. The increase of parallel evaluations drastically reduces the wall-clock time, but the worst case is found later after half of the evaluations. This later finding is due to the parallel nature of the algorithm. Finally, the various sources of uncertainties that arise throughout the entire procedure are assessed and discussed.

1. Introduction

Remarkable wind events such as strong wind gusts are nowadays of high interest for civil engineers due to their devastating consequences on constructions. Besides turbulent fluctuations of the approaching atmospheric boundary layer leading to fatigue of the building materials, rapid but strong wind events can result in a total breakdown. To hinder this, design standards provide guidelines for the dimensioning and testing. These standards are often developed based on measurements and rather simplistic assumptions. Therefore, to improve the design process, the existing standards have to be combined with modern high-fidelity numerical simulations.

Lightweight structures such as tents, roofs and large umbrellas are common in civil engineering design due to the fact that they are transportable and easy to shape. However, the counterpart of these advantages is that they are highly sensitive to wind loads and particularly to wind gusts because of their flexibility. Simulations of such deformable lightweight structures exposed to highly transient turbulent flows require a two-way coupling between the different media to be accurate. Moreover, to ensure the structural integrity of the design, the numerical predictions have to take the worst-case scenario associated with the investigated problem into account. The aim of the current project is to propose a software framework that is able to detect this worst-case scenario for flexible thin structures (membrane, shell) plunged into a turbulent boundary layer and exposed to discrete wind gusts.

This long-term project started with the development of a fluidstructure interaction (FSI) simulation framework relying on a highfidelity finite-volume flow solver using the large-eddy simulation (LES) technique coupled with a finite-element and isogeometric structure solver (Breuer et al., 2012). This partitioned FSI solver was carefully validated for laminar flows based on the FSI3 benchmark (Turek et al., 2010). Then, the FSI validation process switched to turbulent flows for thin-walled quasi-2D shell structures (De Nayer et al., 2014; De Nayer and Breuer, 2014) and a 3D air-inflated membrane (De Nayer et al., 2018a; Apostolatos et al., 2019). In order to be able to tackle the worstcase scenario for such a FSI setup originating from extreme wind events, the fluid solver was extended to include discrete wind gusts. A novel source-term formulation injecting wind gusts at an arbitrary location

https://doi.org/10.1016/j.jweia.2023.105610

Received 17 February 2023; Received in revised form 6 November 2023; Accepted 14 November 2023

0167-6105/© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

^{*} Corresponding author. *E-mail address:* breuer@hsu-hh.de (M. Breuer).

of the computational domain was derived in De Nayer and Breuer (2020). Combining this ingredient with the previously mentioned FSI framework, the deformations of an air-inflated flexible membrane in form of a hemisphere within a realistic turbulent boundary layer and exposed to strong discrete gusts could be investigated in De Nayer et al. (2022). To prepare the determination of the worst-case scenario for this civil engineering case, a sensitivity study was carried out to derive the different relevant input parameters and output objectives (De Nayer and Breuer, 2022). The present work is the direct continuation of this investigation to find the worst-case scenario in an efficient manner and deals with the extension of the computational framework by an optimization algorithm for computationally intensive problems.

The optimization topic is vast. In order to choose an appropriate optimization method for the present civil engineering problem, the selection process is restricted to algorithms especially developed for expensive objective functions. Based on the recent wind engineering and civil engineering literature surrogate-based optimization seems to be efficient for this purpose. Bernardini et al. (2015) and later Ding and Kareem (2018) apply Kriging surrogates calibrated by URANS and LES predictions to determine the aerodynamics of buildings possessing different cross-section shapes. Coupled to a genetic algorithm a multiobjective optimization of the shape of the civil engineering structure is presented. Similarly, Abdelaziz et al. (2021) use a genetic algorithm in order to minimize wind-induced vibration amplitudes of tall buildings. This genetic algorithm is coupled with two meta-models consisting of two artificial neural networks. Another application of a genetic algorithm with a feed-forward neural network as the metamodel is described by Muñoz-Paniagua and Garcia (2019): The nose shape of two crossing high-speed trains under cross-wind is optimized relying either on a single objective, the minimization of the side force coefficient, or on a multi-objective run, where the pressure pulse is also minimized. Another multi-objective optimization investigation of a high-speed train is conducted by Li et al. (2016) based on a genetic algorithm and a Kriging meta-model trained by RANS simulations. In civil engineering the behavior of bridges, particularly the prediction of their flutter velocity, are of high interest. An optimization of their shapes can be conducted as carried out by Kusano et al. (2020) for the Great Belt East Bridge. In this study a surrogate model was built relying on CFD simulations in order to estimate the force coefficients acting on the bridge.

Looking into the aeronautics literature for the determination of the worst-case scenario for the response of aircrafts to wind gusts, surrogate-based optimization also seems to be the best choice. For instance, Khodaparast et al. (2012) compare different meta-models such as Kriging and Radial Basis Function (RBF) as well as optimization techniques in order to identify the critical loads for a model with five objective functions. Boeing's proprietary optimizer denoted Design Explorer (Phoenix Integration, 2017), which relies on the combination of surrogate modeling with a gradient-based optimization, converges faster than the other methods. In their multidisciplinary design optimization (MDO) framework (Guzman Nieto et al., 2019) combine the Kriging meta-model with the efficient global optimization (EGO) algorithm (Jones et al., 1998) in order to obtain fast and accurate critical loads for a commercial aircraft model. Similarly, the MDO framework of the German Aerospace Center (Goertz et al., 2020) uses an enhanced Kriging surrogate model (Wilke, 2019) among others to build reduced-order aircraft models (Ripepi et al., 2018), which are then controlled by an EGO wing shape optimizer.

Furthermore, the literature focusing on optimization algorithms provides a variety of other surrogate-based optimization methods. Among them, the Scatter Search with Kriging for Matlab[®] (SSkm) by Egea et al. (2009) mixes the evolutionary approach with a Kriging response surface. The NOMAD framework (Audet et al., 2006, 2022) can join their implementation of the Mesh Adaptive Direct Search (MADS) algorithm with the Design and Analysis of Computer Experiments (DACE) software (Lophaven et al., 2002), which provides a Kriging model. Gutmann (2001a,b) develops a global optimization technique relying on RBFs, where the next iterate to be evaluated is determined by minimizing a bumpiness function. Regis and Shoemaker (2007b) introduce the Stochastic Response Surface (SRS) method for global optimization problems: At each iteration a response surface model (Kriging, RBF, ...) is updated and exactly one point is selected for function evaluation from a set of randomly generated points based on an user-determined merit function. For various problems it performs better than the RBFs technique by Gutmann (2001b). In order to increase the time efficiency, a parallel version is presented in Regis and Shoemaker (2007a).

Relying on the previous non-exhaustive literature review on the worst-case determination in aeronautics and civil engineering, SRS methods have apparently not yet been applied to this kind of problems. However, the results achieved on different benchmarks presented in the optimization literature are very encouraging. Therefore, a surrogate-based optimization method applying a Stochastic Response Surface method as introduced by Regis and Shoemaker (2007a,b) is selected for the present study. The objective of the optimization process relying on high-fidelity simulations is to find the worst-case scenario for an air-inflated membranous structure exposed to discrete wind gusts.

The manuscript is organized as follows: Section 2 introduces the worst-case scenario prediction framework including the optimization technique, the fluid and structure solvers and the fluid–structure coupling. The geometry and computational setup of the considered civil engineering case is detailed in Section 3. Based on this highly transient FSI problem the determination of the worst case, at which the maximal inner stresses are reached, is discussed and a brief sensitivity analysis on the relevant optimization parameters is conducted in Section 4. In addition, a detailed description of the optimization technique and a summary of the different sources of uncertainty that occur during the complete procedure can be found in the appendices.

2. Worst-case scenario prediction framework

Simulations of a multi-physics problem such as the FSI case tackled in this study are computationally intensive. In order to identify the worst-case scenario for such problems, which includes several input parameters, the number of expensive evaluations, i.e., pre-processing of the case, time-consuming numerical predictions based on coupled FSI simulations and post-processing of a huge amount of data, has to be kept as low as possible. To tackle this kind of issues, diverse response surface algorithms were developed as mentioned in the introduction. In the present framework the Matlab[®] implementation of a Stochastic Response Surface method (Regis and Shoemaker, 2007a,b) will be applied.

2.1. Stochastic response surface algorithm based on RBF surrogates

The determination of the worst-case scenario for the current case is carried out by the optimization function *surrogateopt* included in Matlab^{®1}. It applies a Metric Stochastic Response Surface (MSRS) algorithm with a surrogate model relying on radial basis functions for the response surface part. The description of stochastic response surface methods and their associated mathematical proofs are detailed in Regis and Shoemaker (2007b). An user documentation for *surrogateopt* is available². Obviously, the search for the worst-case scenario bases on the determination of a global maximum. However, to stay consistent with the terminology employed in the optimization literature, the MSRS algorithm described below and sketched in Fig. 1 looks for a global minimum, which can be resolved by the introduction of a minus sign

¹ https://de.mathworks.com/products/matlab.html

 $^{^{2}\} https://de.mathworks.com/help/gads/surrogate-optimization-algorithm. html$

in the objective function (see Section 3.5). In the following, only a brief specification of the entire algorithm is presented. For a detailed description the interested reader is referred to Appendix A.

