



**HAL**  
open science

# Single-step deep reinforcement learning for two- and three-dimensional optimal shape design

H. Ghraieb, J. Viquerat, A. Larcher, P. Meliga, E. Hachem

► **To cite this version:**

H. Ghraieb, J. Viquerat, A. Larcher, P. Meliga, E. Hachem. Single-step deep reinforcement learning for two- and three-dimensional optimal shape design. *AIP Advances*, 2022, 12 (8), pp.085108. 10.1063/5.0097241 . hal-03825017

**HAL Id: hal-03825017**

**<https://hal.science/hal-03825017>**

Submitted on 21 Oct 2022

**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.

# Single-step deep reinforcement learning for two- and three-dimensional optimal shape design

H. Ghraieb<sup>a</sup>, J. Viquerat<sup>a</sup>, A. Larcher<sup>a</sup>, P. Meliga<sup>a</sup>, E. Hachem<sup>a,\*</sup>

<sup>a</sup>*Mines Paris, PSL Research University, Centre de mise en forme des matériaux (CEMEF), CNRS UMR 7635, 06904 Sophia Antipolis Cedex, France*

---

## Abstract

This research gauges the capabilities of deep reinforcement learning (DRL) techniques for direct optimal shape design in computational fluid dynamics (CFD) systems. It uses Policy Based Optimization, a single-step DRL algorithm intended for situations where the optimal policy to be learnt by a neural network does not depend on state. The numerical reward fed to the neural network is computed with an in-house stabilized finite elements environment combining variational multi-scale (VMS) modeling of the governing equations, immerse volume method, and multi-component anisotropic mesh adaptation. Several cases are tackled in two and three dimensions, for which shapes with fixed camber line, angle of attack and cross-sectional area are generated by varying a chord length and a symmetric thickness distribution (and possibly extruding in the off-body direction). At zero incidence, the proposed DRL-CFD framework successfully reduces the drag of the equivalent cylinder (i.e., the cylinder of same cross-sectional area) by 48% at a Reynolds numbers in the range of a few hundreds. At an incidence of 30°, it increases the lift to drag ratio of the equivalent ellipse by 13% in two dimensions and 5% in three dimensions at a chord Reynolds numbers in the range of a few thousands. Although the low number of degrees of freedom inevitably constrains the range of attainable shapes, the optimal is systematically found to perform just as well as a conventional airfoil, despite DRL starting from the ground up and having no priori knowledge of aerodynamic concepts. Such results showcase the potential of the method for black-box shape optimization of practically meaningful CFD systems. Since the resolution process is agnostic to details of the underlying fluid dynamics, they also pave the way for a general evolution of reference shape optimization strategies for fluid mechanics and any other domain where a relevant reward function can be defined.

*Keywords:* Deep Reinforcement Learning; Artificial Neural Networks; Shape optimization; Computational fluid dynamics; Policy Based Optimization.

---

## 1. Introduction

Shape optimization is ubiquitous in engineering applications ranging from magnetostatics [1], acoustics [2], image restoration and segmentation [3], composite material identification [4] to nano-optics [5], just to name a few. Shape optimization in fluid mechanics dates back to the pioneering work of Pironneau on the minimization of energy loss in Stokes and Navier–Stokes flows [6, 7]. Since then, it has become an increasingly important research topic in the attempt to enhance drag reduction capabilities, which is due to the ever growing concerns on aerodynamic energy efficiency (to give a taste, reducing the overall drag by just a few percent while maintaining lift can help reducing fossil fuel consumption and CO2 emission while saving several billion dollars annually in ocean shipping or airline traffic [8]). In the following, the focus is essentially on airfoil shape optimization, a key component of aircraft flight mechanics that has come into prominence in a variety of other applications such as acoustic noise reduction [9] or energy harvesting [10]. One of the major challenges in the field is that the majority of flows of engineering interest are time-dependent

---

\*Corresponding author

*Email address:* elie.hachem[@]mines-paristech.fr (E. Hachem)

14 and even turbulent (e.g., fluttering, buffeting, dynamic stall), and therefore require sophisticated  
15 unsteady methods and optimization techniques, thus drastically increasing the computational cost.

16 Shape optimization has historically been tackled by two main classes of approaches, namely  
17 gradient-based and gradient-free methods. Gradient-based methods rely on the evaluation of the  
18 gradient of the objective function with respect to the design parameters. They have proven effective  
19 in large optimization spaces when said gradient is computed by the adjoint method [11–13],  
20 whose cost is comparable to that of solving the governing equation (unlike more computationally  
21 expensive alternatives such as variance-based and regression-based methods, in which the govern-  
22 ing equations need to be solved repeatedly, up to a hundred times). Nonetheless, gradient-based  
23 algorithms are easily trapped in local optima, meaning that the solution optimality can be very  
24 sensitive to the initial guess, all the more so when applied to stiff nonlinear problems [14]. Gradient-  
25 free methods are better equipped in this regard, but can be more complex to implement and to  
26 use. Among the available methods, genetic algorithms [15], particle swarm optimization [16] or  
27 metropolis algorithms [17] feature good global optimization capabilities, but they can be highly  
28 sensitive to heuristically chosen meta-parameters, plus their cost is usually higher and can easily  
29 exceed the available computational budget, thus limiting the number of design parameters [18]. It  
30 should be noted that both classes of methods can make use of cheap-to-evaluate surrogate models  
31 to approximate expensive objective and constraint functions without resorting systematically to  
32 numerical simulations [19]. Several approaches exist for building such surrogate models, e.g., poly-  
33 nomial response surfaces, radial basis functions, kriging, or supervised artificial neural networks  
34 [20], for which geometric parametrization plays a determinant role, in terms of both the attainable  
35 geometries and the tractability of the optimization process [21].

36 The premise of this research is that the related task of selecting an optimal subset of design  
37 parameters can alternatively be assisted using deep reinforcement learning (DRL). DRL is the  
38 advanced branch of machine learning that couples deep neural networks (DNNs, a family of versatile  
39 tools that can learn how to hierarchically extract informative features from data, and have gained  
40 traction as efficient computational processors for performing a variety of tasks, from exploratory  
41 data analysis to qualitative and quantitative predictive modeling) and reinforcement learning, a  
42 class of decision-making algorithms that can autonomously learn effective policies for sequential  
43 decision problems. In practice, DRL involves DNNs learning how to behave in an environment so  
44 as to maximize some notion of long-term reward, a task compounded by the fact that each action  
45 taken affects both immediate and future rewards. The feature extraction capabilities of DNNs, as  
46 well as their ability to handle quasi-arbitrary nonlinear input/output mappings, have lifted several  
47 major obstacles that hindered classical reinforcement learning and has led unprecedented efficiency  
48 in the context of nonlinear optimal control problems with high-dimensional state spaces. Several  
49 notable works using DRL in mastering games (e.g., Go, Poker) have stood out for attaining super-  
50 human level [22, 23], but the approach has also breakthrough potential for practical applications  
51 such as robotics [24, 25], computer vision [26], finance [27], autonomous cars [28, 29], or data center  
52 cooling [30].

53 The efforts for applying DRL to fluid mechanics are ongoing but still at an early stage, as  
54 recently reviewed in [31]. Nonetheless, the domain has undergone a large inflow of contributions  
55 with clear focus on drag reduction problems [32–44]. This enthusiasm is likely due to the increasing  
56 number of open-source initiatives [32, 45, 46], that has led to an accelerated diffusion of the methods  
57 in the community, and to the sustained commitment from the machine learning community, that  
58 has allowed concurrently expanding the scope from computationally inexpensive, low-dimensional  
59 reductions of the underlying fluid dynamics to complex Navier–Stokes systems [47, 48], all the  
60 way to experimental set-ups [49]. A handful of studies have recently provided insight into the  
61 performance improvements to be delivered in shape optimization, but it is worth emphasizing that  
62 figuring out a fixed shape that best meets a set of required criteria (e.g., high lift-to-drag ratio, low  
63 pressure loss) requires optimizing state-independent parameters, which is not *per se* the original  
64 purpose of DRL. Nonetheless, two main classes of methods have emerged in the community, namely  
65 the direct and incremental approaches. The incremental approach uses the state-to-action mapping  
66 as a way to incrementally modify an initial shape into an optimal one [50–53], which exploits  
67 the capabilities of the DRL paradigm (in which network updates are performed after multi-step  
68 episodes) in performing active flow control. The direct approach [46] conversely relies on single-step  
69 DRL, a subset of DRL in which network updates are performed after one-step episodes (hence the

70 *stateless* moniker), and builds on recent efforts to assess the relevance of DRL in the context of  
71 open-loop control [41, 54].

72 This research introduces a novel framework combining single-step reinforcement learning with  
73 immersed methods for fluid flow shape optimization, that exploits both the ability of neural net-  
74 works to learn to approximate arbitrarily well the mapping function between input and output  
75 spaces, and the dynamic programming built in the reinforcement learning algorithm. It is a follow-  
76 up on to our contribution showcasing the first application of DRL to direct shape optimization [46].  
77 It uses Policy Based Optimization (PBO [55]), a novel single-step algorithm developed in-house,  
78 that improves the convergence rate of the previously used single-step Proximal Policy Optimiza-  
79 tion (PPO [24]) algorithm by adopting several key heuristics from the covariance matrix adaptation  
80 evolution strategy (CMA-ES). In short, PBO learns the mean, variance and correlation parameters  
81 of a multivariate normal search distribution from three separate neural networks, while single-step  
82 PPO updates the mean and variance (the same for all variables) from a single network, which  
83 can prematurely shrink the exploration variance. The objective is twofold: first, to further shape  
84 the capabilities of PBO for fluid mechanics applications (as it has so far been limited to textbook  
85 problems of analytic functions minimization), to help narrow the gap between DRL and advanced  
86 numerical methods for multi-scale, multi-physics computational fluid dynamics (CFD). Second, to  
87 gauge the feasibility of learning optimal designs from a low, yet suitable number of design pa-  
88 rameters, for which Bézier curves, B-splines and NURBS are good candidates. We believe this  
89 is chief to mitigate the computational burden without deteriorating the geometric accuracy, since  
90 the parametrization in the direct approach provides a complete description of the shape itself,  
91 not that of a perturbation to a reference shape. The PBO agent is trained on high-fidelity CFD  
92 simulations, in contrast to most aforementioned studies about incremental shape optimization, in  
93 which a pre-trained surrogate or a simplified model is used for full agent training, or to perform  
94 an initial learning phase before re-training on a CFD environment using transfer learning. This is  
95 because the uncertainty of surrogate models cannot be quantified during optimization, which may  
96 misguide policy updating. We insist that it lies out of the scope of this paper to provide exhaus-  
97 tive performance comparison data against state-of-the art optimization techniques (e.g., evolution  
98 strategies or genetic algorithms). This would indeed require a tremendous amount of time and  
99 resources even though the efforts for developing the method remain at an early stage. Nonetheless,  
100 it is worth mentioning that PBO is shown in [55] to compare well against standard CMA-ES and  
101 to significantly outperform our previous PPO-based single-step algorithm, even though new algo-  
102 rithms cannot be expected to reach right away the level of performance of their more established  
103 counterparts.

104 The organization is as follows: section 2 introduces PBO (together with the baseline principles  
105 of DRL and single-step DRL), and outlines the main features of the finite element CFD environment  
106 used to compute the numerical reward fed to the neural networks. Section 3 revisits the classical  
107 problem of finding the two-dimensional shapes minimizing drag in a uniform flow for the purpose  
108 of validation and assessment part of the method capabilities. In section 4, PBO is applied to more  
109 meaningful aerodynamic optimization problems consisting of finding the two-dimensional shapes  
110 maximizing the lift to drag ratio in the context of turbulent flows at moderately large Reynolds  
111 number (in the range of a few thousands). Finally, an extension to three-dimensional shapes is  
112 proposed in section 5.

## 113 2. Methodology

### 114 2.1. Deep reinforcement learning

115 Reinforcement learning (RL) is a process by which an agent learns to earn rewards through  
116 trial-and-error interaction with its environment. At each turn, the agent observes the state  $s_t$  of  
117 the environment and takes an action  $a_t$ , that prompts both the transition to the next state  $s_{t+1}$  and  
118 the reward received  $r_t$ . This repeats until some termination state is reached, the core objective of  
119 the agent being to learn the succession of actions maximizing its cumulative reward over an episode  
120 (this is the reference unit for agent update, best understood as one instance of the scenario in which  
121 it takes actions). In a deep reinforcement learning context (deep RL or DRL), the agent is a deep  
122 neural network (DNN) patterned after the neural circuits formed by neurons in human brains.

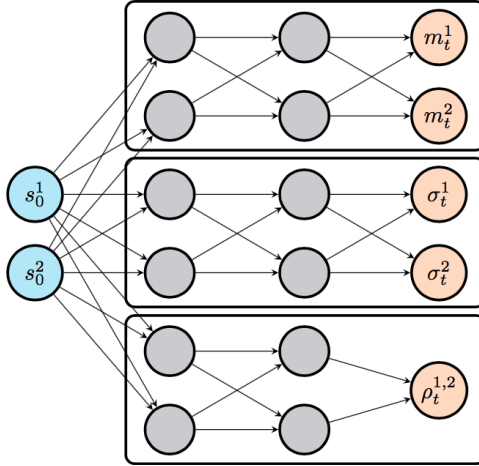


Figure 1: Policy networks used in PBO to map states to policy. Three networks trained separately are used for the prediction of mean, standard deviation, and correlation parameters. Orthogonal weights initialization is used throughout the networks, with a unit gain for all layers except the output layers, for which the gain is set to  $10^{-2}$ .

123 The most general form of neural network architecture is the fully connected DNN, in which the  
 124 processing units (the artificial neurons) are stacked in layers and information propagates forward  
 125 from the input layer to the output layer via “hidden” layers. Each neuron performs a weighted sum  
 126 of its inputs to assign significance with regard to the task the algorithm is trying to learn, adds a  
 127 bias to figure out the part of the output independent of the input, and feeds an activation function  
 128 that determines whether and to what extent the computed value should affect the outcome. The  
 129 neural network learns to represent the relation between input (action) and output (reward) data  
 130 by repeatedly adjusting the weights and biases by back-propagation, from the output layer back  
 131 through the hidden layers to the input layer (a process known as training).

132 *2.2. Single-step deep reinforcement learning*

133 Single-step DRL is a subset of DRL that has recently emerged from the premise that tweaked  
 134 versions of regular DRL algorithms can be used as black-box optimizers. The underlying idea is  
 135 that it may be enough for the agent to interact only once per episode with its environment (hence,  
 136 single-step episodes, and by extension, single-step DRL) if the optimal behavior to be learnt is  
 137 independent of state, as is notably the case in optimization and open-loop control problems. The  
 138 novelty of the approach can be summed up as follows: in DRL, a DNN learns the optimal set of  
 139 observation-based actions  $a^*$  yielding the largest possible reward. In single-step DRL, it learns  
 140 instead the optimal mapping  $f_{\theta^*}$  such that  $a^* = f_{\theta^*}(s_0)$ , where  $s_0$  is some input state (usually  
 141 a constant vector) repeatedly fed to the agent for the optimal policy to eventually embody the  
 142 transformation from  $s_0$  to  $a^*$ . A direct consequence is that single-step DRL algorithms can use  
 143 much smaller networks (compared to the usual agent architecture used in other DRL contributions),  
 144 because the agent is not required to learn a complex state-action relation, but only a transformation  
 145 from a constant input state to a given action.

