15TH IEA HEAT PUMP CONFERENCE
MAY 26–29, 2026 | HOFBURG VIENNA 2026
DECARBONISATION THROUGH INNOVATION

Physics-informed reinforcement learning enabling ergonomic aerodynamic shape optimization

F Sobieczky[1], A Lopez[1], B Scheichl[2], E Dudkin[1], C Lackner[3], M Hochsteger[3], C. Feichtinger[4], C. Leonhardsberger[4], H Sobieczky[2]

[1] Software Competence Center Hagenberg (SCCH), Softwarepark 32a, 4232 Hagenberg i. Mkr., AT
[2] Institute of Fluid Mechanics and Heat Transfer , TU Wien, T.BA/E322, Getreidemarkt 9, 1060 Vienna, AT
[3] CERBSim GmbH. Neubaugürtel 35/40, 1150 Vienna, AT
[4] Windpuls GmbH, Wiener Straße 131, 4020, Linz, Austria.

Abstract

For the purpose of finding the optimal design of a diffusor with respect to highest homogeneity in the flow of an air source heat pump a physics-informed reinforcement learning approach is developed for improved optimization performance of aeroynamic shape design. The physical insight about the homogeneous flow being preserved mostly under a low net degree of turbulence allows introducing a surrogate model yielding an extra term in the reward function. Specifically, it is shown that this physics informed reinforcement learning approach for a two-dimensional Navier–Stokes flow-problem yields an increase in optimization speed.

Selection and/or peer-review under the responsibility of the organizers of the 15th IEA Heat Pump Conference 2026.

Keywords: Reinforcement learning, physics-informed machine learning, aerodynamic shape optimization

IntroductionIntrodution

1.1 Related Work, Context: Physics Informedness in different Learning Models

Physics informedness has seen a rise in interest since 2019, when physics informed neural networks (PINNs [8]) have been demonstrated to be able to improve solving differential equations as well as help solving the corresponding inverse problem. Not only acceleration of the training speed of deep learning models, but also their interpretability in the form of 'physical explanations' [9, 10, 11] could be shown to be feasible: During the training of the deep learning model predicting a solution to a PDE, the corresponding optimization process also allows –by the use of automatic differentiation of the intermediate solution approximations– finding optimal physical parameters (e.g. a diffusion constant), or the coefficients in a parametrized form of the respective candidate differential equation. In the case of a flow-problem, for instance, apart from the approximation to the (forward problem) solution, it is possible to determine physical constants such as the viscosity of the fluid, i.e. the characteristic Reynolds number entering the Navier–Stokes equations governing ~~of~~ the problem, if unknown, beforehand. It is therefore also possible to solve inverse problems in the sense of finding the correct differential equation representing the physical law responsible for the process leading to the observed data.

Other approaches to the forward and inverse problem governed by differential equations, particularly those involving flow equations, include Hamiltonian Neural Networks [14,15,16], Lagrangian Neural Networks [39, 40], NeuralODEs [17, 18], and Neural Operators [19]. All of these methods allow for including physical conditions in the training process of learning model (forward problem), as well as inferring details about the physical processes from the predictions (inverse problem) relevant to completely describe the underlying differential equations (such as constants entering the coefficients of specific terms).

The recent adoption of the physics informed concept by reinforcement learning (RL) is particularly manifold in its approaches [7]. We follow the specific version of 'PIRL' defined by adopting the reward function associated with evaluating a single Markovian step in the associated dynamic optimization problem using a physical surrogate function. In this way we extend our former approach of using reinforcement learning for aerodynamic shape optimization [1] by embracing physics informedness in the form of letting domain knowledge be expressed by a modified reward. On the one hand, we condition on physically feasible solutions by using a surrogate approach, while on the other hand, we adaptively anticipate an appropriate update step-size during the optimization loop - both for an effective calculation performance.

1.2 Problem Description: Out 'pushing' Air Source Heat Pump Concept.

The goal of the research project (AIrFoil [1]) prompting the use of physics informed reinforcement learning is to design an optimized channel guiding air pushed by a ventilator toward the heat exchanger. This concept represents a new design approach as usually the air is sucked in by the ventilator through the heat exchanger. Its advantages include the reduced emission of noise and, therefore, a potentially higher rotations speed, eventually involving greater efficiency of the heat exchange process.