The MSRS algorithm implemented in surrogateopt starts in step 1 with n_0 costly evaluations of the objective function f for a problem of dimension d. In the present case, f contains the FSI solver described in Section 2.2 followed by the post-processing. These initial points are generated by a quasi-random sequence and are called random samples. The so-called incumbent is determined as the point with the minimum objective function value among the costly evaluated points. Now, the core of the MSRS algorithm starts with step 2 which is a loop over the maximum number of costly evaluations. It includes step 2.1 in which the response surface model s_n is generated or updated relying on the already evaluated and costly vector points. Here, surrogateopt applies for the response surface model a radial basis function interpolator based on cubic splines with a linear tail. Step 2.2 explores the domain of evaluation within the bounds. For this purpose, a large amount of possible candidate points are randomly generated by samplers around the incumbent point. The procedure can be related to a local search. In order to explore the neighborhood as good as possible, several samplers (the random sampler, the mesh adaptive direct search OrthoMADS (Abramson et al., 2009) and the Generalized Pattern Search (GPS) (Torczon, 1997)) are applied in a cycle. Accordingly, the generation of the random candidate points is centered around the incumbent point. In Step 2.3 the next function evaluation points are determined. A so-called *merit* function relying on the previously generated response surface model is computed at every random candidate point generated in step 2.2, except those too close to the previously evaluated points defined by a user-defined dimensionless minimum distance d_{\min}^{sample} . The marit function f *merit* function f_{merit} implemented in the *surrogateopt* function is the weighted sum of the response surface value and the distance from the evaluated points as proposed by Regis and Shoemaker (2007b). The transition between a local search around the current incumbent point and a more global search depends on the value of the weight, where a large value encourages the local refining, while a smaller value encourages the global exploration. Since steps 2.2 and 2.3 are nested, the weight is adapted within the sampler cycle from 0.3 to 0.95. The next function evaluation point denoted adaptive point is the one with the highest f_{merit} value. If all candidate points are too close to the evaluated points, no new adaptive point is defined and steps 2.4 and 2.5 are skipped. In that case a local minimum is reached. If the maximum number of loops is not reached, a new search begins at step 1 based on other random points associated with a reset of the surrogate. Step 2.4 comprises the costly function evaluation of the objective function and in step 2.5 an update of the best function value is carried out if required. When the current number of expensive evaluations reaches its predefined maximum, the MSRS algorithm ends with step 3 and returns the best solution found.

As detailed in Appendix A, the optimization function has the option to use a kind of *parallelism* to strongly reduce the wall-clock time. Its associated uncertainty is discussed in Appendix B.6. Optimization runs based on the parallel version can be carried out with i_w costly evaluations treated simultaneously by the compute cluster as sketched in Fig. 1. Here i_w denotes the number of parallel workers. Per construction, the parallelism of the algorithm leads to a total amount of evaluated points for the generation of the surrogate slightly higher than the one defined by the user (which corresponds to n_0). n_0^{parallel} is at most n_0+i_w , since the optimization function switches to the next step, as soon as n_0 is reached, but waits for the workers, which are evaluating points.

For the present study the *surrogateopt* parameters 'MaxFunctionEvaluations' and 'MinSampleDistance' are set to $N_{\rm max} = 100$ and $d_{\rm min}^{\rm sample} = 10^{-3}$, respectively. The number of parallel workers is set to $i_w = 6$. The value $N_{\rm max} = 100$ combined with 6 parallel workers is a good compromise between an accurate solution for the global minimum and the required wall-clock time of the considered problem. One evaluation takes approximately between 8 and 18 h using 42 CPU cores (see Section 2.2). That leads to a total computation time of about one week on a parallel compute cluster for the worst-case determination, which is acceptable in the academic field. The remaining user-defined parameters related to the *surrogateopt* function are retained at their default values.

2.2. Coupled solver for FSI simulations

The objective function f mentioned in the previous section is composed of the preparation of the FSI case (the input parameters of the *n*th costly evaluation \mathbf{x}_n are passed to the input files of the solvers for the fluid and the structure), the numerical predictions of the FSI problem through an in-house FSI simulation framework and finally, the evaluation of the results, for example, the maximum of the inner stresses in the structure (see the sketch in Fig. 1).

The FSI simulation framework is based on a partitioned procedure (Breuer et al., 2012) relying on the computational structure dynamics solver Carat++ (Bletzinger et al., 2006) and the fluid solver FASTEST-3D (Durst and Schäfer, 1996). The coupling and the mapping between the two diverse surface discretizations at the FSI interface are carried out by the open-source software EMPIRE (Sicklinger et al., 2014). Strong and loose coupling schemes are available. Details about the presently applied mortar mapping in case of FEM discretizations can be found in Wang et al. (2016) and Apostolatos et al. (2019). All the data exchange relies on message passing interface (MPI) communications. This FSI simulation framework has been validated among others based on experimental FSI benchmarks (Kalmbach and Breuer, 2013; De Nayer et al., 2014, 2018a; De Nayer and Breuer, 2014; Wood et al., 2018).

Carat++ is a finite-element and an isogeometric structural solver developed with emphasis on the prediction of the mechanical behavior of thin-walled structures such as shells and membranes (Breitenberger et al., 2015; Philipp et al., 2016). The momentum equation written in a Lagrangian frame of reference is applied to describe the dynamic equilibrium of the structure. A St. Venant-Kirchhoff material law is assumed and links the Piola–Kirchhoff stress tensor with the Green–Lagrange strain tensor (Basar and Weichert, 2013). The time is discretized relying on the standard second-order non-linear Newmark scheme.

For FSI problems the fluid solver FASTEST-3D predicts complex turbulent flows using the large-eddy simulation (LES) technique by solving the filtered Navier-Stokes equations in the arbitrary Lagrangian-Eulerian (ALE) form. The equations are discretized based on the finitevolume technique on a curvilinear, block-structured body-fitted grid with a collocated variable arrangement. The surface and volume integrals are approximated by the midpoint rule. Most flow variables are linearly interpolated to the cell faces leading to a second-order accurate central scheme. To stabilize the simulation, the convective fluxes are approximated by the technique of flux blending (Ferziger and Perić, 2002). In the present study the flux blending includes 3% of a first-order accurate upwind scheme and 97% of a second-order accurate central scheme. In order to avoid unwanted oscillations, the momentum interpolation technique of Rhie and Chow (1983) for nonstaggered grids is applied to couple the pressure and the velocity fields. To solve the pressure-velocity coupling problem, a semi-implicit predictor-corrector scheme is applied: First, the momentum equations are time-marched by a low-storage multi-stage Runge-Kutta method to obtain an intermediate velocity. Then, the corrector step ensures that mass conservation is achieved in form of a divergence-free velocity field by solving a Poisson equation for the pressure correction.

Since LES is applied, the large scales of the turbulent flow are resolved and the small scales are modeled by a subgrid-scale (SGS) model. The classical (Smagorinsky, 1963) SGS model combined with the standard parameter $C_s = 0.1$ and with the Van-Driest damping function for the near-wall region is applied, since a preliminary study (De Nayer et al., 2018b) has proven that the model delivers



Fig. 1. Sketch of the complete worst-case scenario prediction framework (parallel version).

reasonable results for this test case. In order to mimic a realistic inflow, synthetic turbulent fluctuations generated by the recursive digital filter technique of Klein et al. (2003) can be injected at the inlet or within the computational domain based on the source-term method proposed by Schmidt and Breuer (2017), Breuer (2018) and De Nayer et al. (2018b). Note that this procedure satisfies the divergence-free condition, since it purely relies on a modification of the momentum equation not disturbing the mass conservation. For more information the reader is referred to the cited references. Note that details about the prescribed velocities, Reynolds stresses and length scale for the current application are specified in Section 3.2. Furthermore, the uncertainties related to the fluid simulation and turbulence modeling are discussed in Appendix B.1.

To study the worst-case scenario on the present FSI case, extreme wind events, i.e., discrete deterministic wind gusts, are injected into the computational domain. Various mathematical functions can be applied for this purpose. In this study solely the Extreme Coherent Gust (ECG) defined in the IEC-Standard (2002) is selected. The parameters used to describe the shapes are provided in Section 3.3 and the uncertainties associated with the gust modeling in Appendix B.2. Similar to the inflow turbulence the wind gusts are injected into the flow using a sourceterm formulation recently introduced by De Nayer and Breuer (2020). According to this method the gust shape is generated within the flow field by purely modifying the momentum equation. Thus, no mass is added to the flow. Therefore, after the solution of the Poisson equation for the pressure correction, the flow is divergence-free. Analogously to the injection of turbulent fluctuations, the source-term formulation for the gust injection also ensures the mass conservation of the flow. The distortions of the structure resulting from the fluid loads are taken into account at the FSI interface. The fluid mesh of the bodyfitted fluid solver is adapted using a fast procedure relying on linear and transfinite interpolations (TFI) in case of small deformations or by the hybrid grid adaption method developed for LES and large deformations by Sen et al. (2017).

The flow solver itself is parallelized based on classical domain decomposition and MPI communications. Since the flow simulation is the most expensive part, the majority of CPU cores (here 40) is used for this purpose. Contrarily, the structural solver Carat++ and the coupling tool EMPIRE solely need one core each.

3. FSI application case

In civil engineering numerous wall-mounted structures in hemispherical form exist such as silos, tents or stadiums. The flow around this kind of three-dimensional bodies is complex as demonstrated in Wood et al. (2016). Besides the flow complexity itself a flexible structure made of a thin membrane leads to a challenging FSI case. The application case described in the following is originating from the experimental investigation carried out by Wood et al. (2018). Here, the objective was to set up a validation case in the wind tunnel for a flexible membranous structure exposed to a turbulent boundary layer. Thus, the real size of such a construction had to be scaled down and an appropriate material had to be found. After finding reasonable parameters of the setup, the measurement campaign was successful and delivered a huge amount of data for the comparison with coupled FSI simulations (De Nayer et al., 2018a; Apostolatos et al., 2019). Although



Fig. 2. Sketch of the considered FSI civil engineering case. The input parameters varied within this optimization study are marked by a box. The others are considered to be fixed.

the experimental study did not consider the effect of wind gusts up to now, the same generic setup is used in the present investigation on the worst-case scenario. Ongoing experimental investigations are trying to incorporate wind gusts into the wind tunnel setup (Wood et al., 2022).

3.1. Geometry and material of the structure

A flexible membrane of average thickness $t_{\text{membrane}} = 1.65 \times 10^{-4}$ m is wall-mounted and exposed to the gravitational acceleration g = -9.81 m/s^2 e₃ (see Fig. 2). The hemispherical form (diameter D) is obtained by applying a pressure difference $\Delta p_{\rm FSI} = p - p_{\infty} = 43$ Pa between the inner gauge pressure of the structure and the ambient pressure. This value is a satisfying compromise between stabilization and contour accuracy (Wood et al., 2018). The membrane material is silicone (Wacker Elastosil 625). For a St. Venant-Kirchhoff material the density, the Young's modulus and the Poisson's ratio are determined to $\rho_{\rm silicone} = 1050 \, {\rm kg/m^3}$, $E_{\rm silicone} = 7 \times 10^5$ Pa and $\nu_{\rm silicone} \approx$ 0.45, respectively. Since silicone is a rubber-type material, the internal structural damping cannot be neglected. Pure structural test cases presented in Wood et al. (2018) allow to set a correct material behavior for the simulations. In the present study the material damping is modeled by the well-established Rayleigh damping. The constant mass- and stiffness-proportional parameters are set to $\alpha_r = 17.47$ and $\beta_r = 1.89 \times 10^{-4}$ as determined in De Naver et al. (2018a). A discussion regarding the inherent uncertainties associated with the structural modeling and with the geometry representation can be found in Appendices B.3 and B.4, respectively.