146 *2.3. Policy based optimization*

147 The present research relies on policy-based optimization (PBO) a single-step, model free, off-  
 148 policy gradient RL algorithm whose key features are summarized as follows:

- 149 • the agent interacts with the environment itself, not a surrogate model of the environment  
 150 (model free, hence no assumptions about the fluid dynamics of the problems to be solved),
- 151 • its behavior is modeled after a parametrized probability distribution of actions  $\pi_{\theta}(a)$ , opti-  
 152 mized by gradient ascent (policy gradient),
- 153 • the agent is not required to sample the training data with the current policy (off-policy),

154 PBO draws actions from a  $d$ -dimensional multivariate normal distribution (with  $d$  the dimension of  
 155 the action required by the environment). A full co-variance matrix is used to improve the balance  
 156 between exploration and exploitation (the single-step PPO algorithm used in [46] conversely as-  
 157 sumes all variables to have the same variance and to be uncorrelated, which can prematurely shrink  
 158 the exploration variance). The co-variance matrix also accelerates convergence to the optimum by  
 159 aligning the contour of the sampling distribution with the contour lines of the objective function  
 160 and thereby the direction of steepest ascent.

161 As shown in figure 1, three independent neural networks output the necessary mean, standard  
 162 deviation, and correlation information, using hypersphere decomposition [56, 57] to generate valid  
 163 symmetric, positive semidefinite covariance matrices. Different meta-parameters and architectures  
 164 can be used for each network, which is shown in [55] to substantially impact the convergence  
 165 rate. Actions drawn in  $[-1; 1]^d$  are then mapped into relevant physical ranges, a step deferred  
 166 to the environment as being problem-specific. Finally, the Adam algorithm [58] runs stochastic  
 167 gradient ascent by computing adaptive learning rates (i.e., the step sizes to be taken in the gradient  
 168 direction) for each policy parameter, using the gradient of the loss function

$$L(\theta) = \mathbb{E}_{a \sim \pi_\theta} \left[ \max(\tilde{r}, 0) \log \pi_\theta(a) \right]. \quad (1)$$

169 In 1,  $\tilde{r}$  is the whitened reward normalized to zero mean and unit variance, considered a suit-  
 170 able advantage estimator. The rationale for this choice is as follows: as is customary in DRL,  
 171 the discounted cumulative reward is approximated by the advantage function, that measures the  
 172 improvement (if positive, otherwise the lack thereof) associated with taking action  $a$  in state  $s$   
 173 compared to taking the average over all possible actions. Because a single-step trajectory consists  
 174 of a unique state-action pair, the discount factor adjusting the trade-off between immediate and  
 175 future rewards can be set to unity, in which case the advantage reduces to the reward; see [41].  
 176 Substituting the whitened reward for  $r$  introduces bias but reduces variance, and thus the number  
 177 of actions needed to estimate the expected value. Finally, the max allows discarding negative-  
 178 advantage actions, that may destabilize learning when performing multiple mini-batch gradient  
 179 steps using the same data (as each step drives the policy further away from the sampled actions).

#### 180 2.4. Computational fluid dynamics environment

181 At the core of the CFD resolution framework is the in-house, CimLIB-CFD parallel finite  
 182 element library [59], whose main ingredients are as follows:

183 - the variational multiscale approach (VMS) is used to solve a stabilized weak form of the governing  
 184 equations using linear approximations ( $\mathbb{P}_1$  elements) for all variables, which otherwise breaks the  
 185 Babuska–Brezzi condition. The approach relies on an a priori decomposition of the solution into  
 186 coarse and fine scale components [60–62]. Only the large scales are fully represented and resolved at  
 187 the discrete level. The effect of the small scales is encompassed by consistently derived source terms  
 188 proportional to the residual of the resolved scale solution, hence ad-hoc stabilization parameters  
 189 comparable to local coefficients of proportionality.

190 - in laminar regimes, velocity and pressure come as solutions to the Navier–Stokes equations. In  
 191 turbulent regimes, the focus is on phase-averaged velocity and pressure modeled after the unsteady  
 192 Reynolds averaged Navier–Stokes (uRANS) equations. In order to avoid transient negative tur-  
 193 bulent viscosities, negative Spalart–Allmaras [63] is used as turbulence model, whose stabilization  
 194 proceeds from that of the convection-diffusion-reaction equation [64, 65].

195 - two-dimensional airfoil sections with fixed camber line are generated by varying a chord length  
 196 and a thickness distribution. The chord direction is constant, just as the angle of attack mea-  
 197 suring the incidence relative to the oncoming flow. The upper (suction/leeward) and lower (pres-  
 198 sure/windward) sides are discretized into  $n_p$  control points equally spaced in the camber line  
 199 direction. All shapes are closed and symmetrical with respect to the chord line, as achieved forcing  
 200 zero thickness at the edges and identical (half)-thicknesses at each forward and rearward facing  
 201 points. Consecutive points are connected by a cubic Bézier curve using local position and curvature

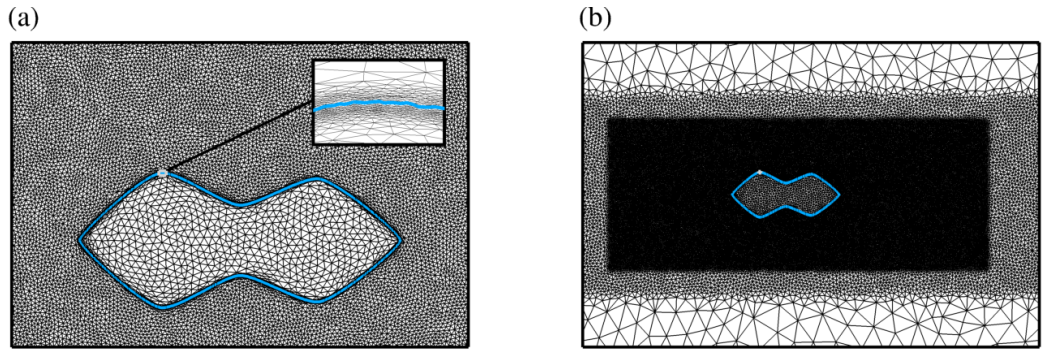


Figure 2: Details of (a) an anisotropic adapted mesh and (b) successive refinement steps of the background mesh. The blue line in (a) indicates the zero iso-contour of the level set function.

202 information. The final step consists of sampling all Bézier curves and in exporting a closed loop,  
 203 to be either used as an immersed mesh in a two-dimensional (2-D) environment, or extruded in  
 204 the off-body direction to serve as an immersed mesh in a three-dimensional (3-D) environment.

205 -. the immersed volume method (IVM) is used to immerse and represent all geometries inside  
 206 a unique mesh. The approach combines level-set functions, using the zero-iso value of a signed  
 207 distance function to localize the solid/fluid interface, and anisotropic mesh adaptation, to align  
 208 the mesh element edges with the interface and refine the mesh interface under the constraint of a  
 209 fixed, number of edges. This ensures that the quality of all actions taken over the course of a DRL  
 210 optimization is equally assessed, even though the interface is action-dependent.

211 Substantial evidence of the flexibility, accuracy and reliability of the numerical framework  
 212 for the intended application is documented in several papers to which the reader is referred for  
 213 exhaustive details regarding the shape generation using Bézier curves [46, 66], the level-set and mesh  
 214 adaptation algorithms [67, 68], the VMS formulations, stabilization parameters and discretization  
 215 schemes used in laminar and turbulent regimes [69–72], and the mathematical formulation of the  
 216 IVM in the context of finite element VMS methods [73, 74].

## 217 2.5. Numerical implementation

218 At each episode, actions drawn from the current policy are distributed to  $n_{env}$  environments  
 219 running in parallel, each of which executes a self-contained MPI-parallel numerical simulation  
 220 (here, all simulations are performed on a few tens of cores on a workstation of Intel Xeon E5-2640  
 221 processors) and feeds the reward associated to its input action to the DRL algorithm. There are thus  
 222 two levels of parallelism related to the environment and the computing architecture. This simple  
 223 parallelization technique is key to use DRL in the context of CFD applications, as a large number  
 224 of actions drawn from the current policy must be evaluated to accurately compute the expected  
 225 value of the policy loss (1). Even though, the high CPU cost of performing massive, unsteady  
 226 numerical simulations involving hundreds of thousands (even millions) of degrees of freedom caps  
 227 the number of environments that can efficiently run in parallel, and thus the number of state-  
 228 action-reward triplets that can be sampled from the current policy (which also makes intractable  
 229 the common practice in DRL studies to gain insight into the performances of the selected algorithm  
 230 by averaging results over multiple independent training runs with different random seeds, as it  
 231 would trigger a prohibitively large computational burden. The same random seeds are thus used  
 232 for all computations to ensure a minimal level of performance comparison between cases.) PBO  
 233 therefore improves the reliability of the loss evaluation by incorporating the reward data available  
 234 from several previous episodes, using an empirical decay parameter that exponentially decreases  
 235 the advantage history (to give recent episodes more weight) while retaining a longer memory of the  
 236 previous episodes as the problem dimensionality increases (in accordance with the idea that more  
 237 state-action-triplets are then needed to build a coherent covariance matrix). The remainder of the  
 238 practical implementation details are as follows:

Mean	Variance	Correlation	Neural network
$5 \times 10^{-3}$	$\gg$	$10^{-3}$	Learning rate
128	8	$\gg$	Nb. epochs
1	8	16	Nb. learning episodes
1	4	8	Nb. mini-batches
[2,2,2]	$\gg$	$\gg$	Architecture

Table 1: Details of the PBO meta-parameters and network architectures. For the architectures, only the sizes of the hidden layers are provided.

239 - the environment consists of CFD simulations of incompressible flows described in a Cartesian  
240 coordinate system with drag (resp. lift) positive in the  $+x$  (resp.  $+y$ ) direction. All equations are  
241 discretized on 2-D and 3-D rectangular grids whose side lengths documented in the coming sections  
242 have been checked to be large enough not to have a discernible influence on the results (with the  
243 exception of the 3-D case in section 5, for which we favor computing all numerical solutions at  
244 affordable CPU cost using a limited transverse dimension). Open flow conditions are used, that  
245 consist of a uniform inflow in the  $x$  direction, together with symmetric lateral, advective outflow  
246 and no-slip interface conditions. In turbulent regime, the ambient value of the Spalart–Allmaras  
247 variable is three times the molecular viscosity, as recommended to lead to immediate transition.  
248 Typical adapted meshes of the interface and wake regions are shown in figure 2, the latter also  
249 being accurately captured via successive refinement of the background elements.

250 - optimal surface shapes subject to a target cross-sectional area  $S_{ref}$  are determined by maximizing  
251 a compound reward function

$$r = \bar{J} - \beta |S - S_{ref}|, \quad (2)$$

252 where  $J$  is the objective function associated to performance,  $S$  is the cross-sectional area (also  
253 abbreviated as CSA in the following) of the shape, the overline indicates time-averaging, and  $\beta$  is  
254 a weighting coefficient that increasingly penalizes the shape when its area strays away from the  
255 target value. In practice, the cross-sectional area is computed as

$$S = \frac{1}{L} \int_{\Omega} H_{\epsilon}(\phi) d\Omega \quad (3)$$

256 where  $H_{\epsilon}$  is the smoothed Heaviside function introduced in [75], and  $L$  is the extrusion length in  
257 the off-body direction (hence equal to unity in 2-D). Moving average rewards and actions are also  
258 computed as the sliding average over the 50 latest values (or the whole sample if it has insufficient  
259 size). Time averages are performed over an interval  $[t_i; t_f]$  with edges large enough to dismiss the  
260 initial transient and achieve convergence to statistical equilibrium. In the following, we take  $J$  to  
261 be a function of the drag and lift coefficients per unit span in the transverse direction, denoted  
262 by  $D$  and  $L$ , respectively, whose instantaneous values are computed with a variational approach  
263 featuring only volume integral terms, reportedly less sensitive to the approximation of the body  
264 interface than their surface counterparts [76, 77].

265 - the agent consists of three identical fully connected networks with 3 hidden layers, each of  
266 which holds 2 neurons (this is by design, as we recall that the PBO networks can theoretically use  
267 different architectures). The only difference lies in the activation function applied to the output  
268 layer, namely the first network uses hyperbolic tangent to output the mean of the  $d$ -dimensional  
269 multivariate normal distribution in  $[-1; 1]^d$ , the second network uses sigmoid to output the standard  
270 deviations in  $[0; 1]^d$ , and the third network also uses sigmoid to output a set of coefficients in  $[0, 1]^d$ ,  
271 eventually assembled into a full correlation matrix by hypersphere decomposition [56, 57]. As to the  
272 meta-parameters, the number of parallel environments used to collect rewards before performing  
273 the network updates is set from the well-established heuristics of CMA-ES (that similarly relies on



274 full co-variance matrices and uses an evolution path to add information about correlations across  
 275 consecutive generations [78] to

$$n_{env} = \lfloor 4 + 3 \ln d \rfloor, \quad (4)$$

276 where  $\lfloor \cdot \rfloor$  denotes the floor function. Each network is updated for  $n_e$  epochs (the number of full  
 277 passes of the algorithm over the entire data set) using a learning rate  $\lambda$  (the size of the step taken in  
 278 the gradient direction for policy update) and a history of  $n_{ep}$  episodes, shuffled and organized in  $n_b$   
 279 mini-batches (whose sizes are in multiples of  $n_{env}$ ). The values used in this study are documented  
 280 in table 1 to ease reproducibility.

### 281 3. Validation

#### 282 3.1. Test case description

283 We assess first the relevance of the proposed numerical framework by revisiting the classical  
 284 problem of finding the 2-D shape minimizing the drag force induced by a surrounding uniform flow  
 285 at zero incidence. A sketch of the configuration is provided in figure 3. The origin of the coordinate  
 286 system is at the half chord length. Several laminar cases at Reynolds number  $Re = U_\infty \sqrt{S_{ref}} / \nu$   
 287 are modeled after the Navier–Stokes equations, where  $U_\infty$  is the inflow velocity,  $\nu$  the kinematic  
 288 viscosity, and we have used the square root of the target cross-sectional area (set to  $S_{ref} = 1$  in our  
 289 implementation) as reference length. The objective function is simply

$$J = -D, \quad (5)$$

290 and the weighing coefficient is set empirically to  $\beta = 8$ . All CFD environments use the simulation  
 291 parameters documented in table 2, found to offer a good compromise between numerical accuracy  
 292 and computational effort since numerical tests carried out at two other grid resolutions and spatial  
 293 extents yield limited variations within 2% – 3%.