As a side problem, the transition from the circular to rectangular cross-sectional geometry involves the use of a diffusor-shaped channel guiding the air from the ventilator to the heat exchanger, in front of which it ideally exhibits a strongly homogeneous flow-field directed perpendicular to the duct's exit plane. More specifically, pronounced turbulence irregularities or insufficient alignment with the mean flow direction can seriously diminish the effectiveness of heat transfer. Applying a reinforcement learning method to a two-dimensional substitute problem to find the optimal diffusor shape, we study the effect of including aerodynamic domain knowledge regarding pressure loss induced by turbulence, i.e. due to viscous action. This approach poses a two-fold optimisation problem under the variation of geometrical parameters: finding a most uniform exit flow for, simultaneously, the smallest possible overall pressure loss. Treating these conflicting goals as fully independent ones, typically only allows for a classical Pareto optimum. However, here the uniform outlet flow represents the primary goal.

The optimization problem addressed in this paper assumes constant outlet pressure boundary conditions and uniform constant horizontal inlet flow-field. We use the method of mass conserving mixed stress formulation (MCS) for the computation of time-dependent implicit large eddy (ILES) solutions to the flow-field [2, 3]. The solutions are averaged over time and used in the analysis [4] of the conditions for the highest homogeneity at the exit plane. Specifically, the typical Reynolds number (Re) at play, formed with the channel length, the volume flow and the kinematic viscosity (air at standard conditions), is of about 105. In a realistic scenario, the flow further upstream encounters a rather involved guiding geometry but is free of (undesired) large-scale separation. The characteristic Reynolds number there attains values of similar magnitude such that the flow at the inlet exhibits typical features of turbulence. Hence, the inlet flow can be safely taken as uniform (and aligned with the x-direction), provoking just forming boundary layers adjacent to the guiding walls. We hence arrive at an accordingly non-dimension flow problem, essentially controlled by Re and a few geometry parameters. Given the stationary boundary, in- and outflow conditions, it principally governs a steady diffusor flow.

It is expedient, however, to adopt a transient approach, where under-resolved (to avoid issues of temporal stability) direct numerical simulation (DNS) is the means of choice. That is, in first step, a steady Stokes flow (virtually Re = 0) is computed, which serves as initial condition. Due to the largeness of Re, cumulative round-off errors and other numerical imperfections ("noise") give rise to a certain level of fluctuations in the transiently obtained solution, which remain for arbitrarily large times. Once the system has transitioned out of the transient state, these fluctuations define a corresponding level of physically manifest turbulence intensity. From this instance of time onwards, the DNS indeed simulates the real turbulent flow. Finally, time-averaging over this second, sufficiently large time interval yields the sought Reynolds-averaged flow-field and thus the averaged outlet flow, which shall ideally be also uniform.

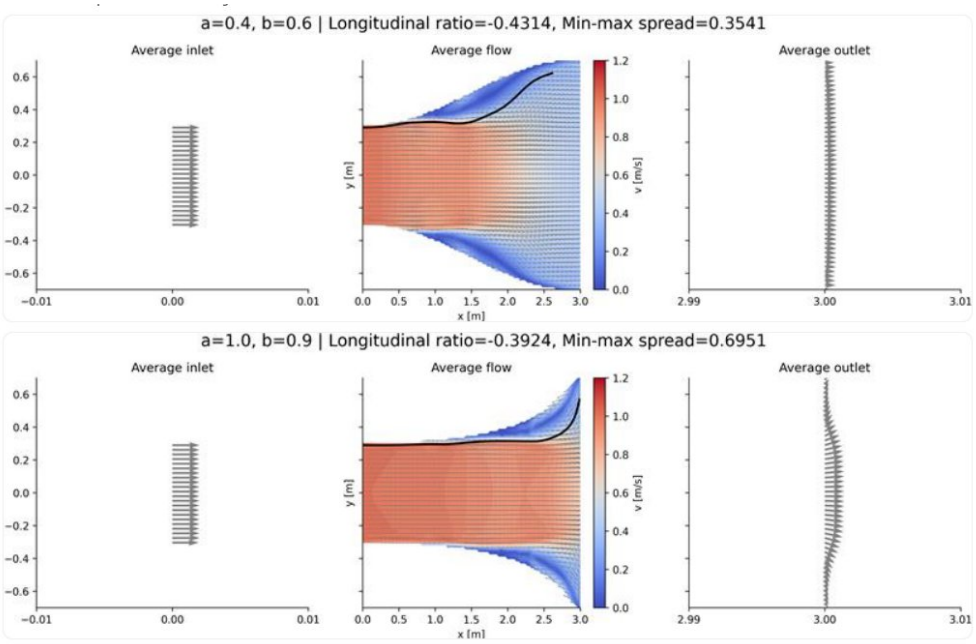

Figure 1: Two candidats of diffusors with walls shaped by cubic two-parameter Bezier-curves with guiding air-flow from left to right at a Reynolds number of 10^5 with ILES calculations of time-dependent solutions to the Navier–Stokes problem. Detachment is seen to occurr at different points along the channel's wall. Left – The flow-field close to the inlet. Middle – Flow-field calculations with a single stramline exhibiting the separation point. Right – Resulting flow-field at the outlet – of which the highes homogeneity is desired.