3.2. Approaching boundary layer

The smooth wall, on which the body is mounted, generates a thick turbulent boundary layer *without wind gust*. Based on the measurements of Wood et al. (2016) the distribution of the time-averaged streamwise velocity at a distance of 1.5 diameters upstream of the structure closely follows the 1/7 power law and the thickness of the boundary layer δ corresponds to the height of the hemisphere ($\delta = H = D/2$). Based on the free-stream velocity $u_{\infty} = 9.95$ m/s in streamwise direction at standard atmospheric conditions ($\rho_{air} = 1.225$ kg/m³, $\mu_{air} = 18.27 \times 10^{-6}$ kg/(m s)) the Reynolds number is 100,000. In order to obtain accurate numerical predictions, this kind of turbulent inflow has to be realistically modeled. Based on the time-averaged velocities and Reynolds stresses measured in the experiment (Wood et al., 2018) and complemented by DNS data of Schlatter et al. (2009), synthetic inflow (STIG) data are generated by the digital filter concept of Klein et al. (2003) and injected at a distance of 1.5 diameters upstream

of the body. The integral length and time scales of the turbulence applied are $L_z^{\text{STIG}}/D = L_y^{\text{STIG}}/D = L^{\text{STIG}}/D = 2.04 \times 10^{-2}$ and $T^{\text{STIG}}u_{\infty}/D = 2.85 \times 10^{-2}$, respectively (for more details, see De Nayer et al. (2018a)). Based on these settings a peak turbulence level of $\text{Tu}_{\text{STIG}} = \sqrt{\frac{1}{3}(\overline{u'u'} + \overline{v'v'} + \overline{w'w'})}/u_{\infty} = 8\%$ is reached close to the bottom wall for the situation without any discrete gust.

Please note that all parameters listed in this section are constant within the current investigation.

3.3. Wind gust modeling

The perturbations included in the synthetic turbulent boundary layer are of limited amplitudes since the objective was to describe the approaching boundary layer. Thus, the effect of strong wind gusts triggering large structural deformations and leading to a worst-case scenario are not taken into account in the description of the approaching turbulent boundary layer. To consider such impacts, discrete deterministic wind gusts are added to the flow as described in Section 2.2. The shape of the gust is defined in the local basis $B_1 = (O, \mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3)$ (see De Nayer et al., 2022, for details about the theory and the basis) based on the velocity by the mathematical functions f_1, f_2, f_3 in the three spatial directions and f_t in time:

$$\mathbf{u}_{g}\Big|_{B_{t}}(t,\xi,\eta,\zeta) = A_{g} f_{t}(t) f_{1}(\xi) f_{2}(\eta) f_{3}(\zeta) \mathbf{g}_{1} .$$
(1)

The strength of the wind gust can be controlled by the input parameter denoted *gust amplitude* A_g . In the present case the gust is injected in the streamwise direction, i.e., $\mathbf{g}_1 = \mathbf{e}_1$, to maximize the effects of the gust on the structure. An adapted version of the ECG gust type introducing the central value ϕ_g is considered (De Nayer et al., 2019) to define the functions f_1, f_2, f_3 and f_1 :

$$f_{i}(\phi) = \begin{cases} \frac{1}{2} \left(1 + \cos\left(\frac{2\pi \left(\phi - \phi_{g}\right)}{L_{g}^{\phi}}\right) \right) & \text{for} \quad \left(\phi - \phi_{g}\right) \in \left[-\frac{L_{g}^{\phi}}{2}, \frac{L_{g}^{\phi}}{2}\right] \\ 0 & \text{else} . \end{cases}$$

$$(2)$$

The variable ϕ is equal to the coordinate $\phi = \{x, y, z\}$ or the time *t* and the subscript *i* is equal to 1, 2, 3 or *t*, respectively. L_g^{ϕ} represents the length and time scales of the wind gust. For the considered FSI application the temporal shape function $f_t(t)$ follows the same analytical function as the spatial shape $f_1(\xi)$ oriented in gust direction, i.e., the streamwise direction. The length scales of the gust in wall-normal and spanwise direction are fixed to $L_g^{\eta} = L_g^{\zeta} = 1 D$, while that in the gust

direction L_g^{ξ} (and consequently L_g^t , since $L_g^t = L_g^{\xi}/u_g^{\text{conv}}$ with $u_g^{\text{conv}} = u_{\infty}$ here) is an input parameter of the optimization problem.

Since the discrete wind gust can be injected everywhere inside the computational domain, the position vector of the gust (x_g^0, y_g^0, z_g^0) is of relevance. In the present FSI study x_g^0/D is set to -1.5 so that the longest gust (largest L_g^{ξ}) can be fully injected before reaching the flexible structure. y_g^0 is set to zero, which means that the center of the gust is in the symmetry plane of the structure. That maximizes the impact of the gust on the body. As demonstrated in De Nayer and Breuer (2022) the altitude z_g^0 at which the gust is injected plays a dominant role for the structural deformations. Therefore, it is also an input parameter of the optimization.

To resume the gust amplitude A_g , its length scale in streamwise direction L_g^{ξ} and its injection altitude z_g^0 are input parameters for the optimization run. In the precursor sensitivity study (De Nayer and Breuer, 2022) the effect of these wind gust parameters was already evaluated. Meaningful ranges for the investigation of the worst-case scenario were derived for A_g and L_g^{ξ} : $0.5 \le A_g/u_{\infty} \le 1.0$, $0.5 \le L_g^{\xi}/D \le 1.0$. The gust injection altitude z_g^0 was shown to have a major impact on the results. Therefore, in order to be sure to reach the worst case in the present work, its variation range is increased in both directions to $1/8 \le z_g^0/D \le 6/8$. The remaining parameters are fixed in the current investigation.

3.4. Computational setup

The thin-walled structure is composed of 1926 finite membrane Constant Strain Triangle (CST) elements and 999 nodes based on the grid-independence study carried out in De Naver et al. (2018a). All nodes are free except those connected to the smooth plate at the bottom, which cannot move in any translational direction. The membrane has an outer side in contact with the air flow solved by the fluid solver and an inner side, where the air is over-pressurized. The domain inside the hemisphere is not solved by the fluid solver. It is assumed that the pressure difference Δp_{FSI} stays constant over time. In order to take that into account, a follower pressure load acting on each surface element is added directly to the structural solver. Concerning the outer side, the fluid loads composed of the pressure and the shear stresses are predicted by the fluid solver FASTEST-3D and exchanged by the coupling and mapping tool EMPIRE (Sicklinger et al., 2014). Additionally, a dead load acting on the volume of each membrane element models the gravitational acceleration g. In order to define the initial state of the membranous structure, a homogeneous and isotropic pre-stress tensor field is prescribed with the value of $n_{\text{membrane}} = 7794.5$ Pa (see De Nayer et al., 2018a, for the determination of this value).

The computational domain for the fluid solver is a perfect hemispherical expansion in radial direction (radius 10D) starting at the center of the structure. A block-structured grid composed of $4.3 \times$ 10⁶ control volumes (CV) was proved to be sufficient for acceptable predictions of the flow close to the walls and close to the body (see De Nayer et al. (2018b) for details). In order to fulfill the recommendations by Piomelli and Chasnov (1996) for wall-resolved LES the first cell center is located at a distance of $\Delta z/D \approx 5 \times 10^{-5}$ from the wall leading to maximal z^+ values of 0.25 based on the time-averaged flow data. Furthermore, the aspect ratio of the CVs on the body are between 1 and 10 and the geometric stretching ratios in radial direction are kept below 1.1. During the FSI simulation this fluid grid is adapted using the fast TFI method to take the deformation of the body into account and to keep the computational effort acceptable. Concerning the boundary conditions, the bottom plate and the flexible membrane are considered as no-slip walls. The outer surface of the hemispherical expansion is divided into inlet and outlet patches. In the lower part of the outlet patch, where large turbulent vortices are present and have to pass the outlet boundary without perceptible disturbances, a convective boundary condition is applied.

Regarding the FSI problem, a loose coupling is applied, since the density ratio $\rho_{\rm fluid}/\rho_{\rm structure}$ is far below unity, i.e., the added-mass effect is very limited. The uncertainties stemming from the FSI coupling are addressed in Appendix B.5. Both solvers use the same constant time step $\Delta t^* = \Delta t u_{\infty} / D = 3.317 \times 10^{-5}$ leading to a CFL number of about 3×10^{-2} (computed with the local velocity and the length of the cell). In the first FSI investigations conducted with the flexible hemisphere (De Naver et al., 2018a; Apostolatos et al., 2019) long predictions in term of simulation time had to be carried out in order to determine the statistics of the turbulent flow field and the structure deformations. In De Nayer et al. (2022) and De Nayer and Breuer (2022) the required simulation time was also long in order to include the whole passing of the gust through the wake. In the present configuration a simulation time of 100,000 time steps ($t^* \approx 3.3$) after the injection of the gust ensures to capture the approaching gust, the impact of the gust on the structure and the time interval when the gust is leaving the structure. Thus, the worst-case scenario resulting from the impact of the wind gust on the body can be evaluated with this reduced number of time steps.

3.5. Objective function

For engineers the determination of the highest inner stresses occurring in the material is of high interest in order to correctly dimension their design and avoid material failures. Therefore, the global maximum of the von Mises stresses in the structure is used as the objective function for the present optimization run. At each time step the von Mises stresses σ_{Mises} are evaluated for each finite element of the membrane and the maximum denoted $\max(\sigma_{\text{Mises}})$ is tracked. At the end of the FSI simulation the maximum of the time history of $\max(\sigma_{\text{Mises}})$ is taken as the objective function *f*. Since *surrogateopt* searches for a global minimum, -max(σ_{Mises}) is returned.