294 The control points used to parametrize the shape are labeled clockwise from 0 at the leading  
 295 edge. All inner (i.e., non-end) curvature radii are set to 0.4 to provide sufficient smoothness (as  
 296 this is a tad below the value 0.5 required for maximal smoothness). This leaves  $n_p + 1$  independent  
 297 design variables, the chord length  $c$ , two end curvature radii  $\rho_{j \in \{0, n_p - 1\}}$  and  $n_p - 2$  inner thicknesses  
 298  $e_{k \in \{1, \dots, n_p - 2\}}$ . The network action output consists accordingly of values  $(\hat{c}, \hat{\rho}_j, \hat{e}_k)$  in  $[-1; 1]^{n_p + 1}$ ,  
 299 mapped into the actual physical quantities using

$$c = \frac{1 - \hat{c}}{2} c_{min} + \frac{1 + \hat{c}}{2} c_{max}, \quad \rho_j = \frac{1 - \hat{\rho}_j}{2} \rho_{min} + \frac{1 + \hat{\rho}_j}{2} \rho_{max}, \quad e_k = e_{k, max} - \frac{1 - \hat{e}_k}{2} \delta e, \quad (6)$$

300 for the chord to vary in  $[c_{min}; c_{max}]$  with  $c_{min} = 1$  and  $c_{max} = 4$ , the curvature radii to vary in  
 301  $[\rho_{min}; \rho_{max}]$  with  $\rho_{min} = 0.1$  and  $\rho_{max} = 0.4$ , and the thickness to vary in  $[e_{k, max} - \delta e; e_{k, max}]$   
 302 with  $\delta e = 0.4$  and  $e_{k, max}$  a maximum value tuned locally for each problem. At each episode, the  
 303 position of the inner points is adjusted to the current chord length to maintain equal spacing.  
 304 Unless specified otherwise, all results documented hereafter are for  $n_p = 5$ , for which DRL evolves  
 305 six design parameters, the chord length, two end curvature radii and three inner thicknesses.

#### 306 3.2. Results

307 Several Reynolds numbers have been considered up to  $Re = 100$ , for which random shapes  
 308 collected over the course of the optimization, are presented in figures 4-7, together with their  
 309 respective iso-contours of vorticity. Because the aspect ratio (as defined from the ratio of the  
 310 maximum thickness to the chord length) barely exceeds unity, all solutions at  $Re \lesssim 50$  relax to  
 311 steady state regardless of the DRL action (hence we do not report a proper averaging span for  
 312 these cases in table 2, as we simply evaluate reward at a final time chosen large enough to flush out  
 313 the transient behavior). Meanwhile, a small number of shapes with aspect ratio close to unity have  
 314 been found to exhibit vortex shedding at  $Re = 100$ , for which we pay attention to performing the  
 315 necessary time averages. Figures 4-7 also provide exhaustive convergence history for the reward,  
 316 the objective function, the ratio of actual to target CSA and the design parameters. The moving  
 317 average reward especially decreases almost monotonically and reaches a plateau after a few ten

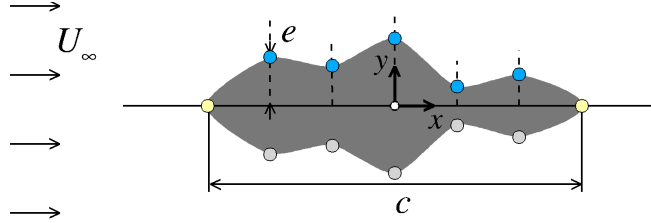


Figure 3: Schematic diagram of the minimum drag test case. The DRL agent optimizes the chord length, the curvature radius at the edge control points marked in yellow, and the thickness at the inner control points marked in blue. The thickness at the inner control points marked in grey deduces by symmetry.

318 episodes. At this point, the optimal CSA exhibits near-perfect agreement with the target value,  
 319 hence evidencing the relevance of the reward penalty approach.

320 At  $Re = 1$ , the minimum drag body in figure 4 is that of a perfectly front-rear symmetric  
 321 rugby ball, with chord length  $1.95 \pm 0.6\%$  and aspect ratio  $0.369 \pm 1.1\%$ . These values have been  
 322 obtained by averaging over the 10 latest episodes (with associated variance interval computed  
 323 from the root-mean-square over the same interval, a simple yet robust criterion that will be used  
 324 systematically to assess convergence for all cases reported in the following). They are close to  
 325 the creeping flow optimal, whose chord length (relative to a unit target surface) and aspect ratio  
 326 derived analytically in [80] are 1.88 and 0.40, respectively. The only noticeable difference lies in  
 327 the fact that the DRL optimal has a pointed rear end with wedge angle about  $90^\circ$ , and a slightly  
 328 more rounded front end with wedge angle  $\sim 120^\circ$ , while the creeping flow optimal has two pointed  
 329 ends with wedge angle about  $100^\circ$ . As the Reynolds number increases, the optimal chord length  
 330 increases but the thickness decreases, hence the aspect ratio of the optimal body decreases (likely  
 331 because the increasing adverse pressure gradient at the front needs to be counterbalance to avoid  
 332 flow separation). At  $Re = 20$ , the optimal shape in figure 5 has chord length  $2.40 \pm 0.8\%$  and  
 333 aspect ratio  $0.228 \pm 1.5\%$ , and remains almost front-rear symmetric, although the rear section is  
 334 slightly more streamlined. Similar results are obtained at  $Re = 50$  (figure 6), with chord length  
 335  $2.65 \pm 0.7\%$  and aspect ratio  $0.204 \pm 1.0\%$ . At  $Re = 100$ , the front-rear symmetry is lost as we obtain  
 336 a streamlined shape with chord length  $2.46 \pm 0.4\%$  and aspect ratio  $0.235 \pm 0.8\%$ ; see figure 7. At  
 337  $Re = 1$ , the optimal drag ( $13.10 \pm 0.02\%$ ) cuts down that of the equivalent cylinder (i.e., the cylinder  
 338 of diameter  $2\sqrt{S_{ref}/\pi}$ , for the area to be equal to  $S_{ref}$ ) by 6%, which is small but simply reflects  
 339 that the ratio of drags on any two bodies tends to 1 in the limit where the Reynolds number tends  
 340 to 0. In comparison, the achieved reduction is by 22% at  $Re = 20$  (optimal drag  $1.83 \pm 0.04\%$ ), 24%  
 341 at  $Re = 50$  ( $1.10 \pm 0.05\%$ ), and 48% at  $Re = 100$  ( $0.71 \pm 4\%$ ).

342 Other than that, it is difficult to accurately validate the results because, although the search  
 343 for optimal profiles of minimum drag in Navier–Stokes flows having received substantial interest in  
 344 the literature, there is a wide variability in the problem formulation, especially in terms of design  
 345 constraints (some authors specify a target surface, others impose only a lower bound, plus the  
 346 values can vary from one reference to another), and the exact geometrical properties of the optimal  
 347 (e.g., length, aspect ratio) are rarely documented. The closest study to our work is from Kondoh  
 348 *et al.* [79], who tackle similar drag minimization problems via topology optimization, using a body  
 349 force to model the effect of classical no-slip boundary conditions at the fluid/solid interface. It has  
 350 not been possible to assess the convergence rate of DRL in the absence of any reference information  
 351 in this regard, and it is not entirely clear whether the exact same optimization problem is solved  
 352 due to inconsistent statements in the study regarding the nature of the design constraint, but  
 353 even so, the reported optimal shapes and drags turn to be in good agreement with the present  
 354 DRL results. One minor difference is that the shapes look pointed in [79] (but the blending of  
 355 the interface makes it difficult to see in the original images), while the present ones are generally  
 356 rounded at both ends, with little to no effect on the reward. On this point, we note that the  
 357 end radii can vary substantially even after the reward has converged (as is the case for instance  
 358 in figure 6(d) at  $Re = 50$ ), which evidences a general lack of sensitivity to these specific design  
 359 parameters). At  $Re = 1$ , the optimal drags differ by approximately 7%, which may seem large at

					Case setup
	1	20	50	100	Reynolds number
	5	»	»	»	Nb. points
	6	»	»	»	Nb. design variables
					CFD
	2	»	»	»	Dimensionality
	0.2	»	0.125	»	Time-step
	[50;50]	»	»	»	Averaging time span
	8	»	»	»	Penalty coeff.
	[-10; 20]×[-10; 10]	»	»	»	Mesh dimensions
	100000	»	110000	»	Nb. mesh elements
	0.0005	»	»	»	Interface ⊥ mesh size
					PBO
	100	120	115	»	Nb. episodes
	10	12	11	»	Nb. environments
	3mn	3mn	5mn	10mn	CPU time <sup>†‡</sup>
	5h	6h	10h	18h	Resolution time <sup>‡</sup>
					Parameter ranges
	[1;4]	»	»	»	Chord length
	[0.1;0.4]	»	»	»	LE curv. radius
	[0.072;0.472]	»	»	»	↓ Thickness
	[0.152;0.552]	»	»	»	
	[0.072;0.472]	»	»	»	
	[0.1;0.4]	»	»	»	TE curv. radius
					Optimal
	1.95	2.41	2.64	2.46	Chord length
	0.309	0.300	0.186	0.344	LE curv. radius
	0.297	0.246	0.223	0.290	↓ Thickness
	0.362	0.273	0.270	0.267	
	0.299	0.227	0.199	0.166	
	0.115	0.366	0.258	0.395	TE curv. radius
	1.000	1.000	1.000	1.001	Ratio of actual to target CSA
	13.1	1.83	1.10	0.71	Drag (present)
	12.10	1.81	1.10	0.76	Drag [79]

Table 2: Case setup, simulation parameters and convergence data for the drag minimization problem, as computed by averaging over the 10 latest learning episodes. Leading-edge (front end) and trailing edge (rear end) data are labeled LE and TE, respectively. † All CPU times provided per episode and per environment. ‡ All values obtained averaging over 5 independent runs using 12 cores.

360 first sight but is actually fair given the high sensitivity of drag to small changes in the Reynolds  
361 number in this regime. The drags and chord lengths are nearly identical at  $Re = 20$  and  $50$ , as we  
362 find the ratio of the chord length at the current Reynolds number to its  $Re = 1$  counterpart to be  
363 1.24 at  $Re = 20$  and 1.35 at  $Re = 50$  using DRL, while extracting data from the reference figures  
364 using a graph digitizer software yields values of 1.26 at  $Re = 20$  and 1.33 at  $Re = 50$  (it has not been  
365 possible to similarly extract the aspect ratio due to blurred and/or mixed pixels). At  $Re = 100$ ,  
366 the shapes somewhat differ as the optimal in [79] is more elongated and less streamlined in the  
367 rear section. Meanwhile the optimal drags differ by only 6%, which raises the possibility that the  
368 objective function has either a unique flat minimum, or several nearly equivalent minima. Figure 7  
369 constitutes a favorable presumption in this regard, as the objective function exhibits surprisingly  
370 low variations over the course of optimization, and most shapes in figure 7(b) actually are within

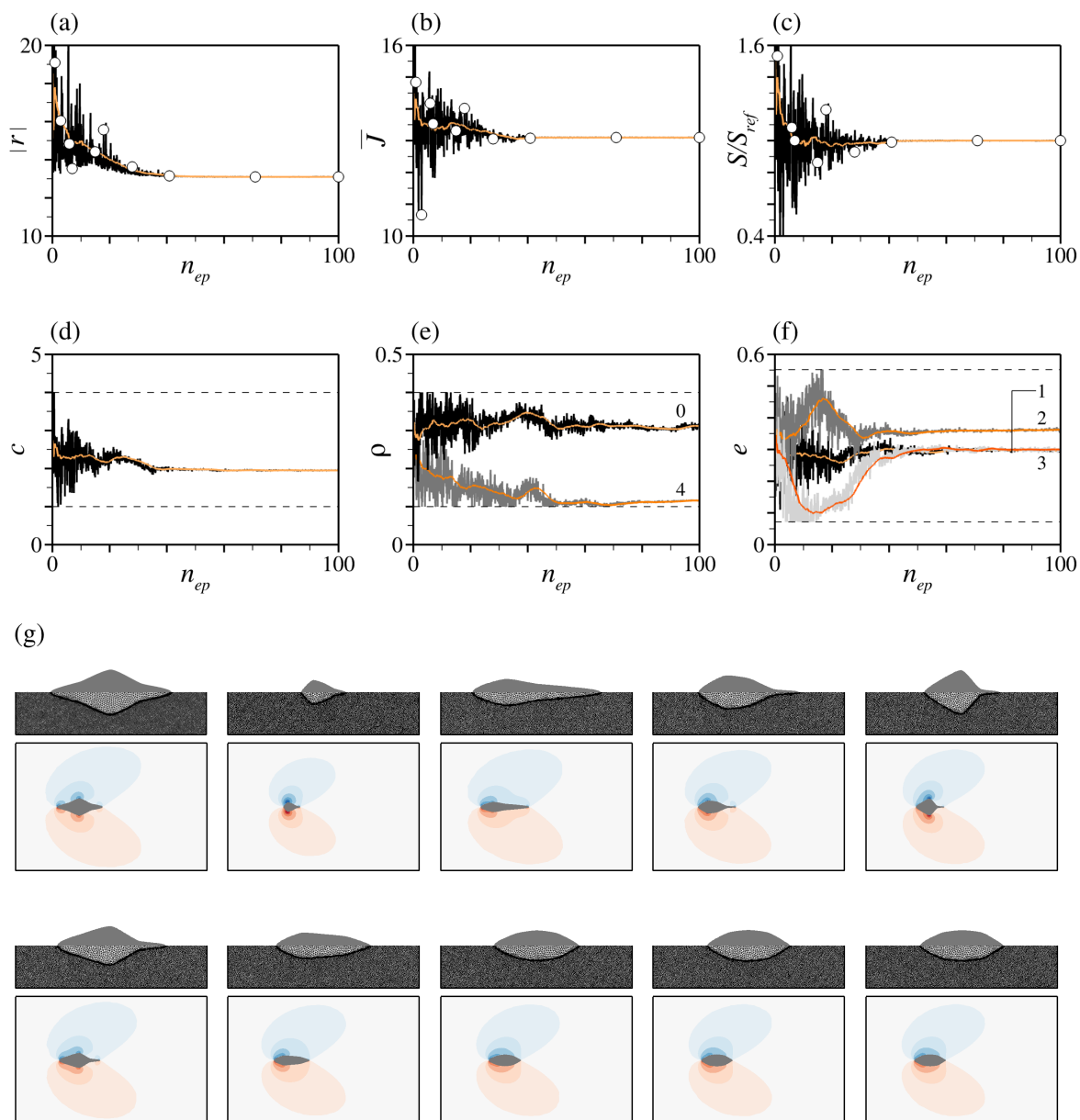


Figure 4: Maximum lift to drag ratio test case at  $Re = 1$  under constant area constraint  $S_{ref} = 1$ . (a) Evolution per episode of the instant (black line) and moving average (over episodes, light orange line) reward (in absolute value). (b-f) Same as (a) for the (b) averaged (over time) drag, (c) ratio of the actual to target cross-sectional areas, (d) chord, (e) edge curvature radii and (f) inner thicknesses. All labels in (e-f) are ordered clockwise from the leading edge. The horizontal dashed lines in (d-f) mark the admissible values. (g) Shapes generated over the course of optimization for random episodes marked by the circle symbols in (a-c), together with corresponding iso-contours of vorticity. The last three shapes pertain to episodes 40, 70 and 100, respectively.