1.3 Surrogate Models and Physics Informedness: Which Surrogate to Use

As seen in Figure 1, the less favourable (lower) configuration in terms of outlet flow-field homogeneity exhibits a separation point further away from the inlet compared with the configuration shown in the upper half. While a 'late detachment' is usually an indicator of lesser turbulence and a favourable configuration for the prediction of separation in subsonic turbulent flows past airfoils see e.g. the classical textbooks it is not a 'diagonstic marker' in the present context, i.e., for the ultimate goal of achieving a homogeneous outlet flow-field. While often physics-informedness may cling to a 'rule-of-thumb' knowledge about flow-field solutions, the example shows that the desired optimization goal generally needs to be supported by a precise, coherent measure of the desired flow quality.

Materials and methods

2.1 Problem formulation

The fluid dynamics design is formulated as a shape optimization problem

$$\min_{(a,\,b)\in G} f(a,b), \tag{1}$$

where $G \subset [0,\,1]^2$ denotes a bounded discrete grid of parameter pairs $(a,\,b)$ defining the curvature of the channel profile. The cost function $f(a,\,b)$ measures the longitudinal homogeneity of the outlet velocity field obtained from CFD simulations of a channel shaped according to $(a,\,b)$. In what follows, we briefly recall the reinforcement learning (RL) framework and subsequently formulate the shape optimization under this setting.

The Markov Decision Process (MDP) is a mathematically tractable approach for RL [5, 6]. state space S, an action space A, and a family of conditional probabilities π: $S \times A \rightarrow [0,\,1]$, referred to as policies. At a given time step n, a policy assigns the probability $\pi(a_n,\,s_n)$ of choosing the action a_n conditioned on the system being in state s_n (written as $\pi(a|\,s)$). Afterwards, the next state s_{n+1} is chosen independently, according to the

transition probability $P(s_n, a_n, s_{n+1})$ with P: S×A×S $\rightarrow$[0, 1], defined as $P(s_{n+1} = s' \mid s_n, a_n)$. Finally, decisions are guided by an immediate reward $R(s_{n+1}, a_n, s_n)$, where R : S×A×S $\rightarrow$ R. The resulting MDP dynamics generates a (random) trajectory $s_1, a_1, r_2, s_2, a_2, \ldots s_0, a_0, r_1$ and the objective is to find the optimal policy π^* that maximizes the expected discounted return

$$V_\pi(s) = E\left[\sum_{n \in N_0} \gamma^n R(s_{n+1}, a_n, s_n) \mid s_0 = s\right] \tag{1}$$

where the expected value is computed using the Markov (product) transition measure $\pi(a|s) \cdot P(s' \mid s, a)$.

It can be shown that the optimal policy satisfies the well-known Bellmann optimality equation

$$V_{\pi^*}(s) = \max_{a' \in A} \sum_{s' \in S} P(s' \mid s, a')\,[R(s', a', s) + \gamma V_{\pi^*}(s)], \tag{2}$$

which is typically solved using iterative methods.

To address problem (1) under the MDP-RL framework introduced above, we define the state space as $S = G$ and the action space as $A = \{(-\delta, 0), (\delta, 0), (0, -\delta), (0, \delta)\}$, which encodes the four nearest-neighbor directions (left, right, up and down), where δ denotes the grid spacing. We further define the transition matrix as $P(s_{n+1}, a_n, s_n) = 1$ if $s_{n+1} = s_n + a_n$ and $P(s_{n+1}, a_n, s_n) = 0$ otherwise, allowing transition from a given state (shape) to any of its four nearest neighbors. Finally, the reward is $R(s_n, a_n, s_{n+1}) = R(s_{n+1}) = -f(s_{n+1})$, i.e, the negative value of the cost optimization function. Observe that, in this case, (2) simplifies to

$$V_{\pi^*}(s) = \max_{a' \in A} \sum_{s' \in S} [R(s + a') + \gamma V_{\pi^*}(s)]. \tag{3}$$