4. Results and discussion

4.1. Summary of the gust-membrane interaction

In De Nayer et al. (2022) the effect of wind gusts on the flow field, the resulting forces on the structure and the corresponding deformations of the flexible structure were analyzed in detail. For a better understanding of the following consideration of the worst-case scenario, some important results of these investigations are briefly summarized here.

Due to the injected gust the streamwise velocity in the region upstream of the flexible structure raises approximately according to $u_1^{\max}/u_{\infty} = 1 + A_g/u_{\infty}$. When the gust reaches the hemisphere, the resulting forces in streamwise and vertical direction increase significantly and the streamwise component reaches its first local maximum (see Fig. 3(a) for a typical time history). The structure starts to deform and the front of the membrane grows hollow due to the impact of the gust visible by a high pressure in this region. Then the gust convects downstream. When the gust is located above the hemisphere, the strong suction pulling the body upwards leads to the maximum vertical force. Afterwards the wind gust detaches from the structure. The low pressure area in the back of the body is responsible for the second peak value of the streamwise force. During this phase the forming indentation at the front still gets larger and deeper. A delay between the maxima of the total forces and the maxima of the local deflections of the front is detected. The latter is observed when the gust has already passed the hemisphere completely (see Fig. 3(c)). Afterwards the front indentation starts to decrease again and the deflections of the inflated hemisphere slowly return to their original level. To analyze failure of ductile materials the von Mises stress was predicted as a classical criterion to be compared with the yield stress of the material. For $z_a^0/D = 2/8$ the largest von Mises stresses during the gust impact were observed very close to the bottom wall at the front gap resulting from



(a) Time history of the streamwise (solid blue line) and vertical (dashed blue line) force coefficients acting on the membrane during a typical gust impact as well as the resulting maximum of the von Mises stresses (solid red line).



(b) Distribution of the von Mises stresses for the time instant at which $\max(\sigma_{\text{Mises}})$ is maximum $(t^* \approx 1.81)$.

(c) Membrane during the maximal deflection by the wind gust $(t^* \approx 2.56)$.

Fig. 3. Typical critical case $(z_g^0/D = 2/8, A_g/u_{\infty} = 1.0, L_g^{\xi}/D = 1.0).$

the gust impact. It has to be noticed that another area of high von Mises stresses is located at the periphery of this gap and particularly at the apex of the hemispherical membrane. That is visible in Fig. 3(b) and again appears earlier ($t^* \approx 1.81$) than the maximal deflection ($t^* \approx 2.56$).

4.2. Determination of the worst case

Applying the framework introduced in Section 2 and sketched in Fig. 1 with 6 parallel workers, the determination of the worst case starts with the initialization phase (**STEP 1**). n_0^{parallel} (here 25) costly evaluations of the objective function are carried out. The values of the input parameters z_g^0/D , A_g/u_∞ and L_g^{ξ}/D of these FSI simulations are chosen randomly by the optimization algorithm to explore the whole space within the user-given bounds provided in Section 3.3. Fig. 4(a) depicts the spreading of those *random points* in the 2D planes z_g^0/D vs. A_g/u_∞ , z_g^0/D vs. L_g^{ξ}/D and A_g/u_∞ vs. L_g^{ξ}/D , respectively. The coupled simulation of the n_0^{parallel} evaluations delivering the smallest objective value, i.e., the largest von Mises stresses, is marked as the *incumbent*.

After this initialization step the second phase of the optimization algorithm, i.e., the determination of the minimum, begins by constructing the RBF surrogate s_{n_0} based on the n_0^{parallel} previously evaluated

random samples (STEP 2.1) (see Fig. 4(b)). Then, the Metric Stochastic Response Surface (MSRS) algorithm generates around the incumbent numerous candidate points (STEP 2.2) to be rated based on their merit function as defined in Eq. (A.3) (STEP 2.3). The selection of the candidate samples, denoted adaptive points, is driven by two factors: On one side, the MSRS algorithm wants to explore the whole space within the bounds. On the other side, it looks for a minimum. This dual task of the MSRS algorithm is clearly visible in Figs. 4(c). Some of the selected and evaluated adaptive points are far from the incumbent and from the minimum, but fill a gap between other already evaluated samples in order to improve the meta-model. Since the parallel version of the MSRS method is applied with 6 workers, 6 adaptive points are asynchronously evaluated in parallel (STEP 2.4). Afterwards, a new incumbent is determined among all evaluated points (STEP 2.5), the RBF surrogate is updated with these 6 new samples and the next candidate points are generated for assessment. This loop proceeds till a minimum is found. Fig. 4(c) gathers all selected and costly evaluated candidate points until the surrogate reset. Since 6 parallel workers are used, each set of 6 newly evaluated points are marked by the same symbol and the same gray scale value on the plot, so that the reader can follow the progression of the algorithm thorough the space. In the present optimization run the minimum is reached at $(z_{\alpha}^{0}/D,$ $A_{\sigma}/u_{\infty}, L_{\sigma}^{\xi}/D) = (0.219, 1, 1)$ with an objective value of $-\max(\sigma_{\text{Mises}})$ = -66.17 kPa. The worst case is the 38th point evaluated (including the 25 initial random points). Thus, it can be remarked that the MSRS algorithm determines the worst case fast, but continues to explore the space around the incumbent to be sure that the minimum is found. This can be observed in Fig. 4(c): The first sequence of adaptive points are mostly concentrated close to the incumbent determined by the initial random points and within the area of high maximal von Mises stress given by the surrogate (marked in red in Fig. 4(b)). This leads to the localization of the point $(z_g^0/D, A_g/u_\infty, L_g^{\xi}/D) = (0.219,1,1)$. Then, the algorithm fails to find a lower minimum in that area. Therefore, it tries to explore the rest of the space as visible in Fig. 4(c). If the MSRS method does not succeed to improve this local minimum and all candidate points are too close to each other, it stops and decides to restart the procedure at STEP 1 to ensure that the global minimum is found. In the present run, this case occurs after 73 evaluations. New initial random points are generated, but they have to be different from the already evaluated ones. The surrogate is reset and the whole procedure proceeds until the maximum number of costly evaluations N_{max} (here $N_{\text{max}} = 100$) is reached. Note that only one surrogate reset is done in the presented run. A run with $N_{\text{max}} = 200$ shows that no better minimum can be found.

The amplitude $A_g/u_{\infty} = 1$ and the length scale of the gust $L_g^{\xi}/D = 1$ leading to the maximal von Mises stress is not a surprise. The stronger and wider the gust is, the higher is its energy leading to large deformations of the structure. Therefore, the MSRS algorithm found A_g/u_{∞} and L_{q}^{ξ}/D equal to their upper bounds. The circumstance that the worst case is determined for a low value of the third parameter z_{σ}^{0}/D was not foreseeable. As visible in Fig. 3(b), the high values of the von Mises stresses concentrate at the periphery of the gap (particularly close to the apex) resulting from the impact of the gust in front of the membrane and also close to the bottom wall due to this boundary. For values of z_{α}^{0}/D larger than 0.252 the high von Mises stresses locate more at the periphery at the apex, for values of z_g^0/D less than 0.252 they are detected close to the bottom wall. This can be explained by the role of the fixed nodes of the membranous structure at the bottom wall. They lead to high strains and thus high inner stresses in the membrane elements close to the bottom plate. It is interesting to remark that the worst case for the maximum von Mises stress $(z_g^0/D, A_g/u_\infty, L_g^{\xi}/D)$ = (0.219, 1.0, 1.0) is not the same as the worst case for the maximum vertical force as another objective function reached at $(z_a^0/D, A_a/u_{\infty})$, L_g^{ξ}/D = (0.368,1.0,1.0), although they occur at nearly the same time instant (see Fig. 3(a)).

The optimization parameter n_0 was set to 20 leading to $n_0^{\text{parallel}} = 25$ in the present parallel run. Since the first surrogate relies on the n_0^{parallel} first costly evaluations of the objective function, this value has a direct influence on the accuracy of the first meta-model s_{n_0} . Fig. 5 depicts the relative error ε between s_{n_0} and $s_{N_{\text{max}}}$ relying on all samples and considering $s_{N_{\text{max}}}$ as reference. The maximum of ε , about 20% in the present run, is located in the area where the worst case is supposed to be. This makes sense: since more points are evaluated in this region and thus the updated meta-model adapts there. Far away only a few points are chosen, therefore the surrogate is barely updated and its error is close to zero. In the current run the error obtained with $n_0^{\text{parallel}} = 25$ is quite low. Consequently, $n_0^{\text{parallel}} = 25$ is considered to be enough and will not be modified.

With the presented framework multiple MSRS optimization runs were carried out on the HPC cluster of the Helmut-Schmidt-Universität, HSUper, built with Intel Icelake sockets and Intel[®] Xeon[®] Platinum 8360Y processors and connected by a non-blocking NVIDIA InfiniBand HDR100 fabric (Neumann and Preuß, 2022). The parallel run with 6 workers and $N_{\rm max} = 100$ costly evaluations described here took about 8 days to finish. The wall-clock time for one costly evaluation is between 8 and 18 h. This large deviation in the wall-clock time is due to the

differences in the membrane deformations resulting from the impact of gusts with different strengths and lengths.

Due to modeling, discretization, and computational errors, the results and thus the worst-case obtained from the computational framework carry a certain degree of uncertainty. To evaluate the reliability of the results, Appendix B provides a catalog of diverse sources contributing to this uncertainty.

4.3. Sensitivity analysis of the optimization settings

Several optimization settings have a major impact on the complete run. The total amount of evaluations N_{max} directly influences the wallclock time consumption of the complete optimization run. As already mentioned, $N_{\text{max}} = 100$ is enough to deliver the worst-case scenario and leads to an acceptable run time of one week.