371 the 6% variance interval marked by the grey shade.

### 372 3.3. Discussion

373 We believe the above results assess the relevance of the proposed DRL-CFD framework for  
 374 optimal shape design. Relying on low-dimensional parametrization of the body shape is one key  
 375 parameter in this regard, as it improves the tractability of the optimization process and avoids the  
 376 oscillations between points that have been found to occur when using a larger (about 10) number  
 377 of control points. Nonetheless, we believe important to discuss the impact on robustness, and the  
 378 extent to which decreasing the number of control points exaggerates (or not) the sensitivity to the  
 379 curvature radii. This is because using different curvature radii to connect the same set of control

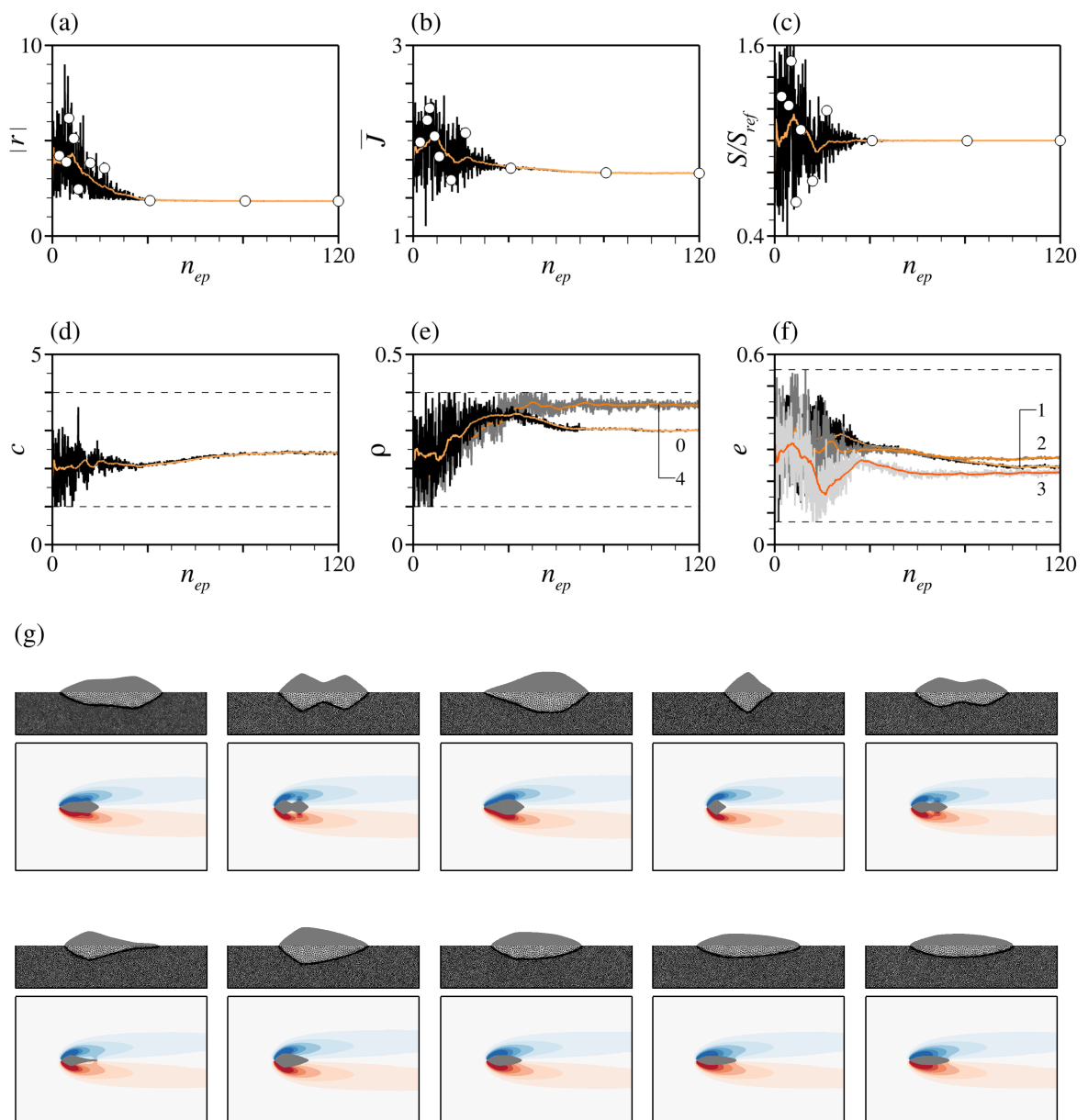


Figure 5: Maximum lift to drag ratio test case at  $Re = 20$  under constant area constraint  $S_{ref} = 1$ . (a) Evolution per episode of the instant (black line) and moving average (over episodes, light orange line) reward (in absolute value). (b-f) Same as (a) for the (b) averaged (over time) drag, (c) ratio of the actual to target cross-sectional areas, (d) chord, (e) edge curvature radii and (f) inner thicknesses. All labels in (e-f) are ordered clockwise from the leading edge. The horizontal dashed lines in (d-f) mark the admissible values. (g) Shapes generated over the course of optimization for random episodes marked by the circle symbols in (a-c), together with corresponding iso-contours of vorticity. The last three shapes pertain to episodes 40, 80 and 120, respectively.

380 points can yield two slightly different cross-sectional areas, that in turn can earn two substantially  
 381 different reward via the penalization term (this is not on the Bézier parametrization itself, though,  
 382 only on the need to smoothly connect a discrete set of control points. For instance, one must also  
 383 specify tangency at both endpoints of a spline).

384 As a first insight into this issue, we report here results obtained at  $Re = 1$  using three alternative  
 385 parametrizations:

- 386 • a case with  $n_p = 7$  control points evolving the chord length, five inner thicknesses and two  
 387 end curvature radii (which amounts to replicating the above reference case, but with two  
 388 additional inner thicknesses, hence 8 independent design parameters),

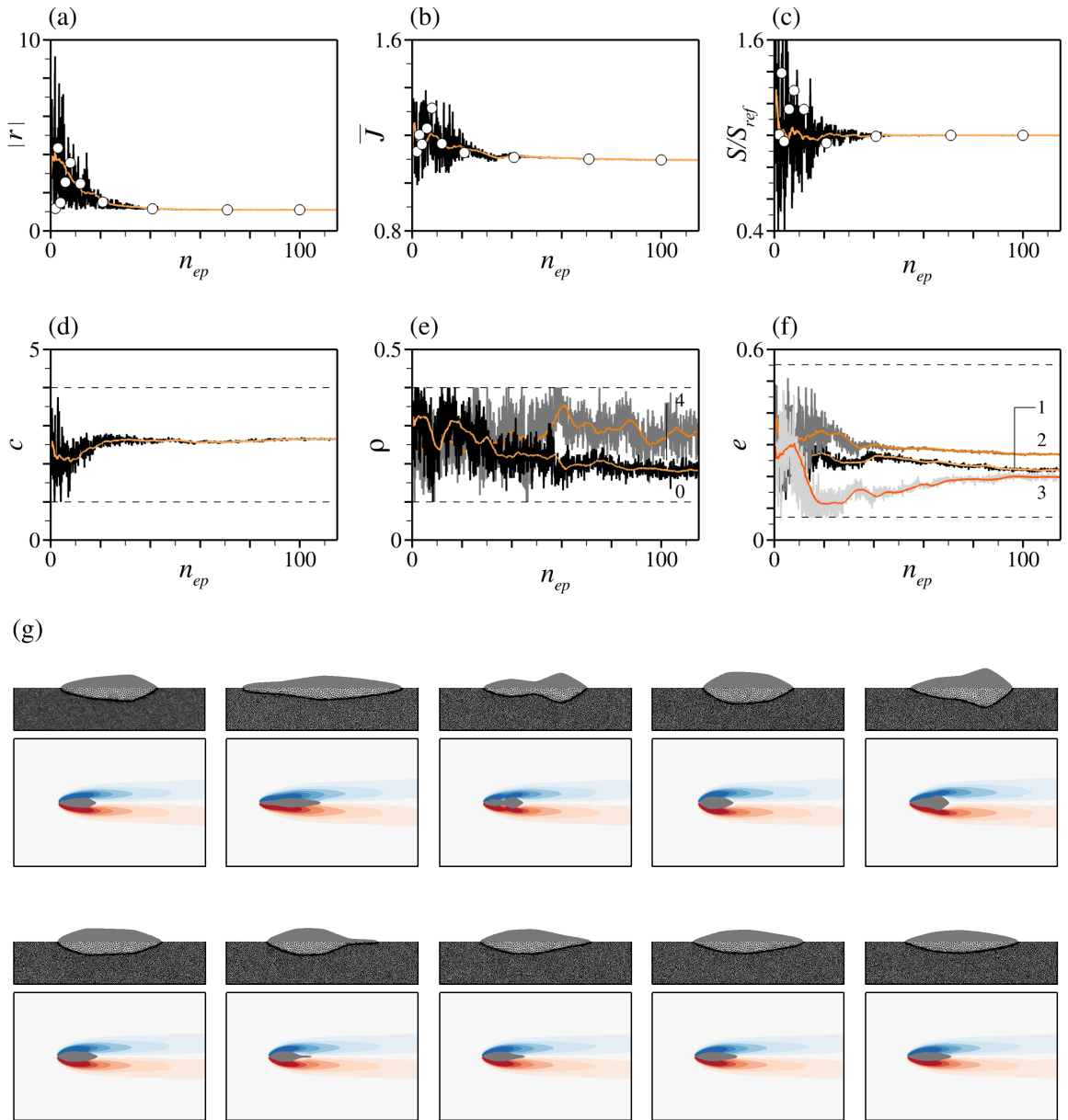


Figure 6: Maximum lift to drag ratio test case at  $Re = 50$  under constant area constraint  $S_{ref} = 1$ . (a) Evolution per episode of the instant (black line) and moving average (over episodes, light orange line) reward (in absolute value). (b-f) Same as (a) for the (b) averaged (over time) drag, (c) ratio of the actual to target cross-sectional areas, (d) chord, (e) edge curvature radii and (f) inner thicknesses. All labels in (e-f) are ordered clockwise from the leading edge. The horizontal dashed lines in (d-f) mark the admissible values. (g) Shapes generated over the course of optimization for random episodes marked by the circle symbols in (a-c), together with corresponding iso-contours of vorticity. The last three shapes pertain to episodes 40, 70 and 120, respectively.

- 389 • a case with  $n_p = 5$  points evolving the chord length, three inner thicknesses, two end cur-  
390 vature radii, plus an additional radius common to all inner control points (which amounts  
391 to replicating the reference case, but with one additional inner curvature radius, hence 7  
392 independent design parameters),
- 393 • a case with  $n_p = 7$  points whose thickness distribution is frozen, as obtained interpolating  
394 from the reference  $n_p = 5$  optimal (for which it suffices to sample the connecting Bezier  
395 curves at the relevant positions), after which a dedicated DRL agent restores the proper  
396 cross-sectional area by evolving two end curvature radii, plus an additional radius common

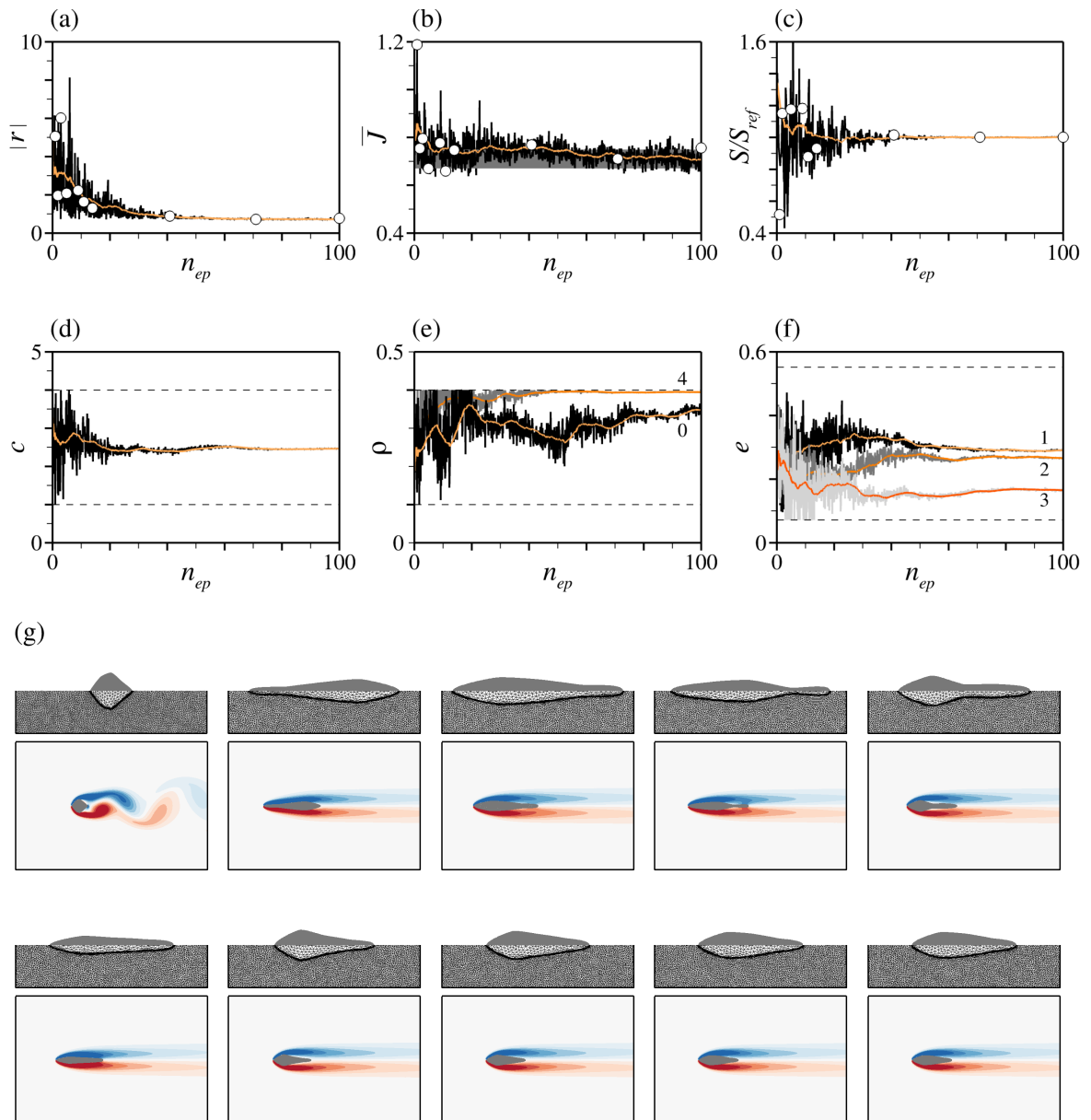


Figure 7: Maximum lift to drag ratio test case at  $Re = 100$  under constant area constraint  $S_{ref} = 1$ . (a) Evolution per episode of the instant (black line) and moving average (over episodes, light orange line) reward (in absolute value). (b-f) Same as (a) for the (b) averaged (over time) drag, (c) ratio of the actual to target cross-sectional areas, (d) chord, (e) edge curvature radii and (f) inner thicknesses. The grey shade in (b) marks the 6% variance interval with respect to the average over the 10 latest learning episodes. All labels in (e-f) are ordered clockwise from the leading edge. The horizontal dashed lines in (d-f) mark the admissible values. (g) Shapes generated over the course of optimization for random episodes marked by the circle symbols in (a-c), together with corresponding iso-contours of vorticity. The last three shapes pertain to episodes 40, 70 and 120, respectively.