We employ a simple multivariate parabolic regression model to locally approximate the proposed reward function over a neighborhood of fixed radius in the parameter space. Physical information is incorporated at the level of the quadratic regression model, which is fitted locally at each step and subsequently used to predict the maximum reward (i.e., minimal cost) within the currently considered neighborhood. In particular, a measure of pressure loss is used to condition the solutions toward configurations exhibiting low turbulence. Figure 1 illustrates the CFD velocity field at the inlet, interior and outlet of the channel for the parameters $(a, b) = (0.4, 0.6)$ and $(a, b) = (1.0, 0.9)$, where the profile of the diffusor walls is defined by the cubic Bezier curve $\boldsymbol{B}(s) = \sum_{i=0\ldots3} \mathbf{p}_i B_i^{(3)}(s)$ with the cubic Bezier-basis $B_i^{(3)}(s) = \binom{3}{i} s^{3-i}(1-s)^i$, and the control points $\mathbf{p}_0 = (0, r)$, $\mathbf{p}_1 = (a, r)$, $\mathbf{p}_2 = (b, R)$, $\mathbf{p}_i = (L, R)$, with fixed channel length (L), inlet (r) and outlet radius (R).

The scatter plot in Figure 2 illustrates the relationship between the outlet cost (vertical axis) and the pressure loss (horizontal axis) for all CFD realizations over the shape parameter space G. The data points are color-coded by the parameter difference $b - a$. The pressure loss is defined as the difference between the ideal (Bernoulli) increase in static pressure and the observed pressure drop between the inlet and outlet planes, namely

$$PL(s) = (P_{ideal}(entry) - P_{ideal}(exit)) - (P_{obs}(entry) - P_{obs}(exit)). \tag{4}$$

It can be observed that $b - a$ is a numerical feature strongly correlated with the observed pressure loss, however, close to the minimum of the cost function, a PL even closer to zero can be achieved while the homogeneity of the velocity magnitudes decreases again, seen by the rise of the cost value $f(a, b)$. Therefore, a surrogate is chosen, which corrects the cost function by the (negative) pressure drop as long as $b - a < -0.1$ where the desired monotonicity between PL and f (and therefore also between $b - a$ and f) prevails.

Figure 2 reveals the strong correlatedness of the spread (difference between minimum and maximum shown on Y-achsis) of the flow-field's velocity vector in absolute value and the difference $b - a$ between the geometric parameters a and b. However, it is also seen that this monotonic relationship ceises to hold close to the opimal (minimal) value of the spread, around pressure difference values of − 0.1Pa. This has to be respected in the surrogate modeling using the $b - a$ as an input variable for physical conditioning in the estimation of the value function.

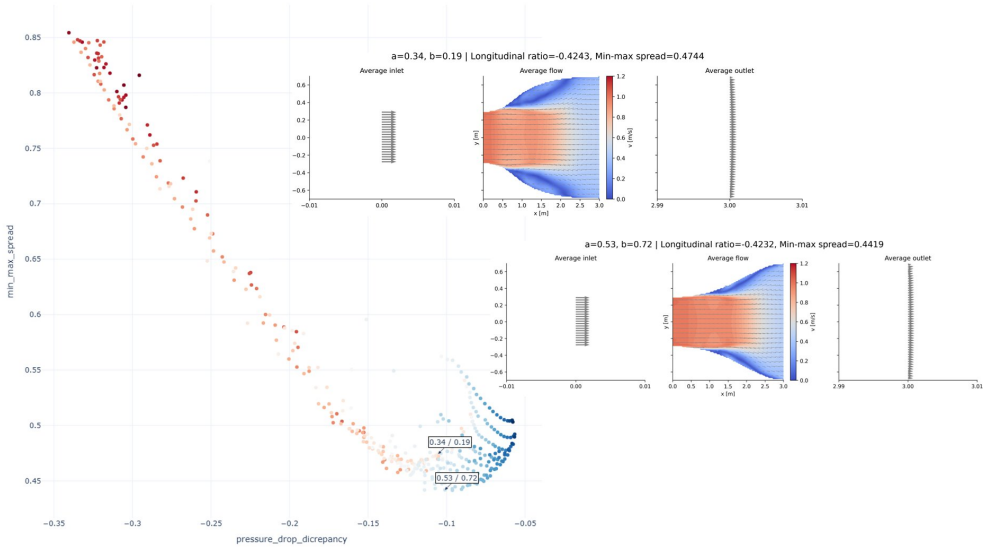

Figure 2: Evaluations of the cost function $R_0(a, b)$ = mean_max_spread (used as a negative reward function) in m/s which measures the degree of homogeneous magnitudes of the velocity vectors in the exit plane as a function of the pressure difference given in Pa. The two parameters a and b define two of four control points of a cubic Bezier function with vanishing gradients at the entry and exit plane. A value close to zero (or one) lets the corresponding control points be located strongly on the left (or, respectively, right), making $(a, b) = (0,1)$ correspond to the linear (non-curved) wall of the diffusor shape. The points' color coding refers to the difference $b - a$. Note the change in pressure of a factor of approx. 2 corresponds with the reduction in speed by a factor of approx. 1/2, by Bernoulli's law in the case of the density of air ($\pm\ \rho = 1.24 kg/m^3$), resulting in approx. $(\frac{1}{2} - \frac{1}{2}\frac{5}{4}\left(\frac{1}{4}\right))\ Pa \sim \frac{11}{32} Pa \sim 0.325\ Pa$.