The minimal sample distance d_{\min}^{sample} is an important dimensionless parameter. Applying the same seed for the quasi-random generator, the standard value of $n_0 = 20$ and the given fixed bounds for the input parameters, a variation of d_{\min}^{sample} has no impact on the generation of the initial random points. Therefore, the incumbent among the random points remains identical and the latter are not depicted in the already crowded Fig. 6. However, the choice of d_{\min}^{sample} impacts the selection of sample points. In the run presented before, d_{\min}^{sample} is set to 10^{-3} . Increasing d_{\min}^{sample} to a value larger than 5×10^{-3} has for consequence that the algorithm is more limited for the choice of candidate points within the bounds. Thus, the evaluated samples are more spread through the space as visible in Fig. 6. With this configuration the worst case at $(z_g^0/D, A_g/u_\infty, L_g^\xi/D) = (0.219, 1.0, 0.95)$ is found faster than with $d_{\min}^{sample} = 10^{-3}$ (28th vs. 38th evaluation including initial points). points). However, the magnitude of its objective value $-\max(\sigma_{\text{Mises}}) =$ -63.82 kPa is 3.5% less than that of the objective value found with $d_{\min}^{\text{sample}} = 10^{-3}$. Thus, the value of the minimal sample distance d_{\min}^{sam} is decisive for the optimization run: A too large value prevents the framework to determine an accurate worst case. Decreasing d_{\min}^{sample} to 10^{-6} has for consequence that the algorithm considers candidate points, which are closer to each other. The accuracy of the worstcase parameters will be higher: The worst case is determined for $(z_g^0/D, A_g/u_{\infty}, L_g^{\xi}/D) = (0.217, 1.0, 1.0)$ with an objective value of - $\max(\sigma_{\text{Mises}}) = -66.18$ kPa, which has an amplitude 0.016% larger than with $d_{\min}^{\text{sample}} = 10^{-3}$. However, the MSRS algorithm requires more with d_{\min}^{samp} evaluations to reach this result. With $d_{\min}^{\text{sample}} = 10^{-3}$ the worst case is attained at the 20th sample. is attained at the 38th evaluation. With a value of $d_{\min}^{\text{sample}} = 10^{-6}$ it takes about 40% more evaluations (about 53). Moreover, the d_{\min}^{sample} parameter directly influences the surrogate reset. The higher d_{\min}^{sample} is, the faster the surrogate reset occurs.

The parameter i_w , i.e., the number of parallel workers, plays a major role in the time consumption. A complete parallel run with 12 workers takes about 4 and half days, i.e., half of the time needed by a run with 6 workers. This result is not surprising, since about the same amount of costly evaluations N_{max} are carried out in both cases. Please remark that a parallel run with more workers carries out slightly more evaluations due to the asynchronism of the algorithm. When the parallel MSRS method switches between phases, the samples being evaluated remain in service, but any other points in the queue are removed from the queue. Therefore, when the $N_{\rm max}$ th evaluation terminates, the optimization algorithm stops, but waits until the other running evaluations finish. The parameter i_w has no influence on the incumbent among the initial random points. Again, due to the asynchronism of the parallel algorithm, slightly more initial random points (31 vs. 25) are used with $i_w = 12$. With the same minimal sample distance $d_{\min}^{\text{sample}} = 10^{-3}$ the parameter i_w has no impact on the final worst case: $i_w = 12$ determines the point $(z_g^0/D, A_g/u_{\infty}, L_g^{\xi}/D) = (0.219, 1.0, 1.0)$ as the worst-case scenario exactly as $i_w = 6$. However, due to the parallelism of the MSRS algorithm it is discovered later (52th vs. 38th evaluation including initial points). Fig. 7 compares the distributions of the sample



(a) **STEP 1:** n_0^{parallel} costly evaluations at *random points* generated by a quasi-random sequence to explore the space. The point with a blue square is the incumbent \mathbf{x}_n^* among these initial *random points*.



(b) **STEP 2.1:** A surrogate s_{n_0} is generated relying on the n_0^{parallel} random points.



(c) **STEPS 2.2-2.5:** Based on the surrogate, 6 *adaptive samples* (square in gray) are chosen around the incumbent and costly evaluated. Then, the surrogate is updated and 6 new *adaptive samples* (delta symbols) are chosen for evaluation. The sequence is repeated (gradient, circle, diamond, left-, right-triangle and + symbols) until a local minimum (diamond highlighted by a square in red) is found.

Fig. 4. Typical stochastic response surface optimization run (6 parallel workers with $d_{\min}^{\text{sample}} = 10^{-3}$).



Fig. 5. Relative error between the surrogate s_{n_0} based on the n_0^{parallel} initial random samples and the surrogate $s_{N_{max}}$ relying on all samples (6 parallel workers with $d_{\min}^{\text{sample}} = 10^{-3}$).

points in the space $(z_g^0/D, A_g/u_\infty, L_g^\xi/D)$ for $i_w = 6$ and 12. These distributions are not exactly identical but similar, showing that the parallelism implementation of the MSRS method works well.

5. Conclusions

To determine the worst-case scenario of lightweight membranous structures attacked by strong discrete wind gusts, a new methodology based on high-fidelity solvers has been developed. The core is a partitioned FSI solver that predicts the resulting loads and stresses in the structure when exposed to discrete gusts. The most critical wind gust defined by its properties (gust strength, length and vertical position) are found by an optimization algorithm. Since the evaluations of the objective function are expensive, particular importance was put on this aspect when selecting the metric stochastic response surface algorithm with a surrogate model based on radial basis functions as an appropriate optimization framework. Thus, costly evaluations of the objective function can be minimized. The whole process can be fully automated. Furthermore, to shorten the wall-clock time of the procedure, a certain number of parallel workers can be assigned which simultaneously evaluate the objective function.

The following main conclusions can be drawn from this study:

- The proposed algorithmic framework works reliably and determines the global minimum of the considered problem, i.e., here the worst-case scenario given by the highest von Mises stress in the structure. Throughout the entire project, a systematic investigation was conducted to identify and assess the various sources of uncertainties within the procedure, with the aim of minimizing their impact.
- · For the present case of a wall-mounted flexible membranous structure impacted by discrete wind gusts, a number of $n_0 = 20$ initial random samples (25 evaluations are done with 6 parallel workers, 31 with 12 parallel workers) is sufficient to generate an accurate RBF response surface model for the first pass. Furthermore, the user has to pay attention to the minimal sample distance d_{\min}^{sample} . With a low value, i.e., $d_{\min}^{\text{sample}} = 10^{-6}$, the worst case is accurately tracked, but requires more costly evaluations after the surrogate reset. With a value larger than d_{\min}^{sample} 5×10^{-3} a "worst" case is found faster. However, this sample is not exactly the worst case, since the magnitude of the von Mises stress maximum is 3.5% lower than the one predicted with $d_{\min}^{\text{sample}} = 10^{-6}$. For the present case a value of $d_{\min}^{\text{sample}} = 10^{-3}$ is found to be the best compromise between the accuracy of the worst case and the assessment of the whole space. A value of $N_{\rm max} = 100$ maximal costly evaluations is proven to be sufficient to determine the global minimum of the problem.
- The methodology is parallelized on two levels. First, the timeconsuming coupled FSI simulations relying on a partitioned solver

are parallelized based on classical domain decomposition with explicit message passing. Second, the application of a certain number of parallel workers for the determination of the worst case speedups the optimization process significantly. Thus, within a few days the worst case of the highly challenging application case can be determined.

- From the application of the devised methodology to an airinflated hemispherical structure the following conclusions can be drawn:
 - Although the discrete wind gust model represents only a simplifying assumption, it allows to compare the critical gust strength with the Beaufort wind force scale. In the present case the maximum gust strength leading to critical loads has a Beaufort number of 8 and is denoted a *gale*.
 - During the gust impact the forces on the structure rise by a factor of less than two, however, the maximum von Mises stress increases by a factor of more than 3 with significant deflections of the membrane.
 - It is worth noting that the worst-case scenario found for the maximum von Mises stress does not align with the worst-case scenario relying on the maximum vertical force. Even though these critical points occur at almost the same time instant, the former is reached at a lower gust injection altitude ($z_g^0 = 0.219 D$) compared to the latter ($z_g^0 = 0.368 D$). From a civil engineering perspective, prioritization of the worst-case scenario associated with the maximum stress is more pertinent for ensuring structural integrity.
 - Comparing the height of the hemispherical dome *D* with the altitude z_g^0 at which the gust is injected, the worst-case scenario resulting in the highest von Mises stresses is found for the situation when the gust center is quite close to the bottom plate ($z_g^0 = 0.219 D$). The reason is the fixation of the membranous structure at the bottom wall which leads to high strains and thus high inner stresses in that region. Conversely, one cannot conclude that wind gusts close to the ground are always particularly critical, as this depends heavily on the structure under consideration.
- The methodology developed is neither restricted to the present application case (membranous structure) nor to the simulation technique currently applied (partitioned solver/large-eddy simulation). For example, the usage of the Reynolds-averaged Navier– Stokes approach could further drastically reduce the CPU-time requirements, but this is accompanied by losses of the accuracy and reliability of the predictions.

The presented framework for the worst-case scenario prediction will be applied to further FSI cases. A large-span membranous four-points



Fig. 6. Effect of *a*^{sample}_{min} on the sample creation for 6 parallel workers. Incumbent among the initial *random points* highlighted in blue, *adaptive samples* depicted by gray symbols, worst case depicted by a diamond highlighted by a square in red.

hypar flexible canopy is currently under investigation and leads to new challenges.

CRediT authorship contribution statement

G. De Nayer: Conceptualization, Methodology, Software, Validation, Investigation, Visualization, Writing – original draft, Writing – review & editing. **M. Breuer:** Conceptualization, Methodology, Resources, Writing – original draft, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The project is financially supported by the Deutsche Forschungsgemeinschaft, Germany under the contract numbers BR 1847/17-1. The authors want to thank the Chair of Structural Analysis of the Technical University of Munich for providing the computational structure dynamics solver Carat++ (Bletzinger et al., 2006) and the open-source coupling software EMPIRE (Sicklinger et al., 2014). Additionally, the authors are grateful to Andreas Apostolatos (MathWorks[®]) for the fast and deep support on the different Matlab[®] functions used in this study. The major part of the optimization runs and their costly FSI simulations were conducted on HSUper (Neumann and Preuß, 2022), the central computer cluster of the Helmut-Schmidt-Universität Hamburg provided by the project hpc.bw, funded by dtec.bw – Digitalization and Technology Research Center of the Bundeswehr³. dtec.bw is funded by the European Union – NextGenerationEU.

Appendix A. Details on the stochastic response surface algorithm based on RBF surrogates

As briefly explained in Section 2.1 the determination of the worstcase scenario is carried out by the optimization function *surrogateopt* included in Matlab^{®4}. It applies a Metric Stochastic Response Surface (MSRS) algorithm with a surrogate model relying on radial basis functions for the response surface part. The different steps of the procedure were briefly described in Section 2.1 and schematically shown in Fig. 1. Contrary to the above mentioned section, here the different steps are described in detail.