397 to all inner control points (hence, 3 independent design parameters) with reward

$$r = -|S - S_{ref}|, \quad (7)$$

398 formally identical to (2) with  $J = 0$  and  $\beta = 1$ .

399 The results reported in table 3 exhibit limited discrepancy with respect to the reference (reproduced  
400 from table 2 in the first column), as the maximum deviation on the chord and the inner thicknesses  
401 is by 4%. All runs converge to similar curvature radii at the front. The value at the rear is  
402 noticeably different, but with little to no effect on the reward, objective function and shape (as  
403 shown in figure 8), which simply reflects the smallness of the reward gradients with respect to the

					Case setup
5	5	7	7	Nb. points	
×	✓	×	✓	Inner curv. radius	
6	7	8	3	Dimensionality	
					Optimal
1.95	1.96	1.87	1.95	Chord length	
0.4	0.332	0.4	0.398	Inner curv. radius	
0.309	0.359	0.310	0.394	LE curv. radius	
0.297	0.284	0.233	0.206	Thickness	
0.362	0.367	0.340	0.324		
0.299	0.303	0.369	0.359		
-	-	0.341	0.331		
-	-	0.258	0.227		
0.115	0.159	0.392	0.389	TE curv. radius	
1.00	1.00	1.00	1.00	Ratio of actual to target CSA	
13.10	13.09	13.09	13.09	Drag	

Table 3: Sensitivity of the drag minimization problem at  $Re = 1$  to the discretization parameters. Leading-edge (front end) and trailing edge (rear end) data are labeled LE and TE, respectively. The first column is reproduced from table 2.



Figure 8: (a) Reference optimal shape for the minimum drag test case at  $Re = 1$  with  $n_p = 5$  control points and fixed inner curvature radius. (b) Same as (a) with  $n_p = 7$  control points and fixed inner curvature radius. (c) Same as (a) with  $n_p = 5$  control points and variable inner curvature radius. (d) Reference optimal shape discretized with  $n_p = 7$  control points, after DRL has adjusted the end and inner curvature radii to restore the proper cross-sectional area.

control variables in the vicinity of the optimal. Although the impact needs to be assessed on a case  
to case basis, this suggests that the method ability to provide robust optima may not be strained  
by the use of low-end geometrical parametrizations.

#### 4. Application to optimal aerodynamic design

##### 4.1. Test case description

We apply now the method to more meaningful aerodynamic shape optimization problems, as we  
seek the shape maximizing the lift to drag ratio (used as an indicator of the aerodynamic efficiency)  
induced by a surrounding uniform flow at angle of attack of  $\alpha = 30^\circ$ . A sketch of the configuration  
is provided in figure 9. A Cartesian coordinate system is used with origin at quarter chord length  
from the leading edge. The target cross-sectional area is set to  $S_{ref} = 0.0822$ , which corresponds to  
the CSA of a NACA (National Advisory Committee for Aeronautics) 0012 airfoil. The objective  
function is

$$J = \frac{L}{D}, \quad (8)$$

and the weighing coefficient is set to  $\beta = 100$ . Two time-dependent flow regimes are modeled  
after either the Navier–Stokes or the uRANS equations, for which all CFD environments use the  
numerical simulation parameters provided in table 4.

As has been done for the drag minimization test case, we simplify the parametrization by  
setting all inner curvature radii to 0.4. Additionally, we fix the chord length to  $c = 1$  for the  
chord Reynolds number  $Re = U_\infty c / \nu$  to remain constant over the course of optimization (which  
we believe is necessary to meaningfully compare the performances). This leaves  $n_p$  independent



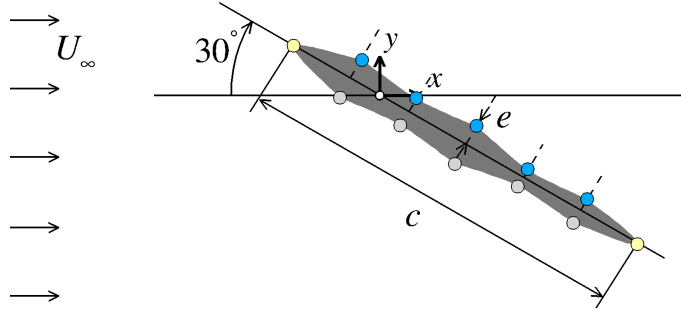


Figure 9: Schematic diagram of the maximum lift to drag ratio test case.

423 design variables, two end curvature radii  $\rho_{j \in \{0, n_p - 1\}}$  and  $n_p - 2$  inner thicknesses  $e_{k \in \{1, \dots, n_p - 2\}}$ .  
 424 The network action output consists accordingly of values  $(\hat{\rho}_j, \hat{e}_k)$  in  $[-1; 1]^{n_p}$ , converted into the  
 425 actual physical quantities using the same mapping (6), only we set  $\delta e = 0.03$  to account for the  
 426 smaller target CSA. All results in the following are for  $n_p = 5$ , for which DRL evolves five design  
 427 parameters, two end curvature radii and three inner thicknesses.

#### 4.2. Laminar regime at $Re = 250$

429 We consider first a 2-D laminar case at  $Re = 250$  modeled after the Navier–Stokes equations,  
 430 for which the dimensions of the computational domain provided in table 4 yield a blockage ratio of  
 431 2.5%. All mesh adaptations are performed under the constraint of a fixed total number of elements  
 432  $n_{el} = 100000$ . A total of 100 episodes has been run for this case, that yield the variety of shapes  
 433 illustrated in figure 10, together with their respective iso-contours of (instantaneous) vorticity. The  
 434 general picture is that all shapes exhibit an oscillating pattern of leading- and trailing-edge vortex  
 435 shedding following the shedding of the initial leading-edge vortex. This stems from the interaction  
 436 between the (lower) negative vorticity sheet, that separates at the leading edge and then rolls up  
 437 into a large clockwise vortex, and the (upper) positive vorticity sheet, that remains attached to  
 438 the windward side and rolls up counter-clockwise from the trailing edge (in average, this yields a  
 439 massive separation originating at the leading edge and extending on the leeward side, all the way  
 440 to the trailing edge; not shown here). The Strouhal number for vortex shedding frequency built  
 441 from the windward width is  $S_t = fc \sin \alpha / u_\infty \sim 0.13$  (regardless of the shape), which is identical to  
 442 experimental measurements performed on a high-aspect ratio NACA 0012 airfoil under the same  
 443 incidence at  $Re = 100$  [81].

444 The moving reward in figure 10(a) increases almost monotonically and reaches a plateau after  
 445 about 40 episodes. The optimal lift to drag ratio computed as the average over the 10 latest  
 446 episodes is  $1.24 \pm 1.0\%$ , at which point the cross-sectional area is equal to its target value down to  
 447 the fifth decimal place. We note that 40 episodes is actually the number of episodes needed for the  
 448 end radii to converge, as the thickness distribution exhibits excellent convergence after as little as  
 449 20 episodes. Interestingly, the agent has generated a wing-like optimal shape representative of a  
 450 high-lift configuration without any priori knowledge of aerodynamic concepts: the optimal features  
 451 a rounded leading edge to help maintain a smooth airflow (with curvature radius  $0.394 \pm 0.01\%$   
 452 close to maximum) and a sharp trailing edge to generate lift (with curvature radius  $0.156 \pm 0.02$   
 453 close to minimum). The optimal lift to drag ratio exceeds that of the equivalent ellipse (i.e., of  
 454 major diameter  $c$  and minor diameter  $2S_{ref}/\pi c$ , for the area to be equal to  $S_{ref}$ ) by 6% and that  
 455 of a NACA 0012 airfoil by 1%, as has been estimated from dedicated in-house calculations. This  
 456 is small but consistent with the overall lack of sensitivity, as the objective function in figure 10(c)  
 457 actually remains within 3% of the optimal over the course of optimization, as indicated by the grey  
 458 shade delimiting the related variance interval.

#### 4.3. Turbulent (transitional) regime at $Re = 5000$

460 We consider now a case at  $Re = 5000$  corresponding to the ultra-low Reynolds number regime,  
 461 that has assumed greater significance in the last few decades due to relevance for micro air vehicles  
 462 and micro-turbines [82, 83]. We believe this constitutes a valuable first step towards applying

				Case setup
250	5000	>		Reynolds number
5	>	>		Nb. points
5	>	3		Nb. design variables
				CFD
2	>	3		Dimensionality
-	RANS	>		Turb. model
0.125	0.05	>		Time-step
[100;150]	[150;200]	[100;150]		Averaging time span
100	>	90		Penalty coeff.
$[-10; 20] \times [-10; 10]$	$[-6; 15] \times [-7; 7]$	$[-5; 10] \times [-5; 5] \times [0; 5]$		Mesh dimensions
100000	120000	500000		Nb. mesh elements
0.0005	>	0.001		Interface $\perp$ mesh size
				PBO
100	100	80		Nb. episodes
14	14	12		Nb. environments
20mn	2h45mn	9h30mn		CPU time <sup>††</sup>
35h	275h	760h		Resolution time <sup>‡</sup>
				Parameter ranges
-	-	-		Chord length
[0.1;0.4]	>	-		LE curv. radius
[0.024;0.084]	>	>		↓ Thickness
[0.03;0.09]	>	>		
[0.024;0.084]	>	>		↓
[0.1;0.4]	>	-		
				Optimal
1	1	1		Chord length
0.394	0.398	0.3		LE curv. radius
0.0638	0.0549	0.0420		↓ Thickness
0.0514	0.0627	0.0536		
0.0253	0.0252	0.0454		↓
0.156	0.104	0.1		
1.00	1.00	0.996		Ratio of actual to target CSA
1.24	1.54	1.34		Lift to drag ratio

Table 4: Case setup, simulation parameters and convergence data for the lift to drag ratio maximization problem, as computed by averaging over the 10 latest learning episodes. † All CPU times provided per episode and per environment. ‡ All values obtained averaging over 5 independent runs using 12 cores.

463 the method to more prototypal aerodynamic applications in which airfoils operate at Reynolds  
464 numbers of  $\sim 10^6$  and exhibit some degree of stochastic dynamics (as they carry turbulent energy  
465 distributed over a wide range of scales with varying degrees of spatial and temporal coherence),  
466 which might lead to high variance gradient estimates and hamper learning. Here, at the high value  
467 of angle of attack considered, the flow is expected to be transitional, for instance, transition in  
468 the wake of a NACA 0012 has been shown to occur in the separated shear-layer, shortly after  
469 the leading edge, at a location strongly dependent on the level of external noise [84]. This has  
470 been confirmed vetting preliminary Navier–Stokes simulations for which the built-in small-scale  
471 component of the VMS solution acts as an implicit large eddy simulation. While the solutions  
472 (not reported here for the sake of conciseness) are dominated by the large-scale component, with  
473 small-scale turbulence noticeably absent downstream, intermittent small-scale fluctuations develop  
474 on the leeward side, that prompt asymmetric vortex street (at least is the trailing edge is not too

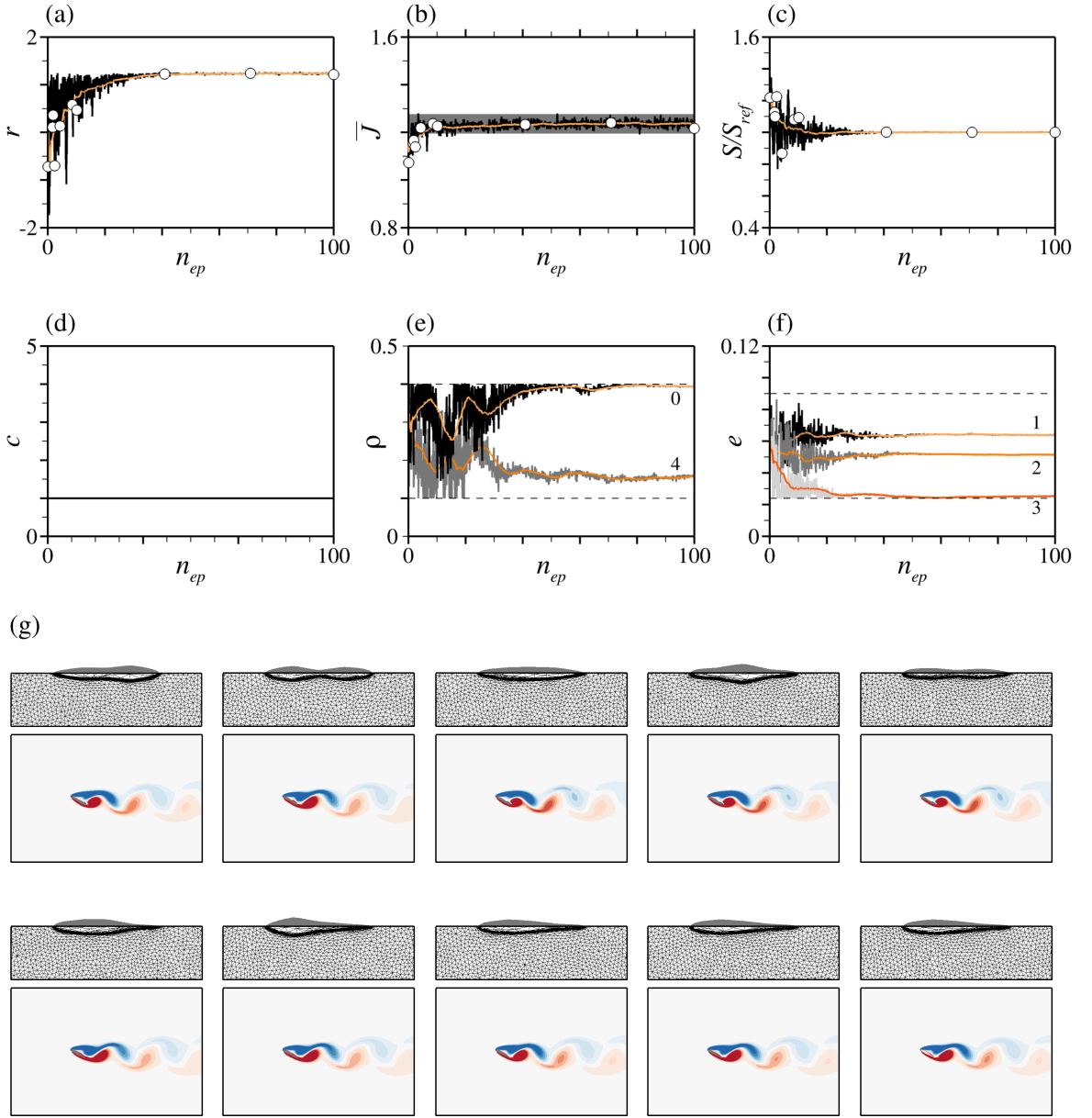


Figure 10: Maximum lift to drag ratio test case at  $Re = 250$  under constant area constraint  $S_{ref} = 0.0822$ . (a) Evolution per episode of the instant (black line) and moving average (over episodes, light orange line) reward. (b-f) Same as (a) for the (b) averaged (over time) lift to drag ratio, (c) ratio of the actual to target cross-sectional areas, (d) chord (fixed), (e) edge curvature radii and (f) inner thicknesses. The grey shade in (b) marks the 3% variance interval with respect to the average over the 10 latest learning episodes. All labels in (e-f) are ordered clockwise from the leading edge. The horizontal dashed lines in (e-f) mark the admissible values. (g) Shapes generated over the course of optimization for random episodes marked by the circle symbols in (a-c), together with corresponding iso-contours of vorticity. The last three shapes pertain respectively to episodes 40, 70 and 100.