2.2 Algorithm

We use the concept of a 'mesoscopic scale', which refers to neighborhoods of parameter-vectors of a certain small enough size (radius), larger than steps between adjacent points on the grid, but small enough so that on this neighbourhood, the reward function, (hence the cost function), can be modeled well by a polynomial regression model R_{surr}. For a given current point $s_n = (a_n, b_n)$ in the parameter space S, we search within a small square-grid neighborhood $N(s_n, r_n) = \{s' : |s' - s_n| < r_n\}$ for a minimum of the surrogate of the true cost function. More precisely, we consider the auxiliar MDP-RL problem with state space restricted to $N(s_n, r_n)$. We then compute the optimal policy π^* and associated value function $V_{\pi*}$ based on (3), using the restriction of P to $N(s_n, r_n)$ and the surrogate regression reward R_{surr} fitted over the neighborhood. We compute the optimal value function and the corresponding policy using the well-known value-iteration algorithm, combined with a softmax (Boltzmann) policy update of the form

$$\pi(s,a) = \frac{exp(-\beta max(V(s+a-V(s), 0))}{\sum_{a' \in A} exp(-\beta max(V(s+a'-V(s), 0))} \quad (5)$$

The parameter βcontrols the degree of exploration: small values of β yield a more exploratory (nearly uniform) policy, facilitating escape from local minima of the value function, whereas larger values increasingly favor greedy action selection.

Once the value function approximation has converged to an estimate $\hat{V}_n(s)$, we compute the estimator of $V(s)$ over the local neighborhood $N(s_n, r_n)$

$$s_{n+1} = argmin_{s \in N(s_n, r_n)} \hat{V}_n(s), \quad (6)$$

which is typically attained at the boundary of the neighourhood. The process repeats considering an updated neighbourhood $N(s_{n+1}, r_{n+1})$ centered at the current selected state s_{n+1}. See Figure 4 on the right for a typical 'mesoscopic path' (cf. [1]). for an iteration of this process.

The hat (caret) on the value function indicates that it is only approximated by function iteration using the surrogate of the cost function restricted to the neighborhood $N(s_n, r_n)$. The three functions of Figure 3 show the cost function evaluated on the considered 21x21 sub-grid of the uni-square of possible values of

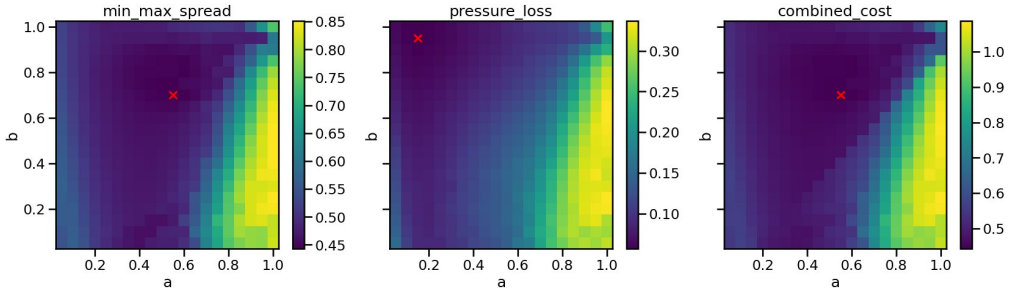

Figure 3: Left - The cost function $R_0(a,b)$; Middle - The pressure loss function PL; Right - the combined cost function $R(a,b) = R_0(a, b) + 1_{b\text{-}a<\text{-}0.1}(a,b) \times (PL(a,b) - PL(a,a - 0.1))$. The color coding of the graph in Figure 2 suggests this rule as it prevents the influence of PL where it doesn't accelerate convergence to the minimum of R_0.

It is seen, how a mesoscopic path with equal sized neighborhoods involves finding solutions (6) on very differently curved restrictions of the estimated value function on the neighbourhoods. In other words, the gradients of $\hat{V}(s)$ restricted to $N(s_n, r_n)$ are very differently curved, which is reflected in very different parabolic approximations. To take advantage of this circumstance in order to accelerate our optimization algorithm by reducing the number of mesoscopic steps is the subject ot the experiments carried out in order to use the physical information ($PL(s)$) advantageously.