The MSRS algorithm implemented in *surrogateopt* starts in **STEP 1** (Initialization) with n_0 costly evaluations of the objective function f (here the FSI solver described in Section 2.2 followed by the post-processing) for a problem of dimension d. These initial points represented by the vector of input parameters $\mathbf{x}_i \in \mathbb{R}^d$ for $i = 1, ..., n_0$ are generated by a quasi-random sequence. Therefore they are called *random samples*. For this purpose, an uniform random deviate between 0 and 1 is scaled and shifted to remain within the respective bounds. When n_0 reaches the limit of max (20, 2 d), the initialization step is over. In case of a parallel optimization run n_0^{parallel} is slightly larger than n_0 , since it depends also on the total number of parallel workers $(n_0 \le n_0^{\text{parallel}} \le n_0 + i_w)$. This limit n_0 is arbitrary, but chosen against the background that the evaluations are expensive. n_0 is stored in n and \mathbf{x}_n^* denoted *incumbent* is determined as the point with the best function value $f_n^* = f(\mathbf{x}_n^*)$ among the costly evaluated points. Now, the core of the MSRS algorithm starts.

STEP 2 (While ($n < N_{\rm max}$)) is a while loop over $N_{\rm max}$, i.e., the maximum number of costly evaluations.

• STEP 2.1 (Fit/Update Response Surface Model) generates or updates the response surface model $s_n(\mathbf{x})$ relying on the *n* already evaluated and costly vector points $(\mathbf{x}_i, f(\mathbf{x}_i))$, for i = 1, ..., n. *surrogateopt* applies for the response surface model a radial basis function interpolator s_n based on cubic splines with a linear tail *p* as introduced by Gutmann (2001b) in order to minimize the measure of bumpiness:

$$s_n(\mathbf{x}) = \sum_{i=1}^n \lambda_i \|\mathbf{x} - \mathbf{x}_i\|^3 + p(\mathbf{x}).$$
(A.1)

 $\|.\|$ is the Euclidean norm, i.e., the L₂-norm. The polynomial $p(\mathbf{x})$ is defined based on the basis $\mathcal{B} = (\mathbf{b}_0 = 1, \mathbf{b}_1, \dots, \mathbf{b}_d)$ with their associated real number coefficients c_0, \dots, c_d . \mathcal{B} is the basis of the linear space $\Pi^{d \times (d+1)}$ containing the linear polynomials formed by each d input parameters of the considered problem. The real

number coefficients $\lambda = (\lambda_1, ..., \lambda_n)^T$ and polynomial coefficients $\mathbf{c} = (c_0, ..., c_d)^T$ are determined by solving the system:

$$\begin{bmatrix} \boldsymbol{\Phi} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \lambda \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{0} \end{bmatrix} , \qquad (A.2)$$

where $\boldsymbol{\Phi}$ is the matrix $\boldsymbol{\Phi}_{kl} = \|\mathbf{x}_k - \mathbf{x}_l\|^3$ for k, l = 1, ..., n and **P** has the coefficients $P_{kl} = \mathbf{b}_l(\mathbf{x}_k)$ for k = 1, ..., n and l = 0, ..., d. The vector **F** is composed of the *n* expensive evaluations of the objective function *f*, i.e., $\mathbf{F} = (f(\mathbf{x}_1), ..., f(\mathbf{x}_n))^T$.

- STEP 2.2 (Randomly Generate Candidate Points) has the important role to explore the domain of evaluation within the bounds. In order to realize that, a large amount of possible candidate points are randomly generated by samplers around the incumbent point, i.e., the evaluated point with the best function value. The procedure can be related to a local search. In order to explore the neighborhood (or *trust region* of radius ρ_{trust}) as good as possible in different directions, several samplers are applied in a cycle: The random sampler (in which the random number generator is set to the Matlab[®] default *Mersenne Twister*) is applied twice in a row and generates random points according to a Gaussian distribution with zero mean and a standard deviation set to ρ_{trust} . Then, the mesh adaptive direct search OrthoMADS (Abramson et al., 2009) followed by the Generalized Pattern Search (GPS) (Torczon, 1997) sampler come into play. Accordingly, the generation of the random candidate points is centered around the incumbent point and within the trust region of radius ρ_{trust} . The scaling factor ρ_{trust} is set to 0.2 times the size of the box specified by the input parameter bounds. It doubles (up to 0.8) in case of a successful search for improving the objective function and halves (down to 10^{-5}) in case of an unsuccessful search for improving the objective function.
- Step 2.3 (Select the Next Function Evaluation Point) determines which candidate point is the best to be evaluated by the costly objective function f. To proceed a so-called *merit* function relying on the previously generated response surface model $s_n(\mathbf{x})$ is computed at every random candidate points generated in step 2.2, except those too close to the previously costly evaluated points. A minimum distance d_{\min}^{sample} is set by default and can be adapted by the user. The *merit* function implemented in the *surrogateopt* function is the weighted sum of the response surface value $S(\mathbf{x})$ and the distance from the evaluated points $D(\mathbf{x})$ proposed by Regis and Shoemaker (2007b):

$$f_{\text{merit}}(\mathbf{x}) = wS(\mathbf{x}) + (1 - w) D(\mathbf{x}) \quad \text{with} \quad S(\mathbf{x}) = \frac{s_n(\mathbf{x}) - s_{\min}}{s_{\max} - s_{\min}} \text{ and}$$
$$D(\mathbf{x}) = \frac{d_{\max} - d(\mathbf{x})}{d_{\max} - d_{\min}}.$$
(A.3)

 $s_n(\mathbf{x})$ is the current RBF response surface model computed for the candidate point x, s_{\min} the minimum surrogate value among the candidate points and s_{\max} the maximum. Defining the distance d_{ii} by the L₂-norm between a candidate point *i* and an evaluated point j, d_{\min} and d_{\max} are the minimum and maximum of d_{ij} over all candidate points *i* and evaluated points *j*. $d(\mathbf{x})$ is the minimum of the distance between the candidate point \mathbf{x} and all evaluated points. Since the vector \mathbf{x} is normalized by the bounds in each dimension, all distances are dimensionless as the userparameter d_{\min}^{sample} . The transition between a local search around the current *incumbent* point and a more global search depends on the value of the weight w: A large w encourages the local refining of the current best value, while a smaller w encourages the global exploration. Since Steps 2.2 and 2.3 are nested in the surrogateopt implementation, the value of w is adapted within the sampler cycle from 0.3 to 0.95 following the suggestion by Regis and Shoemaker (2007b) (see the documentation for details). The

³ https://dtecbw.de

⁴ https://de.mathworks.com/products/matlab.html



Fig. 7. Effect of the parallel worker number i_w on the sample creation for $d_{\min}^{\text{sample}} = 10^{-3}$. Incumbent among initial random points highlighted in blue, adaptive samples depicted by gray symbols, worst case depicted by a diamond highlighted by a square in red.

next function evaluation point \mathbf{x}_{n+1} is the one with the highest f_{merit} value. This point is called an *adaptive* point.

If all candidate points are too close to the evaluated points, i.e., below the given minimum sample distance d_{\min}^{sample} , no new *adaptive* point is defined and Steps 2.4 and 2.5 are skipped. In that specific case a local minimum is reached. If *n* is lower than N_{\max} , a new search begins at **STEP 1** based on other random points associated with a *reset of the surrogate*.

- Step 2.4 (Do Costly Function Evaluation) contains the expensive evaluation of the objective function f at the point \mathbf{x}_{n+1} .
- Step 2.5 (Update Information): The new evaluated point becomes a part of the pool available for the surrogate. If the new evaluated point possesses an objective function $f(\mathbf{x}_{n+1})$ lower than the current one from the *incumbent* $f(\mathbf{x}_n^*)$, an update of the best function value is carried out: $\mathbf{x}_{n+1}^* = \mathbf{x}_{n+1}$ and $f_{n+1}^* = f(\mathbf{x}_{n+1}^*)$. n+1 is stored in n.

When the current number of expensive evaluations reaches $N_{\rm max}$, the MSRS algorithm ends with **STEP 3** and returns the best solution found.

An optimization run using *surrogateopt* is also composed of several local searches to determine the global optimum. Since f_{merit} penalizes any new candidate point that is close to the previously evaluated points, the iterates provided by the MSRS algorithm can stay during the second (or higher) pass away from the minimum found during the first surrogate pass. The minimum is highly influenced by the initial surrogate and the incumbent, so a reset gives another chance to find the global minimum.

The optimization function has the option to use a kind of *parallelism* to strongly reduce the wall-clock time as in Regis and Shoemaker (2007a). A fixed number of parallel workers i_w is defined by the user. The routine performs its main algorithm (generation of surrogate, local search, ...) on the host, not on the parallel workers. Based on the latest surrogate, the parallel implementation fills a queue of *adaptive* points to

be evaluated by the expensive objective function. The Matlab® scheduler assigns the adaptive points from the queue to workers as those become available. These workers act asynchronously. When all workers return one evaluation of the time-consuming objective function, the RBF surrogate is updated and a local search is started. When the algorithm decides to switch between phases (for example from STEP 1 to STEP 2.1 or from STEP 2.5 to STEP 2.1), the points being evaluated remain in service, and any other points in the queue are discarded from the queue. Therefore, generally, the number of initial random points n_0^{parallel} that the parallel algorithm creates in STEP 1 is at least n_0 but at most $n_0 + i_w$. In practice, the expensive objective function can be defined in Matlab[®] as a wrapper function containing the following steps: Preprocessing of the case, start of the FSI solver through the job scheduling system available on the cluster in use (such as Slurm⁵) and postprocessing. In that way, optimization runs based on the parallel version can be carried out with i_w costly evaluations treated simultaneously by the compute cluster as sketched in Fig. 1.

Appendix B. Uncertainties in the overall workflow

Due to modeling, discretization, and computational errors, the results obtained from the computational framework have a certain level of uncertainty. Given that the present work is built upon numerous preceding publications, this section aims to provide a comprehensive overview of various sources of uncertainty from the fluid simulation part, the gust modeling part, the structure simulation part, the geometry representation, the coupling part and the optimization part.