475 sharp for the separation point to be free to move) with vortices convected downstream along an  
 476 axis inclined upward with respect to the streamwise direction, similar to the behavior observed in  
 477 2-D LES simulations of the transitional flow past a circular cylinder [85].

478 Accordingly, the case is modeled here after the uRANS equations, using negative Spalart-  
 479 Allmaras as turbulence model. Such an approach is not without shortcomings (namely RANS is  
 480 inherently designed to damp out the small-scales, and Spalart-Allmaras assumes fully turbulent  
 481 behavior), but given the cost of accurately resolving the complex, unsteady vortex interaction  
 482 described above, we believe the deficiencies are more than offset by the tremendous gain in compu-  
 483 tational efficiency derived from the relatively coarse meshes necessary to predict the most important

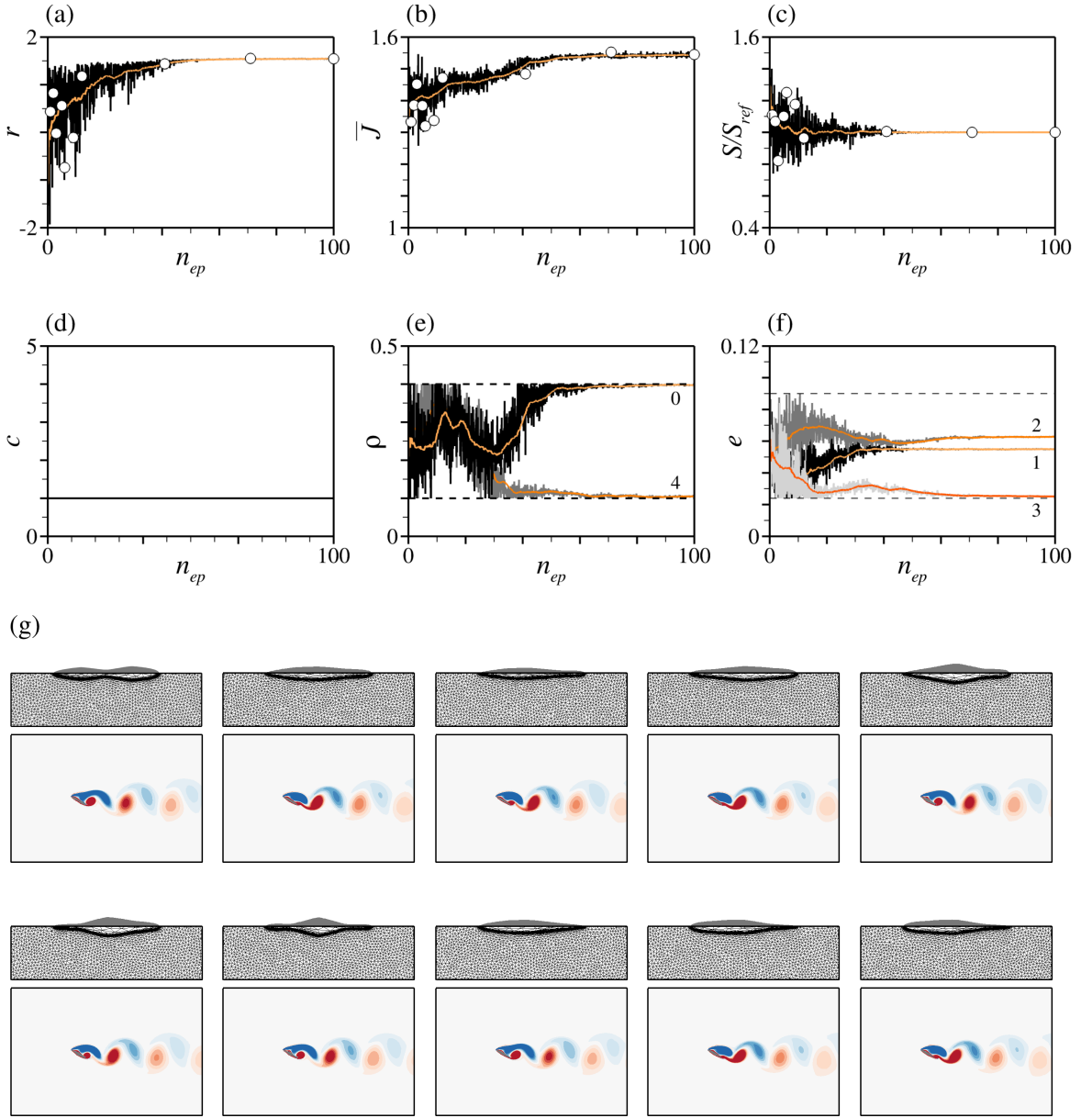


Figure 11: Maximum lift to drag ratio test case at  $Re = 5000$  with negative Spalart–Allmaras turbulence model, under constant area constraint  $S_{ref} = 0.0822$ . (a) Evolution per episode of the instant (black line) and moving average (over episodes, light orange line) reward. (b-f) Same as (a) for the (b) averaged (over time) lift to drag ratio, (c) ratio of the actual to target cross-sectional areas, (d) chord (fixed), (e) edge curvature radii and (f) inner thicknesses. All labels in (e-f) are ordered clockwise from the leading edge. The horizontal dashed lines in (d-f) mark the admissible values. (g) Shapes generated over the course of optimization for random episodes marked by the circle symbols in (a-c), together with corresponding iso-contours of vorticity. The last three shapes pertain respectively to episodes 40, 70 and 100.

484 large scale features of the flow. In practice, a scaled-down computational domain is used, whose  
 485 dimensions reported in table 4 yield a blockage ratio of 3.5%. All mesh adaptations are performed  
 486 under the constraint of a fixed total number of elements  $n_{el} = 120000$ . A total of 100 episodes has  
 487 been run, for which the selected iso-contours of vorticity documented in figure 11 are reminiscent of  
 488 their laminar counterparts, with in-line vortex shedding (since the effect of the intermittent small-  
 489 scale fluctuations has been lumped into the eddy viscosity model) and robust shedding frequency  
 490  $S_t = 0.15$ .

491 The moving average reward in figure 11(a) is seen to converges within about 50 episodes but  
 492 the thickness distribution again converges faster (within roughly 40 episodes). As was already the

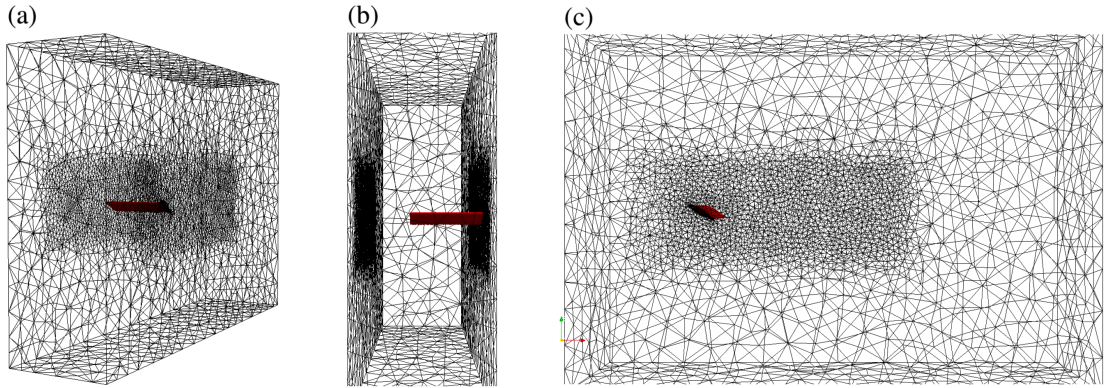


Figure 12: Anisotropic adapted mesh around an immersed three-dimensional unswept, rectangular wing. (a) Three-dimensional view. (b) Front view. (c) Side view.

493 case at  $Re = 250$ , the optimal resembles the airfoil of an airplane wing, with a rounded leading  
 494 edge and a sharp trailing edge. The end radii are nearly identical to their laminar counterparts,  
 495 but the shape is streamlined differently, namely it is a tad thinner in the front ( $0.0549 \pm 0.2\%$   
 496 at  $Re = 5000$  vs.  $0.0638 \pm 0.2\%$  at  $Re = 250$ ) but slightly thicker in the center ( $0.0627 \pm 0.4\%$   
 497 at  $Re = 5000$  vs.  $0.0514 \pm 0.25\%$  at  $Re = 250$ ). The optimal lift to drag ratio ( $1.54 \pm 0.3\%$ ) exceeds  
 498 that of the equivalent ellipse by 13% but is ultimately identical to that of a NACA 0012, despite  
 499 the objective function exhibiting substantial variations in figure 11(b). The inability to outperform  
 500 a conventional airfoil should not be interpreted as failure of the method, though, as aerodynamic  
 501 shape design classically requires fine-tuning of the local geometry for a gain that often adds up  
 502 to a few percent. This is not manageable here because the low number of degrees of freedom  
 503 inevitably constrains the underlying space of shapes, and the expected gain is comparable to the  
 504 typical convergence threshold of a DRL run. We believe the results should rather be considered  
 505 proof that DRL can start from the ground up and generate shapes that perform just as well as  
 506 a conventional airfoil. Actually, there is ample room for improvement if the optimization is to be  
 507 tailored to airfoil shape optimization problems (which it is not here for the sake of generality),  
 508 one may seek for instance to locally refine the DRL optimal by repeating the same analysis, but  
 509 clustering the control points in specific regions of interest (e.g., the leading-edge, or the rear-end  
 510 of the leeward side), or to rely on alternative parametrizations better suited to airfoils, such as  
 511 CST [86].

## 512 5. Extension to 3-D shape optimization.

513 The ultra low-Reynolds number case at  $Re = 5000$  is extended here to 3-D to assess the extent  
 514 to which the approach carries over to three-dimensional shape optimization. All shapes generated  
 515 over the course of optimization are unswept, rectangular wings, whose cross-section is set up from  
 516 the DRL outputs following the exact same process as in sections 3 and 4. The span aspect ratio  
 517 (relative to the chord length) is set to 3 in our implementation. A Cartesian coordinate system is  
 518 used with origin in the mid-span plane, at quarter chord length from the leading edge. The number  
 519 of control points remains set to  $n_p = 5$ , but we force the leading and trailing edge curvature radii  
 520 to 0.3 (round edge) and 0.1 (sharp edge) to keep the computational cost manageable, which leaves  
 521  $n_p - 2 = 3$  independent design variables corresponding to the inner thicknesses. In practice, only a  
 522 half-span wing body is simulated with symmetry boundary condition prescribed at the mid-span.  
 523 The computational domain shown in figure 12 is a rectangular prism, whose dimensions reported  
 524 in table 4 yield a blockage ratio of 5%. All mesh adaptations are performed under the constraint  
 525 of a fixed total number of elements  $n_{el} = 500000$ . This is likely insufficient to claim true numerical  
 526 accuracy, but given the numerical cost (960 3-D simulations total, each of which is performed on  
 527 12 cores and lasts about 10h, hence 9600h of total CPU cost), we believe this is a reasonable  
 528 compromise to assess feasibility while producing qualitative results to build on.