3. Results

3.1 Description of the Experiment

Using ideas from [1], we evaluate the exact value function on four points (midpoints of the four bounding sides of the current neighnorhood) and use these values together with that of the current neighbourhood's midpoint to interpolate all other values on that neighborhood by a parabolic regression model (6).

$$\hat{V}(s = (x, y)) = \langle s, H\, s\rangle + \langle v, s\rangle + c \tag{6}$$

The parameters $H_{11}, H_{12} = H_{21}, H_{22}, v_1, v_2, c$ are calculated so as to interpolate between the value function at the five points given by $\theta_n, \theta_n - re_1, \theta_n + re_1, \theta_n - re_2, \theta_n + re_2$ and the interpolating function

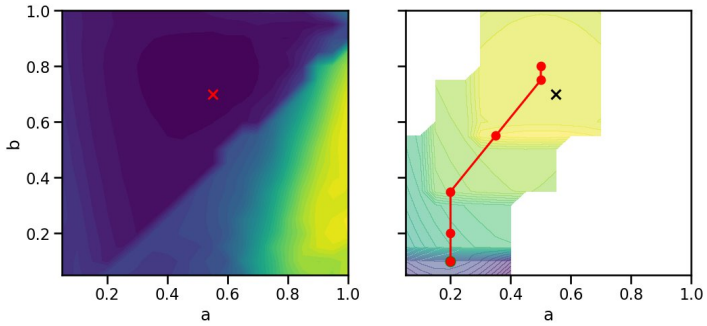

Figure 4 Left - The estimated value function $\hat{V}(s)$ derived only from (1) and using $R_0(a, b)$ in Step 1; Right - A 'mesoscopic path' of found minima θ_n found in the iteration of predicting the cost function, locally, on $N(\theta_{n-1}, r_{n-1})$. In this example, the 'physical information' of the pressure loss is not used ($PL(\cdot)$ doesn't enter the calculation of $\hat{V}(s)$). Also, the radii are held fixed at a constant value of $r_n \equiv 3$, inducing patches of 25 (24 neighbouring) states s.

is $\hat{V}(s)$, with $s \in \boldsymbol{N(\theta_n, r)}$. It is a *surrogate model* as it predicts the true value function locally based on physical ground truth simulated at a few points in the neighborhood. We recall that simulations are computationally expensive which is why working with a (sufficiently accurate) prediction $\hat{V}(s)$ accelerates the global optimization process.

Our experiment concerns the aim of speeding up the optimization process of finding the global minimum of $V(s)$ by reducing the number of steps in the mesoscopic path. We propose to allow for varying radii $r = r_n$ of the local neighborhoods $N(\theta_n, r_n)$. Namely, by choosing the following rule, we achieve increases of the size of the neighbourhoods on 'flat' stretches during the walk in the parameter space, whence the gradient of $\hat{V}(s)$ is locally close to constant. The intuition of this statement lies in the notion familliar from Newton's method, that a little curved surface allows larger steps, or similarily, paths following the steepest descent on a linearly (flat) defined surface can be undertaken by a single, larger step. We consider two radii, one small one, and another, bigger one, i.e., $r_n \in \{r_{small}, r_{large}\}$:

Definition: *We call the* ***total amount of curvature*** *of a function of type (6) the l_1-norm of the vector of eigenvalues of the Hessian H: $C(\hat{V}) := |\lambda_1| + |\lambda_2|$, where $\{\lambda_1, \lambda_2\}$ is the spectrum of H.*

RULE A(ρ): *If the total amount of curvature of the approximated value function $\hat{V}$ on the small current neighborhood $N(\theta_n, r_{small})$ around θ_n is smaller or less than the threshold ρ, then pass on to considering finding the next optimal parameter θ_n on the large neighbourhood $N(\theta_n, r_{large})$. In the other case, if there is a total amount of curvature larger than ρ for $\hat{V}(s)$ restricted to the small neighborhood of θ_n, keep that small radius and use $N(\theta_n, r_{small})$ in (5) to find the next optimal parameter θ_n.*

Note that this rule is designed so as to decide on the currently used radius r_n only on the basis of the curvature information on the *small* neighbourhood. The reason for this choice of action is that evaluating the correct value function $V(s)$ for a single parameter vector $s = \langle a, b \rangle$ is computationally expensive as the corresponding flow-field has to be computed. In the case that the small radius is chosen by RULE A(ρ), the flow-field computations at $\theta_n - re_1$, $\theta_n + re_1$, $\theta_n - re_2$, $\theta_n + re_2$ for $r = r_{large}$ don't have to be performed.

We now apply RULE A(ρ) and use the cost function $R(s)$ for the initial function in the iteration to compute the value function (see Figure 3). The resulting performance on the overall optimization process is a shortening of the mesoscopic path, as seen in the following results.