B.1. Fluid simulation

As mentioned in Section 2.2, both fluid and structure solvers apply well-established schemes leading to a second-order accuracy in time

⁵ https://slurm.schedmd.com

and space. In order to achieve a good compromise between low computational errors and low CPU costs, the variables are stored in double precision and the limit of the residuum for both solvers is set to 10^{-10} . A CFD grid sensitivity study was carried out for the rigid model of the hemisphere. In Wood et al. (2016) the time-averaged predictions were compared with the time-averaged measurements delivering a very good agreement not only for the velocity components, but also for the Reynolds stresses, proving that the uncertainties arising from the discretization of the fluid equations were kept low. The turbulence modeling and the influence of the subgrid-scale model on the flow around the rigid hemisphere were also investigated in Wood et al. (2016), leading to the conclusion that the classical Smagorinsky model with $C_s = 0.1$ delivers nearly identical results as the dynamic Smagorinsky model usually referred to. In order to reduce the propagation of uncertainties in the whole problem, one set of turbulence inflow data containing a large number of time steps was generated by the recursive digital-filter technique of Klein et al. (2003). This set is used for all simulations of the same Reynolds number. Thus, if two simulations restart from the same restart file and use the same set of turbulence inflow data (useful for an optimization run), the outcome will be identical.

B.2. Gust modeling

Given that the gust modeling applied in the present framework only describes the underlying physics approximately, it inherently introduces modeling uncertainties. Unfortunately, due to a lack of experimental data no direct evaluation of these uncertainties is currently possible. However, this rather simple gust modeling is a must for control and optimization, since it is deterministic, i.e., for a given set of gust parameters the same gust shape is always achieved. The gust shape is superimposed on the underlying turbulent inflow. The time instant at which the gust is superimposed is an important source of uncertainty as highlighted in De Nayer et al. (2022). Indeed, the addition of a fixed gust velocity amplitude to a fluctuating background velocity mimicking turbulence leads to variations of up to 19% in the maximal force coefficients and up to 6% in the maximal von Mises stresses. Therefore, within the optimization run it is important to select the same superposition instant on the same set of turbulence inflow data.

B.3. Structure simulation

Concerning the structure, a rather simple material modeling, i.e., the St. Venant-Kirchhoff material, was selected to assure an easy reproducibility for the community. The associated material parameters, the Young's modulus E_{silicone} and the Poisson's ratio v_{silicone} were experimentally determined by Wood et al. (2018). The value of E_{silicone} is closely approximated by a linearization of the stress/strain relation under the assumption that the strain values are below 10%. In the present setup, the situations involving strong gusts ($A_g/u_{\infty} = 1.5$) have a maximum strain of 6.5 %. Thus, the St. Venant-Kirchhoff material modeling can be applied. During the experiments of Wood et al. (2018) several flexible membranes were manufactured by a casting process in order to evaluate the reproducibility of the results. That leads to a unique value for the material density $\rho_{\rm silicone}$ to complete the material modeling. The wall thickness of the membrane was measured by a micrometer with a precision of 5 µm and two different contact-free laser-based methods delivering a unique averaged value. For the structure simulation, particular attention was paid to the initial pre-stress of the membrane, recognizing its potential to introduce uncertainty. Since it determines the initial state of equilibrium of the membrane, the structural response during the FSI case strongly depends on it. For the present setup, the analytical Barlow's formula is available delivering a first approximation of the initial pre-stress. Starting with this value, a calibration was performed in De Nayer et al. (2018a) to achieve a

close agreement with measurements from a steady-state test mitigating the uncertainty in that part of the structure modeling. As the coupled FSI scenario is strongly time-dependent, the structural damping in the membrane has to be taken into account. The material employed for thin-walled structures often leads to structural damping exhibiting a non-linear behavior. However, to coincide with the approach used for simple material modeling, the well-established Rayleigh damping technique was applied and its coefficients were determined in De Naver et al. (2018a) to achieve the decay observed in the displacement amplitude of the experimental dynamic structure test introduced by Wood et al. (2018). The combination of the St. Venant-Kirchhoff material and the Rayleigh damping model yields satisfying results in terms of displacement response and its associated main frequencies on this dynamic structure test. The uncertainty arising from structural finiteelement discretization was minimized through a grid-independence study conducted in De Nayer et al. (2018a).

B.4. Geometry representation

The representation of the geometry introduces two major sources of error to the investigated case. First, the surface of the body in the simulation is considered as smooth. In the experiments the surface roughness was measured and a value of $R_a = 0.8 \,\mu\text{m}$ was determined for the rigid hemisphere in Wood et al. (2016) and $R_a = 16 \,\mu\text{m}$ for the flexible membrane in Wood et al. (2018). According to the relatively low values, the surface roughness was not taken into account in the simulations. Second, the curvy shape of the hemisphere, is not an issue in case of pure wall-resolved LES for the body-fitted fluid solver, since a fine grid is required. This changes in case of coupled simulations, as the mapping procedure at the FSI interface links the structural discretization to the fluid counterpart. Due to the typically coarser nature of finite element-based structural grids compared to fluid grids, the structural elements dictates the quality of the FSI interface. As a consequence, non-physical kinks appear on the fluid side, which generates a kind of an artificial roughness. This phenomenon was investigated in Apostolatos et al. (2019). A good compromise in terms of geometry representation and low CPU costs for the present case is a medium-sized finite-element structural mesh. Even for large deflections of the membrane, the kinks visible at the interface (see Fig. 3(c)) do not significantly affect the fluid flow. For a perfect representation of the FSI interface, an isogeometric discretization of the membrane or an isogeometric mapping introducing an exact coupling layer as proposed by Apostolatos et al. (2019) should be considered.

B.5. Coupling scheme

The coupling between the fluid and the structure domains can be a further source of uncertainties at different levels. First, the coupling algorithm itself. For the present setup a loose coupling algorithm is sufficient, since air (i.e., the fluid medium) is much lighter than the structure. In a precursor study, a coupled FSI simulation was carried out with the strong coupling algorithm delivering the same outcome (force coefficients, membrane deflections, ...). Therefore, the loose coupling is not a significant source of uncertainty in the present case. The mapping applied at the FSI interface between the fluid and structure discretizations is another source of uncertainty for the transferred data. In order to correctly predict highly transient FSI phenomena, the total transferred fluid load experienced by the body is important. However, the correct spatial distribution of the mapped fluid loads is even more relevant in the present case, where local gusts of diverse amplitudes hit the structure. Therefore, the dual mortar mapping method developed by Wang et al. (2016) and extended in Apostolatos et al. (2019) is applied. It ensures the consistency of the fluid load along the FSI interface. The relative error in the total fluid force made during the transfer stays below 1.5% The maximal relative error in the mapped displacements is notably smaller staying below 0.01 %.

Nevertheless, the close agreement between the predicted timeaveraged deflections of the flexible hemisphere (without gust but at three different turbulent inflows) and its related measurements, as well as the correct predictions of the instantaneous phenomena observed during the experiments, show that modeling uncertainties, discretization errors and coupling errors were kept low (see De Nayer et al. (2018a)).

B.6. Optimization procedure

Regarding the MSRS optimization procedure, the following conclusions can be drawn. If the user sets the parameter $N_{\rm max}$ such that enough evaluations of the objective function are carried out, only the parameter d_{\min}^{sample} can introduce uncertainty into the optimization outcome. A large value of d_{\min}^{sample} can restrict the search for the minimum, hindering the determination of the worst-case scenario. The reproducibility of the optimization outcome is also an issue, since the Matlab documentation states about the surrogateopt function: "Parallel surrogate optimization does not necessarily give reproducible results, due to the non reproducibility of parallel timing, which can lead to different execution paths". For each configuration presented, three runs involving the same parameters were always started on the same compute cluster, nearly leading to the same optimization paths and always determining the same worst-case scenario. Thus, the results were reproducible and no uncertainty on the outcome was observed regarding the non-reproducibility of parallel timing.

References

- Abdelaziz, K.M., Alipour, A., Hobeck, J.D., 2021. A smart façade system controller for optimized wind-induced vibration mitigation in tall buildings. J. Wind Eng. Ind. Aerodyn. 212, 104601.
- Abramson, M.A., Audet, C., Dennis, Jr., J.E., Le Digabel, S., 2009. OrthoMADS: A deterministic MADS instance with orthogonal directions. SIAM J. Optim. 20 (2), 948–966.
- Apostolatos, A., De Nayer, G., Bletzinger, K.-U., Breuer, M., Wüchner, R., 2019. Systematic evaluation of the interface description for fluid-structure interaction simulations using the isogeometric mortar-based mapping. J. Fluids Struct. 86, 368–399.
- Audet, C., Le Digabel, S., Rochon Montplaisir, V., Tribes, C., 2006. The NOMAD project. Software available at https://www.gerad.ca/nomad/.
- Audet, C., Le Digabel, S., Rochon Montplaisir, V., Tribes, C., 2022. Algorithm 1027: NOMAD version 4: Nonlinear optimization with the MADS algorithm. ACM Trans. Math. Softw. 48 (3), 35:1–35:22.
- Basar, Y., Weichert, D., 2013. Nonlinear Continuum Mechanics of Solids: Fundamental Mathematical and Physical Concepts. Springer Science & Business Media.
- Bernardini, E., Spence, S.M.J., Wei, D., Kareem, A., 2015. Aerodynamic shape optimization of civil structures: A CFD-enabled Kriging-based approach. J. Wind Eng. Ind. Aerodyn. 144, 154–164.
- Bletzinger, K.-U., Wüchner, R., Kupzok, A., 2006. Algorithmic treatment of shells and free form-membranes in FSI. In: Bungartz, H.-J., Schäfer, M. (Eds.), Fluid-Structure Interaction. In: Lecture Notes in Computational Science and Engineering, LNCSE, vol. 53, Springer, Heidelberg, pp. 336–355.
- Breitenberger, M., Apostolatos, A., Philipp, B., Wüchner, R., Bletzinger, K.-U., 2015. Analysis in computer aided design: Nonlinear isogeometric B-Rep analysis of shell structures. Comput. Methods Appl. Mech. Engrg. 284, 401–457.
- Breuer, M., 2018. Effect of inflow turbulence on an airfoil flow with laminar separation bubble: An LES study. Flow Turbul. Combust. 101 (2), 433–456.
- Breuer, M., De Nayer, G., Münsch, M., Gallinger, T., Wüchner, R., 2012. Fluid-structure interaction using a partitioned semi-implicit predictor-corrector coupling scheme for the application of large-eddy simulation. J. Fluids Struct. 29, 107–130.
- De Nayer, G., Apostolatos, A., Wood, J.N., Bletzinger, K.-U., Wüchner, R., Breuer, M., 2018a. Numerical studies on the instantaneous fluid-structure interaction of an air-Inflated flexible membrane in turbulent flow. J. Fluids Struct. 82, 577–609.
- De Nayer, G., Breuer, M., 2014. Numerical FSI investigation based on LES: Flow past a cylinder with a flexible splitter plate involving large deformations (FSI-PfS-2a). Int. J. Heat Fluid Flow 50, 300–315.
- De Nayer, G., Breuer, M., 2020. A source-term formulation for injecting wind gusts in CFD simulations. J. Wind Eng. Ind. Aerodyn. 207, 104405.De Nayer, G., Breuer, M., 2022. Assessment of discrete wind gust parameters: Towards

Wind Eng. Ind. Aerodyn. 231, 105207.

the worst-case scenario of a FSI application in form of an inflated hemisphere. J.