529 A total of 80 episodes has been run for this case, using a slightly lower weighing coefficient  $\beta = 90$   
 530 (to take into account that the coarser mesh yields a small loss in accuracy in the computation of

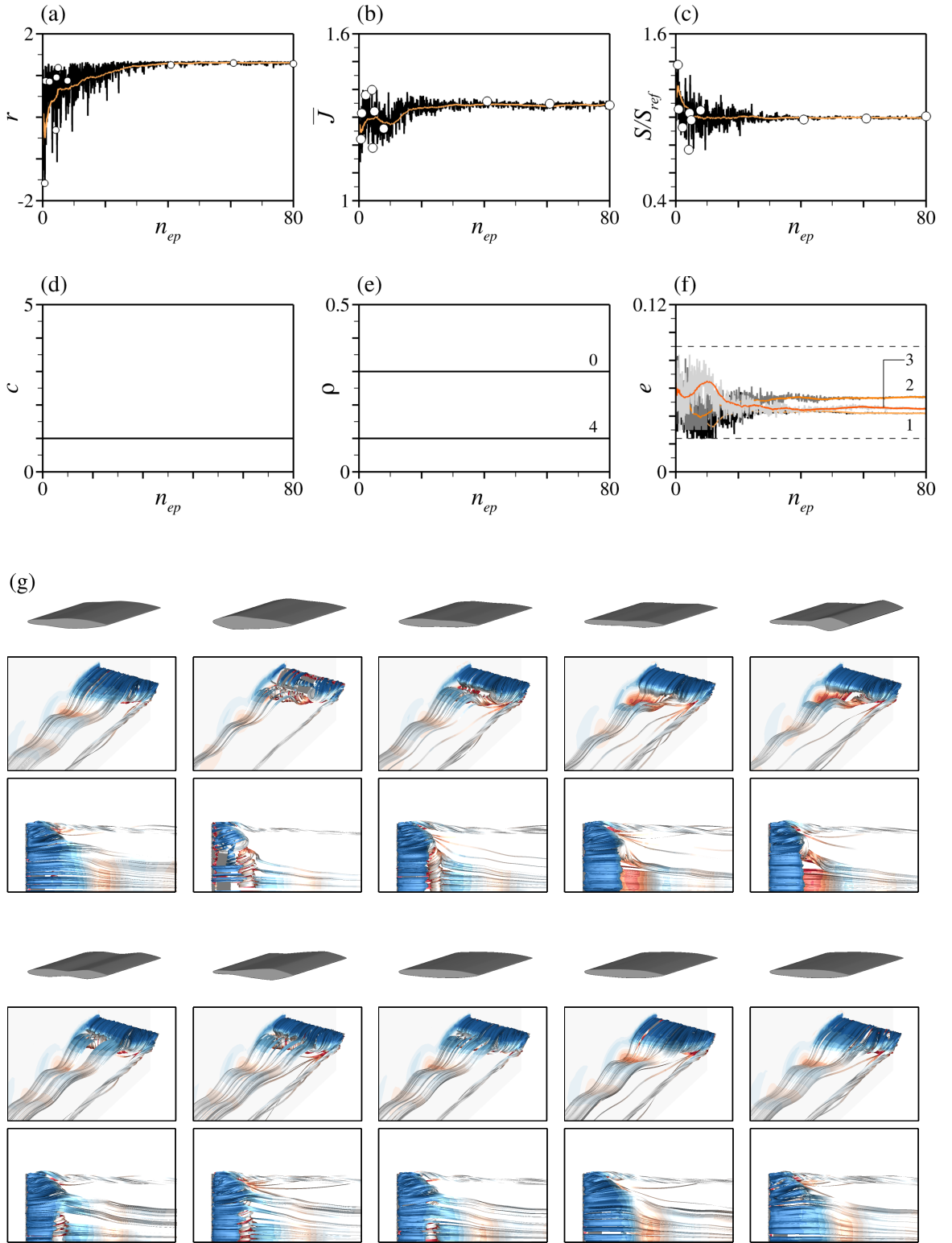


Figure 13: Maximum lift to drag ratio test case in 3-D at  $Re = 5000$  with negative Spalart–Allmaras turbulence model, under constant area constraint  $S_{ref} = 0.0822$ . (a) Evolution per episode of the instant (black line) and moving average (over episodes, light orange line) reward. (b-f) Same as (a) for the (b) averaged (over time) lift to drag ratio, (c) ratio of the actual to target cross-sectional areas, (d) chord (fixed over the course of optimization), (e) edge curvature radii (also fixed) and (f) inner thicknesses. (g) Shapes generated over the course of optimization for random episodes marked by the circle symbols in (a-c), together with corresponding iso-contours of vorticity. The last three shapes pertain respectively to episodes 40, 70 and 100.

531 the cross-sectional area). Several representative flow patterns computed over the course of opti-  
532 mization are illustrated in figure 13 to display the increased degree of complexity due to transverse  
533 inhomogeneities. All solutions exhibit vortex shedding, which is because the span aspect ratio is  
534 large enough for the tip vortex to remain relatively steady. Conversely, preliminary simulations  
535 carried out at lower aspect ratios of order 1 systematically relaxed to steady-state, due to the  
536 strong tip-vortex induced downwash over the entire span (the same behavior has been reported in  
537 laminar flows at Reynolds numbers of about in the range of a few hundreds [87], and is ascribed  
538 here to the RANS damping of the small-scale transverse motion, that should otherwise strengthen  
539 the unsteadiness). The moving averager reward in figure 13 plateaus after about 35 episodes. The  
540 3-D distribution is almost front-rear symmetric but the shape itself surprisingly slightly thinner  
541 in the front than in the rear, although the rear is ultimately more streamlined due to the smaller  
542 trailing edge curvature radius. Compared to its 2-D counterpart, the 3-D optimal is thinner in the  
543 front and in the center, but much thicker in the rear. The optimal lift to drag ratio ( $1.34 \pm 0.5\%$ )  
544 exceeds that of the equivalent ellipse by 5% and is identical to that of a NACA 0012. This is  
545 consistent with the above findings, in the sense that the DRL optimal performs at the level of a  
546 conventional airfoil, and that the limited improvement with respect to the equivalent ellipse should  
547 not be taken as an indictment of the method, just a consequence of the flow regime considered  
548 (precisely because a similar improvement is achieved using a NACA 0012).

## 549 6. Conclusion

550 Shape optimization in computational fluid dynamics systems is achieved here training fully  
551 connected networks with PBO, a recently introduced deep reinforcement algorithm at the crossroad  
552 of policy gradient methods and evolution strategies. PBO is single-step, meaning that the DRL  
553 agent gets only one attempt per learning episode at finding the optimal. The numerical reward  
554 fed to the PBO agent is computed with a finite elements CFD environment solving stabilized  
555 weak forms of the governing equations (Navier–Stokes, otherwise uRANS with negative Spalart–  
556 Allmaras as turbulence model) with a combination of variational multiscale approach, immersed  
557 volume method and anisotropic mesh adaptation.

558 Several cases are documented, for which shapes with fixed camber line, angle of attack and cross-  
559 sectional area are generated by varying a chord length and a symmetric thickness distribution (and  
560 possibly extruding in the off-body direction), connecting consecutive points by a cubic Bézier curve  
561 using local position and curvature information. The classical problem of finding the 2-D shape of  
562 minimum drag in a uniform flow is revisited first to validate and assess the method capabilities.  
563 The method is also applied to the more practically meaningful problem of finding the shape of  
564 maximum lift to drag ratio (in 2-D or 3-D) at an incidence of  $30^\circ$  and under constant chord  
565 Reynolds number. The DRL optimal increases the performance the equivalent ellipse (i.e., the  
566 ellipse of same cross-sectional area) by 13% in 2-D and 5% in 3-D. It is systematically found to  
567 perform just as well as a conventional airfoil, despite DRL starting from the ground up and having  
568 no priori knowledge of aerodynamic concepts. Exhaustive convergence and efficiency data are  
569 reported here with the hope to foster future comparisons, but it is worth emphasizing that we did  
570 not seek to optimize said efficiency, neither by optimizing the PBO meta-parameters, nor by using  
571 pre-trained deep learning models (as is done in transfer learning).

572 Fluid dynamicists have just begun to gauge the relevance of DRL and its application to opti-  
573 mal shape design. This research weighs in on this issue and shows that the proposed single-step  
574 method holds a high potential as a reliable, go-to black-box optimizer for complex CFD problems.  
575 Moreover, the optimization process is entirely domain-agnostic, meaning that the proposed frame-  
576 work allows for easy application to any domain in which shape optimization may be beneficial. We  
577 believe further work should now focus on the challenges specific to fluid mechanics that still pre-  
578 vent DRL capabilities from meeting the requirements for practical deployment, e.g., stochasticity,  
579 sampling efficiency (CFD environments are resource expensive as they routinely involve numerical  
580 simulations with tens or hundreds of millions of degrees of freedom, while classical RL methods  
581 have low sample efficiency, i.e., many trials are required for the agent to learn a purposive behav-  
582 ior), the need to leverage experience from multiple agents learning concurrently (multi-agent DRL)  
583 or to train an agent in reasoning about several weighted objectives (multi-objective reward).

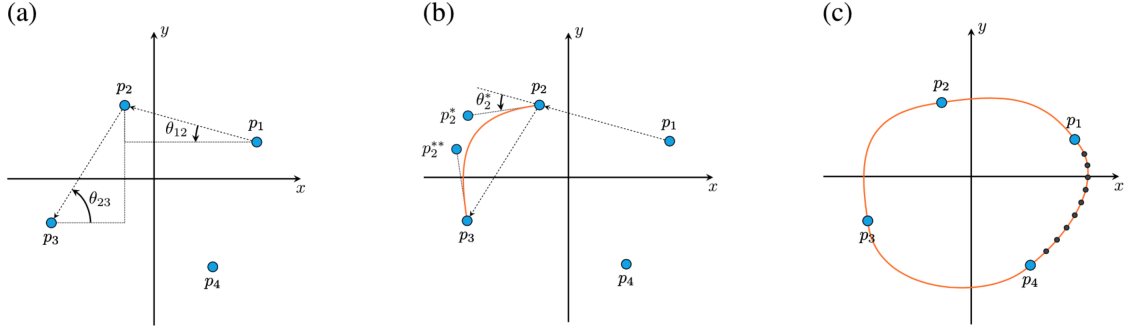


Figure 14: Shape generation using cubic Bézier curves. Each subfigure illustrates one of the consecutive steps used in the process. (a) Compute angles between points and compute an average angle  $\theta_i^*$  around each point. (b) Compute supplemental control points coordinates from averaged angles and generate cubic Bézier curve. (c) Sample all Bézier lines and export for mesh immersion.

## 584 Acknowledgements

585 This work is supported by the Carnot M.I.N.E.S. Institute through the M.I.N.D.S. project  
586 (CFL2022).

## 587 Data availability

588 The data that support the findings of this study are available from the corresponding author  
589 upon reasonable request.

590

## 591 Appendix A. Shape generation using Bézier curves

592 This section describes the process followed to generate shapes from a set of  $n_p$  control points.  
593 Once the position has been reconstructed from the agent outputs, the angles between consecutive  
594 points are computed. An average angle is then computed around each point (see Fig. 14(a)) as

$$\theta_i^* = r\theta_{i-1,i} + (1-r)\theta_{i,i+1}, \quad (\text{A.1})$$

595 where  $r \in [0; 1]$  is the curvature radius that control the local sharpness of the curve. Then, each  
596 pair of points is joined using a cubic Bézier curve, defined by four points: the first and last points,  
597  $p_i$  and  $p_{i+1}$  belong to the curve, while the second and third ones,  $p_i^*$  and  $p_i^{**}$ , are supplemental  
598 control points that define the tangent of the curve at  $p_i$  and  $p_{i+1}$ . The tangents at  $p_i$  and  $p_{i+1}$   
599 are respectively controlled by  $\theta_i$  and  $\theta_{i+1}$  (Fig. 14(b)). A final sampling of the successive Bézier curves  
600 leads to a boundary description of the shape (Fig. 14(c)). Using this method, a wide variety of  
601 shapes can be attained.

## 602 References

- 603 [1] P. Gangl, U. Langer, A. Laurain, H. Meftahi, K. Sturm, Shape optimization of an electric  
604 motor subject to nonlinear magnetostatics, *SIAM J. Sci. Comput.* 37 (2015) B1002–B1025.
- 605 [2] R. Udawalpola, M. Berggren, Optimization of an acoustic horn with respect to efficiency and  
606 directivity, *Int. J. Numer. Methods Eng.* 73 (2008) 1571–1606.
- 607 [3] M. Hintermüller, W. Ring, A second order shape optimization approach for image segmenta-  
608 tion, *SIAM J. Appl. Math.* 64 (2004) 442–467.
- 609 [4] J. Pinzon, M. Siebenborn, A. Vogel, Parallel 3d shape optimization for cellular composites on  
610 large distributed-memory clusters, *Adv. Model. Simul. Eng. Sci.* 7 (2020) 117–135.



- 611 [5] P.-I. Schneider, X. G. Santiago, V. Soltwisch, M. Hammerschmidt, S. Burger, C. Rockstuhl,  
612 Benchmarking five global optimization approaches for nano-optical shape optimization and  
613 parameter reconstruction, arXiv preprint arXiv:1809.06674 (2019).
- 614 [6] O. Pironneau, On optimum profiles in stokes flow, *J. Fluid Mech.* 59 (1973) 117–128.
- 615 [7] O. Pironneau, On optimum design in fluid mechanics, *J. Fluid Mech.* 64 (1974) 97–110.
- 616 [8] J. J. Corbett, H. W. Koehler, Updated emissions from ocean shipping, *J. Geophys. Res.* 108  
617 (2003) 4650–64.
- 618 [9] A. L. Marsden, M. Wang, B. Mohammadi, P. Moin, Shape optimization for aerodynamic noise  
619 control, Center for Turbulence Research Annual Brief (2001).
- 620 [10] I. Rodriguez-Eguia, I. Errasti, U. Fernandez-Gamiz, J. M. Blanco, E. Zulueta, A. Saenz-  
621 Aguirre, A parametric study of trailing edge flap implementation on three different airfoils  
622 through an artificial neuronal network, *Symmetry* 12 (2020) 828.
- 623 [11] M. C. G. Hall, Application of adjoint sensitivity theory to an atmospheric general circulation  
624 model, *J. Atmos. Sci.* 43 (1986) 2644–2651.
- 625 [12] A. Jameson, L. Martinelli, N. A. Pierce, Optimum aerodynamic design using the Navier–Stokes  
626 equations, *Theor. Comput. Fluid Dyn.* 10 (1998) 213–237.
- 627 [13] M. D. Gunzburger, *Perspectives in flow control and optimization*, SIAM, Philadelphia, 2002.
- 628 [14] S. N. Skinner, H. Zare-Behtash, State-of-the-art in aerodynamic shape optimisation methods,  
629 *Appl. Soft Comput.* 62 (2018) 933–962.
- 630 [15] J. H. Holland, Genetic algorithms, *Sci. Am.* 267 (1992) 66–73.
- 631 [16] J. Kennedy, R. Eberhart, Particle swarm optimization, in: *Procs. of ICNN’95-international  
632 conference on neural networks, 1995*, pp. 1942–1948.
- 633 [17] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, Equation of state  
634 calculations by fast computing machines, *J. Chem. Phys* 21 (1953) 1087–1092.
- 635 [18] R. Hassan, B. Cohanin, O. De Weck, G. Venter, A comparison of particle swarm optimization  
636 and the genetic algorithm, *AIAA 2005-1897* (2005).
- 637 [19] Z. Han, C. Xu, L. Zhang, Y. Zhang, K. Zhang, S. Wenping, Efficient aerodynamic shape opti-  
638 mization using variable-fidelity surrogate models and multilevel computational grids, *Chinese  
639 J. Aeronaut.* 33 (2020) 31–47.
- 640 [20] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, P. K. Tucker, Surrogate-  
641 based analysis and optimization, *Prog. Aerosp. Sci.* 41 (2005) 1–28.
- 642 [21] O. Chernukhin, D. W. Zingg, Multimodality and global optimization in aerodynamic design,  
643 *AIAA J.* 51 (2013) 1342–1354.
- 644 [22] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert,  
645 L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche,  
646 T. Graepel, D. Hassabis, Mastering the game of go without human knowledge, *Nature* 550  
647 (2017) 354–359.
- 648 [23] M. Moravčik, M. Schmid, N. Burch, V. Lisy, D. Morrill, N. Bard, T. Davis, K. Waugh,  
649 M. Johanson, M. Bowling, DeepStack: expert-level artificial intelligence in heads-up no-limit  
650 poker, *Science* 356 (2017) 508–513.
- 651 [24] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal Policy Optimization  
652 Algorithms, arXiv preprint arXiv:1707.06347 (2017).