Figure 4 also illustrates the advantage of the adaptive concept in connection with the physics-informed. The performance of the iterated optimization to find the global minimum under conditioning of the surrogate which keeps the mesoscopic path of parameter changes to migrate towardts the local minimum at $\langle 0.5, 0.1 \rangle$. For comparison, Figure 5 shows the same initial parameter pair $\langle 0.2, 0.1 \rangle$ without the physics-informed influence and a fixed radius $r_n \equiv r$, leading to an excursion induced by this non-convexity of the value function (without the help of the domain knowledge) under RULE A.

3.2 Influence of RULE A

The use of adaptiveness in the choice of the radius can be decribed in the reinforcement learning context in

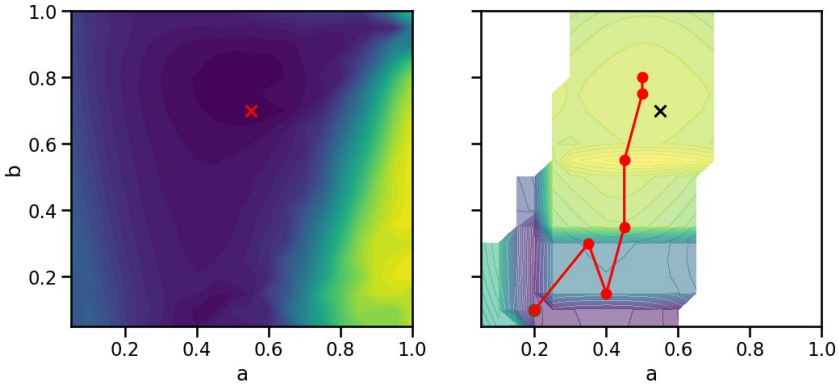

Figure 5: Left - Value function estimate $\hat{V}(s)$ on the whole parameter space for the given cost function in Step 1 equal to $R_0(s)$, ***not*** *including the information about the pressure loss (PL(·)). Right - Induced mesoscopic path (θ_n) of positions of minima of $\hat{V}(s)$, with $s \in N(\theta_{n-1}, r_{n-1})$. The radius $r_n \equiv r$ is is chosen adaptively according to RULE A(15).*

the form of policy re-evalutaion [6] (see Section 3.1): The action space A, instead of being restricted to the transitions between the states in $N(\theta_n, r)$, i.e., $A = \{a = s' - s \mid s, s' \in N(\theta_n, r)\} =: A(r)$ is replaced by there being two sets $A(r_{small})$, and $A(r_{large})$. The policy $\pi(s_{n+1} | a_n, s_n) = P_{s_n, s_{n+1}} s_{n+1} s_{n+1}$ is re-evaluated after each

mesoscopic step and changed to be valid on the newly chosen action set $A(r_n)$. Since there is no training of a model learning the decision of how to make the best choice in the re-evaluation, our approach is *model-free*. Also, it is *online*, as it collects data following only the latest policy.

4. Discussion

4.1 Quantitave comparison with gradient decent using phyics-informedness in reward function

In a sweep of initial parameter-pairs from $\{0.2, 0.4, 0.6, 0.8\} \times \{0.1, 0.3, 0.5, 0.7, 0.9\}$ we evaluate the length of the mesoscopic paths using either the standard or the physics-informed and adaptive technique. We compare the mesoscopic path lengths for the fixed radius shape optimization process without the help of the information *PL* on the pressure loss with the same sequence of initial values with several runs with the physics-informed measure (4) using the approximation *Estimate_V(R)* of the value function $V(s)$, where $R(s)$ is the cost function defined in the caption of Figure 3.