- 103462.
 De Nayer, G., Breuer, M., Perali, P., Grollmann, K., 2019. Modeling of wind gusts for large-eddy simulations related to fluid-structure interactions. In: Salvetti, M., Armenio, V., Fröhlich, J., Geurts, B.J., Kuerten, H. (Eds.), ERCOFTAC Series, Direct and Large-Eddy Simulation: DLES-11, May 29–31, 2017, Vol. 25. Springer Nature Switzerland AG 2019, Pisa, Italy, pp. 453–459.
- De Nayer, G., Kalmbach, A., Breuer, M., Sicklinger, S., Wüchner, R., 2014. Flow past a cylinder with a flexible splitter plate: A complementary experimental-numerical investigation and a new FSI test case (FSI-PfS-1a). Comput. & Fluids 99, 18–43.
- De Nayer, G., Schmidt, S., Wood, J.N., Breuer, M., 2018b. Enhanced injection method for synthetically generated turbulence within the flow domain of Eddy-resolving simulations. Comput. Math. Appl. 75 (7), 2338–2355.
- Ding, F., Kareem, A., 2018. A multi-fidelity shape optimization via surrogate modeling for civil structures. J. Wind Eng. Ind. Aerodyn. 178, 49–56.
- Durst, F., Schäfer, M., 1996. A parallel block-structured multigrid method for the prediction of incompressible flows. Int. J. Numer. Methods Fluids 22 (6), 549–565.
- Egea, J.A., Vazquez, E., Banga, J.R., Martí, R., 2009. Improved scatter search for the global optimization of computationally expensive dynamic models. J. Global Optim. 43 (2), 175–190.
- Ferziger, J.H., Perić, M., 2002. Computational Methods for Fluid Dynamics, third ed. Springer Berlin.
- Goertz, S., Abu-Zurayk, M., Ilic, C., Wunderlich, T.F., Keye, S., Schulze, M., Kaiser, C., Klimmek, T., Süelözgen, Ö., Kier, T., et al., 2020. Overview of collaborative multifidelity multidisciplinary design optimization activities in the DLR project VicToria. In: AIAA Aviation 2020 Forum. p. 3167.
- Gutmann, H.M., 2001a. Radial Basis Function Methods for Global Optimization (Ph.D. thesis). University of Cambridge, Cambridge, England.
- Gutmann, H.M., 2001b. A radial basis function method for global optimization. J. Global Optim. 19 (3), 201–227.
- Guzman Nieto, M., ElSayed, M.S., Walch, D., 2019. Efficient global optimization and modal strain energy coefficient-based algorithm for fast prediction of dynamic aeroelastic loads. Struct. Multidiscip. Optim. 60 (2), 817–834.
- IEC-Standard, 2002. 61400-21. Measurement and Assessment of Power Quality of Grid Connected Wind Turbines. Technical Report.
- Jones, D.R., Schonlau, M., Welch, W.J., 1998. Efficient global optimization of expensive black-box functions. J. Global Optim. 13 (4), 455–492.
- Kalmbach, A., Breuer, M., 2013. Experimental PIV/V3V measurements of vortex– Induced fluid–structure interaction in turbulent flow — A new benchmark FSI-PfS–2a. J. Fluids Struct. 42, 369–387.
- Khodaparast, H.H., Georgiou, G., Cooper, J., Riccobene, L., Ricci, S., Vio, G., Denner, P., 2012. Efficient worst case 1-cosine gust loads prediction. J. Aeroelasticity Struct. Dyn. 2 (3), 33–54.
- Klein, M., Sadiki, A., Janicka, J., 2003. A digital filter based generation of inflow data for spatially-developing direct numerical or large–eddy simulations. J. Comput. Phys. 186, 652–665.
- Kusano, I., Cid Montoya, M., Baldomir, A., Nieto, F., Jurado, J.À., Hernàndez, S., 2020. Reliability based design optimization for bridge girder shape and plate thicknesses of long-span suspension bridges considering aeroelastic constraint. J. Wind Eng. Ind. Aerodyn. 202, 104176.
- Li, R., Xu, P., Peng, Y., Ji, P., 2016. Multi-objective optimization of a high-speed train head based on the FFD method. J. Wind Eng. Ind. Aerodyn. 152, 41–49.
- Lophaven, S.N., Nielsen, H.B., Søndergaard, J., 2002. A Matlab Kriging toolbox. Technical Report No. IMM-TR-2002-12, Citeseer, Technical University of Denmark, Kongens Lyngby.
- Muñoz-Paniagua, J., Garcia, J., 2019. Aerodynamic surrogate-based optimization of the nose shape of a high-speed train for crosswind and passing-by scenarios. J. Wind Eng. Ind. Aerodyn. 184, 139–152.
- Neumann, P., Preuß, H., 2022. HSUper Technical Specifications. Technical Report, URL https://www.hsu-hh.de/hpc/en/hsuper/.
- Philipp, B., Wüchner, R., Bletzinger, K.-U., 2016. Advances in the form-finding of structural membranes. Proceedia Eng. 155, 332–341.
- Phoenix Integration, 2017. ModelCenter 12.0. URL http://www.phoenix-int.com/ software/phx-modelcenter.php.
- Piomelli, U., Chasnov, J.R., 1996. Large–eddy simulations: Theory and applications. In: Hallbäck, M., Henningson, D.S., Johansson, A.V., Alfredson, P.H. (Eds.), Turbulence and Transition Modeling. Kluwer, pp. 269–331.
- Regis, R.G., Shoemaker, C.A., 2007a. Parallel radial basis function methods for the global optimization of expensive functions. European J. Oper. Res. 182 (2), 514–535.
- Regis, R.G., Shoemaker, C.A., 2007b. A stochastic radial basis function method for the global optimization of expensive functions. INFORMS J. Comput. 19 (4), 497–509.
- Rhie, C.M., Chow, W.L., 1983. Numerical study of the turbulent flow past an airfoil with trailing-edge separation. AIAA J. 21 (11), 1525–1532.
- Ripepi, M., Verveld, M.J., Karcher, N.W., Franz, T., Abu-Zurayk, M., Görtz, S., Kier, T.M., 2018. Reduced-order models for aerodynamic applications, loads and MDO. CEAS Aeronaut. J. 9 (1), 171–193.

G. De Nayer and M. Breuer

- Schlatter, P., Orlu, R., Li, Q., Brethouwer, G., Fransson, J.H.M., Johansson, A.V., Alfredsson, P.H., Henningson, D.S., 2009. Turbulent boundary layers up to $Re_{\theta} = 2500$ studied through simulation and experiment. Phys. Fluids 21 (5), 51702.
- Schmidt, S., Breuer, M., 2017. Source term based synthetic turbulence inflow generator for Eddy–Resolving predictions of an airfoil flow including a laminar separation bubble. Comput. & Fluids 146, 1–22.
- Sen, S., De Nayer, G., Breuer, M., 2017. A fast and robust hybrid method for blockstructured mesh deformation with emphasis on FSI-LES applications. Int. J. Numer. Methods Eng. 111 (3), 273–300.
- Sicklinger, S., Belsky, V., Engelmann, B., Elmqvist, H., Olsson, H., Wüchner, R., Bletzinger, K.-U., 2014. Interface Jacobian-based co-simulation. Int. J. Numer. Methods Eng. 98 (6), 418–444.
- Smagorinsky, J., 1963. General circulation experiments with the primitive equations I: The basic experiment. Mon. Weather Rev. 91 (3), 99–165.
- Torczon, V., 1997. On the convergence of pattern search algorithms. SIAM J. Optim. 7 (1), 1–25.
- Turek, S., Hron, J., Razzaq, M., Wobker, H., Schäfer, M., 2010. Numerical benchmarking of fluid-structure interaction: A comparison of different discretization and solution approaches. In: Bungartz, H.-J., Mehl, M., Schäfer, M. (Eds.), Fluid-Structure Interaction II – Modelling, Simulation, Optimization. In: Lecture Notes in Computational Science and Engineering, LNCSE, vol. 73, Springer, Heidelberg, pp. 413–424.
- Wang, T., Wüchner, R., Sicklinger, S., Bletzinger, K.-U., 2016. Assessment and improvement of mapping algorithms for non-matching meshes and geometries in computational FSI. Comput. Mech. 57 (5), 793–816.
- Wilke, G., 2019. Variable-fidelity methodology for the aerodynamic optimization of helicopter rotors. AIAA J. 57 (8), 3145–3158.
- Wood, J.N., Breuer, M., De Nayer, G., 2018. Experimental studies on the instantaneous fluid-structure interaction of an air-Inflated flexible membrane in turbulent flow. J. Fluids Struct. 80, 405–440.
- Wood, J.N., Breuer, M., Neumann, T., 2022. A novel approach for artificially generating horizontal wind gusts based on a movable plate: The paddle. J. Wind Eng. Ind. Aerodyn. 230, 105170.
- Wood, J.N., De Nayer, G., Schmidt, S., Breuer, M., 2016. Experimental investigation and large-eddy simulation of the turbulent flow past a smooth and rigid hemisphere. Flow Turbul. Combust. 97 (1), 79–119.