- 653 [25] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, M. Hutter, Learning  
654 agile and dynamic motor skills for legged robots, *Sci. Robot.* 4 (2019) eaau5872.
- 655 [26] A. Bernstein, E. Burnaev, Reinforcement learning in computer vision, in: *Procs. of the 10th*  
656 *International Conference on Machine Vision*, 2018.
- 657 [27] Y. Deng, F. Bao, Y. Kong, Z. Ren, Q. Dai, Deep direct reinforcement learning for financial  
658 signal representation and trading, *IEEE Trans. Neural Netw. Learn. Syst.* 28 (2017) 653–664.
- 659 [28] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley,  
660 A. Shah, Learning to drive in a day, arXiv preprint arXiv:1807.00412 (2018).
- 661 [29] A. Bewley, J. Rigley, Y. Liu, J. Hawke, R. Shen, V.-D. Lam, A. Kendall, Learning to drive  
662 from simulation without real world labels, arXiv preprint arXiv:1812.03823 (2018).
- 663 [30] W. Knight, Google just gave control over data center cool-  
664 ing to an AI, [http://www.technologyreview.com/s/611902/  
665 google-just-gave-control-over-data-center-cooling-to-an-ai/](http://www.technologyreview.com/s/611902/google-just-gave-control-over-data-center-cooling-to-an-ai/) (2018).
- 666 [31] J. Viquerat, P. Meliga, E. Hachem, A review on deep reinforcement learning for fluid mechan-  
667 ics: an update, arXiv preprint arXiv:2107.12206 (2021).
- 668 [32] J. Rabault, M. Kuchta, A. Jensen, U. Réglade, N. Cerardi, Artificial neural networks trained  
669 through deep reinforcement learning discover control strategies for active flow control, *J. Fluid*  
670 *Mech.* 865 (2019) 281–302.
- 671 [33] J. Rabault, A. Kuhnle, Accelerating deep reinforcement learning strategies of flow control  
672 through a multi-environment approach, *Phys. Fluids* 31 (2019) 094105.
- 673 [34] M. A. Elhawary, Deep reinforcement learning for active flow control around a circular cylinder  
674 using unsteady-mode plasma actuators, arXiv preprint arXiv:2012.10165 (2020).
- 675 [35] M. Holm, Using deep reinforcement learning for active flow control, Ph.D. thesis, Master  
676 Thesis University of Oslo (2020).
- 677 [36] J. Rabault, F. Ren, W. Zhang, H. Tang, H. Xu, Deep reinforcement learning in fluid mechanics:  
678 a promising method for both active flow control and shape optimization, *J. Hydrodynam.* 32  
679 (2020) 234–246.
- 680 [37] F. Ren, H. Hu, H. Tang, Active flow control using machine learning: A brief review, *J.*  
681 *Hydrodynam.* 32 (2020) 247–253.
- 682 [38] H. Tang, J. Rabault, A. Kuhnle, Y. Wang, T. Wang, Robust active flow control over a range  
683 of reynolds numbers using an artificial neural network trained through deep reinforcement  
684 learning, *Phys. Fluids* 32 (2020) 053605.
- 685 [39] M. Tokarev, E. Palkin, R. Mullyadzhanov, Deep reinforcement learning control of cylinder  
686 flow using rotary oscillations at low Reynolds number, *Energies* 13 (2020) 5920.
- 687 [40] H. Xu, W. Zhang, J. Deng, J. Rabault, Active flow control with rotating cylinders by an  
688 artificial neural network trained by deep reinforcement learning, *J. Hydrodynam.* 32 (2020)  
689 254–258.
- 690 [41] H. Ghraieb, J. Viquerat, A. Larcher, P. Meliga, E. Hachem, Single-step deep reinforcement  
691 learning for open-loop control of laminar and turbulent flows, *Phys. Rev. Fluids* 6 (2021, in  
692 press) 053902.
- 693 [42] R. Paris, R. Beneddine, J. Dandois, Robust flow control and optimal sensor placement using  
694 deep reinforcement learning, *J. Fluid Mech.* 913 (2021).
- 695 [43] S. Qin, S. Wang, G. Sun, An application of data driven reward of deep reinforcement learning  
696 by dynamic mode decomposition in active flow control, arXiv preprint arXiv:arXiv:2106.06176  
697 (2021).

- 698 [44] F. Ren, J. Rabault, H. Tang, Applying deep reinforcement learning to active flow control in  
699 weakly turbulent conditions, *Phys. Fluids* 33 (2021) 037121.
- 700 [45] V. Belus, J. Rabault, J. Viquerat, Z. Che, E. Hachem, U. Réglade, Exploiting locality  
701 and translational invariance to design effective deep reinforcement learning control of the  
702 1-dimensional unstable falling liquid film, *AIP Adv.* 9 (2019) 125014.
- 703 [46] J. Viquerat, J. Rabault, A. Kuhnle, H. Ghraieb, A. Larcher, E. Hachem, Direct shape opti-  
704 mization through deep reinforcement learning, *J. Comput. Phys.* 428 (2021) 110080.
- 705 [47] G. Novati, S. Verma, D. Alexeev, D. Rossinelli, W. M. van Rees, P. Koumoutsakos, Syn-  
706 chronisation through learning for two self-propelled swimmers, *Bioinspir. Biomim.* 12 (2017)  
707 036001.
- 708 [48] S. Verma, G. Novati, P. Koumoutsakos, Efficient collective swimming by harnessing vortices  
709 through deep reinforcement learning, *Proc. Natl. Acad. Sci. U.S.A.* 115 (2018) 5849–5854.
- 710 [49] D. Fan, L. Yang, Z. Wang, M. S. Triantafyllou, G. E. Karniadakis, Reinforcement learning for  
711 bluff body active flow control in experiments and simulations, *Proc. Natl. Acad. Sci. U.S.A.*  
712 117 (2020) 26091–26098.
- 713 [50] X. Yan, J. Zhu, M. Kuang, X. Wang, Aerodynamic shape optimization using a novel optimizer  
714 based on machine learning techniques, *Aerosp. Sci. Technol.* 86 (2019) 826–835.
- 715 [51] X. Hui, H. Wang, W. Li, J. Bai, F. Qin, G. He, Multi-object aerodynamic design optimization  
716 using deep reinforcement learning, *AIP Advances* 11 (2021) 085311.
- 717 [52] R. Li, Y. Zhang, H. Chen, Learning the aerodynamic design of supercritical airfoils through  
718 deep reinforcement learning, *AIAA J.* 59 (2021) 3988–4001.
- 719 [53] S. Qin, S. Wang, L. Wang, C. Wang, G. Sun, Y. Zhong, Multi-objective optimization of  
720 cascade blade profile based on reinforcement learning, *Appl. Sci.* 11 (2021) 106.
- 721 [54] E. Hachem, H. Ghraieb, J. Viquerat, A. Larcher, P. Meliga, Deep reinforcement learning for  
722 the control of conjugate heat transfer, *J. Comput. Phys.* 436 (2021) 110317.
- 723 [55] J. Viquerat, R. Duvigneau, P. Meliga, A. Kuhnle, E. Hachem, Policy-based optimiza-  
724 tion: single-step policy gradient method seen as an evolution strategy, *arXiv preprint*  
725 *arXiv:2104.06175* (2021).
- 726 [56] R. Rebonato, P. Jäckel, The most general methodology to create a valid correlation matrix  
727 for risk management and option pricing purposes, Available at SSRN 1969689 135 (2011).
- 728 [57] K. Numpacharoen, A. Atsawarungruangkit, Generating correlation matrices based on the  
729 boundaries of their coefficients, *PLoS One* 7 (2012) e48902.
- 730 [58] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley,  
731 A. Shah, Adam: a method for stochastic optimization, *arXiv preprint arXiv:1412.6980* (2014).
- 732 [59] T. Coupez, E. Hachem, Solution of high-reynolds incompressible flow with stabilized finite  
733 element and adaptive anisotropic meshing, *Comput. Methods Appl. Mech. Engrg.* 267 (2013)  
734 65–85.
- 735 [60] T. J. R. Hughes, G. R. Feijóo, L. Mazzei, J.-B. Quincy, The variational multiscale method -  
736 a paradigm for computational mechanics, *Comput. Methods Appl. Mech. Engrg.* 166 (1998)  
737 3–24.
- 738 [61] R. Codina, Stabilization of incompressibility and convection through orthogonal sub-scales in  
739 finite element methods, *Comput. Methods Appl. Mech. Engrg.* 190 (2000) 1579–1599.
- 740 [62] Y. Bazilevs, V. M. Calo, J. A. Cottrell, T. J. R. Hughes, A. Reali, G. Scovazzi, Variational  
741 multiscale residual-based turbulence modeling for large eddy simulation of incompressible  
742 flows, *Comput. Methods Appl. Mech. Engrg.* 197 (2007) 173–201.

- 743 [63] S. R. Allmaras, F. T. Johnson, P. R. Spalart, Modifications and clarifications for the im-  
744 plementation of the Spalart–Allmaras turbulence model, in: *Procs. of the 7th International*  
745 *Conference on Computational Fluid Dynamics*, 2012.
- 746 [64] R. Codina, Comparison of some finite element methods for solving the diffusion-convection-  
747 reaction equation, *Comput. Methods Appl. Mech. Engrg.* 156 (1998) 185–210.
- 748 [65] S. Badia, R. Codina, Analysis of a stabilized finite element approximation of the transient  
749 convection-diffusion equation using an ALE framework, *SIAM J. Numer. Anal.* 44 (2006)  
750 2159–2197.
- 751 [66] J. Viquerat, E. Hachem, A supervised neural network for drag prediction of arbitrary 2d  
752 shapes in laminar flows at low Reynolds number, *Comp. Fluids* 210 (2020) 104645.
- 753 [67] J. Bruchon, H. Dignonnet, T. Coupez, Using a signed distance function for the simulation of  
754 metal forming processes: formulation of the contact condition and mesh adaptation, *Int. J.*  
755 *Numer. Meth. Eng.* 78 (2004) 980–1008.
- 756 [68] C. Gruau, T. Coupez, 3D tetrahedral, unstructured and anisotropic mesh generation with  
757 adaptation to natural and multidomain metric, *Comput. Methods Appl. Mech. Engrg.* 194  
758 (2005) 4951–4976.
- 759 [69] E. Hachem, B. Rivaux, T. Kloczko, H. Dignonnet, T. Coupez, Stabilized finite element method  
760 for incompressible flows with high Reynolds number, *J. Comput. Phys.* 229 (23) (2010) 8643–  
761 8665.
- 762 [70] T. Coupez, G. Jannoun, N. Nassif, H. C. Nguyen, H. Dignonnet, E. Hachem, Adaptive time-step  
763 with anisotropic meshing for incompressible flows, *J. Comput. Phys.* 241 (2013) 195–211.
- 764 [71] J. Sari, F. Cremonesi, M. Khalloufi, F. Cauneau, P. Meliga, Y. Mesri, E. Hachem, Anisotropic  
765 adaptive stabilized finite element solver for RANS models, *Int. J. Numer. Meth. Fl.* 86 (2018)  
766 717–736.
- 767 [72] G. Guiza, A. Larcher, A. Goetz, L. Billon, P. Meliga, E. Hachem, Anisotropic boundary layer  
768 mesh generation for reliable 3D unsteady RANS simulations, *Finite Elem. Anal. Des.* 170  
769 (2020) 103345.
- 770 [73] E. Hachem, H. Dignonnet, E. Massoni, T. Coupez, Immersed volume method for solving natural  
771 convection, conduction and radiation of a hat-shaped disk inside a 3d enclosure, *Int. J. Numer.*  
772 *Method H.* 22 (2012) 718–741.
- 773 [74] E. Hachem, S. Feghali, R. Codina, T. Coupez, Immersed stress method for fluid-structure  
774 interaction using anisotropic mesh adaptation, *Int. J. Numer. Meth. Eng.* 94 (2013) 805–825.
- 775 [75] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incom-  
776 pressible two-phase flow, *J. Comput. Phys.* 114 (1994) 146–159.
- 777 [76] V. John, *Parallele Lösung der inkompressiblen Navier–Stokes Gleichungen auf adaptiv verfein-*  
778 *erten Gittern*, Ph.D. thesis, Otto-von-Guericke-Universität Magdeburg, Fakultät für Mathe-  
779 matik (1997).
- 780 [77] V. John, Reference values for drag and lift of a two-dimensional time-dependent flow around a  
781 cylinder, *Int. J. Numer. Meth. Fl.* 44 (2004) 777–788.
- 782 [78] N. Hansen, The CMA Evolution Strategy: a tutorial, arXiv preprint arXiv:1604.00772 (2016).
- 783 [79] T. Kondoh, T. Matsumori, A. Kawamoto, Drag minimization and lift maximization in laminar  
784 flows via topology optimization employing simple objective function expressions based on body  
785 force integration, *Structural and Multidisciplinary Optimization* 45 (2012) 693–701.
- 786 [80] S. Richardson, Optimum profiles in two-dimensional Stokes flow, *Proc. R. Soc. A* 450 (1995)  
787 603–622.

- 788 [81] J.-Y. Andro, G. Dergham, R. Godoy-Diana, L. Jacquin, D. Sipp, Conditions critiques de  
789 déclenchement du lâcher tourbillonnaire au cours du vol des insectes, in: Procs. of the 19ème  
790 Congrès Français de Mécanique, 2009.
- 791 [82] S. Sunada, T. Yasuda, K. Yasuda, K. Kawachi, Comparison of wing characteristics at an  
792 ultralow Reynolds number, *J. Aircr.* 39 (2002) 331–338.
- 793 [83] D. Funda Kurtulus, Vortex flow aerodynamics behind a symmetric airfoil at low angles of  
794 attack and reynolds numbers, *Int. J. Micro Air Veh.* 13 (2021) 17568293211055653.
- 795 [84] S. Wang, Y. Zhou, M. Mahbub Alam, H. Yang, Turbulent intensity and reynolds number  
796 effects on an airfoil at low reynolds numbers, *Phys. Fluids* 26 (2014) 115107.
- 797 [85] M. Breuer, Large eddy simulation of the subcritical flow past a circular cylinder: Numerical  
798 and modeling aspects, *Int. J. Numer. Meth. Fl.* 28 (1998) 1281–1302.
- 799 [86] B. Kulfan, J. Bussoletti, “Fundamental” parameteric geometry representations for aircraft  
800 component shapes, in: Procs. of the 11th AIAA/ISSMO multidisciplinary analysis and opti-  
801 mization conference, 2006, p. 6948.
- 802 [87] K. Zhang, S. Hayostek, M. Amitay, W. He, V. Theofilis, K. Taira, On the formation of three-  
803 dimensional separated flows over wings under tip effects, *J. Fluid Mech.* 895 (2020).