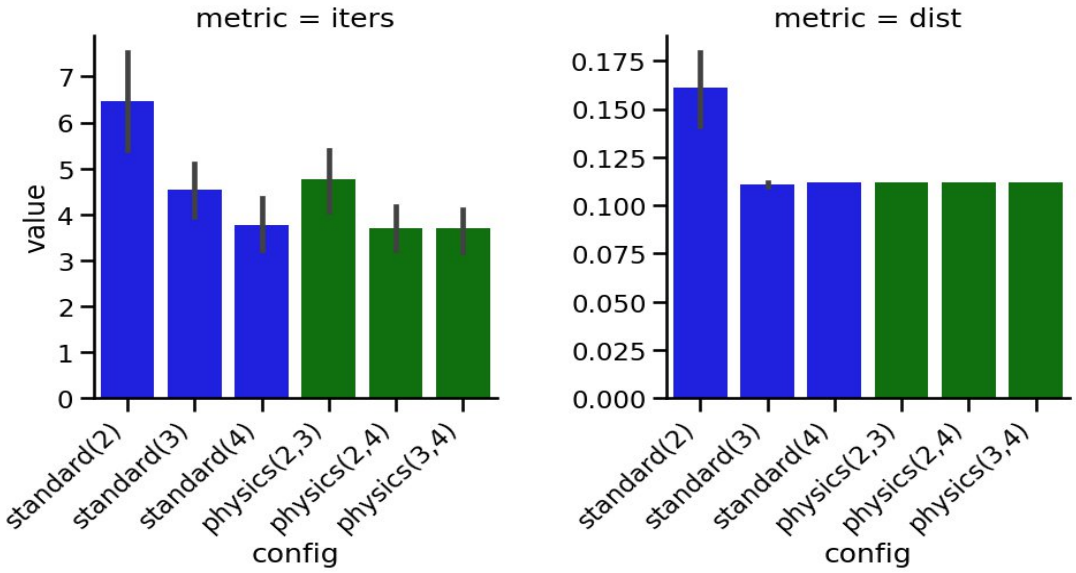

Figure 6: Left - Result of the average lengths of the mesoscopic paths from the initial parameter-pairs to the approximated location for various configurations of selections of fixed window sizes (blue) and pairs of small and large radii for adaptive choices of the radii (green) and physics-informed measures $P_{s,s'}$. Right – Average distance of the found global minimum and true minimum of the estimated value function.

From the resulting mean (with stan. dev.) mesoscopic path lengths and their standard deviations, we see that the positive effects of using an adaptive, physics-informed RL approach to finding the optimum is generally a general decrease of the mean and a decrease in standard deviations when switching from a standard configuration to an adaptive, physics-informed configuration in which there is a higher larger radius.

We also present (Figure 6, right) the difference in the estimated global minimum from the location of the true minimum in the parameter space. The iteration of finding the global minimum stops at a breaking criterion involving the comparison of the change of the surrogate $\widehat{V}(s)$ *estimate* at two consecutive steps. If there is no further change to this estimate, the then current best candidate for the global minimum is taken as the result of the global optimization problem. It is seen that the smallest radius of 2 (with 9 vertices in the corresponding neighbourhoods) entails stronger discrepancies between estimate and true function value $V(s)$ than the other configurations, and that there is no dispersion in this distance with the physics-informed configuations.

4.2 Summary of the effects

The example of switching from a radius 3 standard to an radius 2-4 adaptive physics-informed configuration represents an improvement of the mean mesoscopic path length to drop from 4.4 to about 3.9 and entails not

only the enlargement of the radius with the possibility of larger steps is helpful to an accelerated optimization process, but also locally reducing the neighbourhood represents a potential improvement for the directness of the mesoscopic path, as in the case of strongly varying valuef unctions the step in a direction pointing away from the global minimum is reduced. Figure 5 shows the possibility of getting caught in a local minimum. In this case, the use of the physics-informed measure (4) helps additionally to condition the random walk to move away from the non-convex regions of the initial reward function.

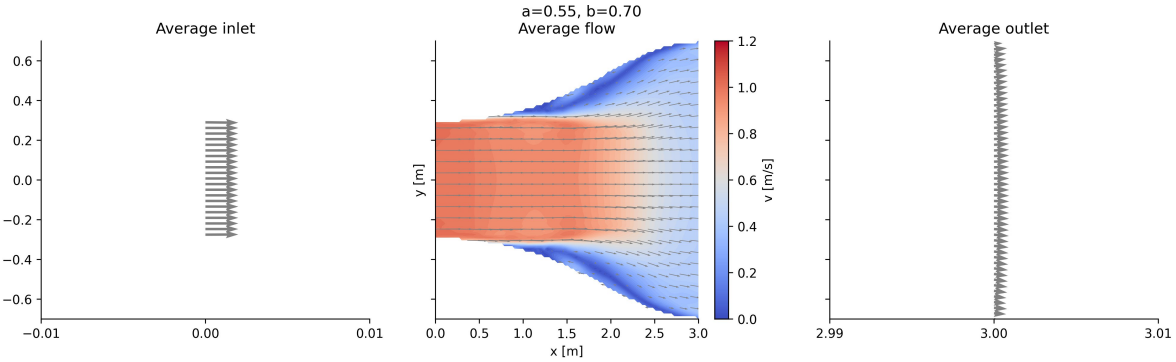

Figure 7: Optimal shape belonging to the 'true' minimum of the cost functionR(s) as shown by the red cross in Figure 3, Left.

5. Conclusion and Future Work

In this paper we show that physics-informed reinforcement learning methods can lead to improved aerodynamic shape optimization performance. We realize a model-free, online Monte Carlo driven parameter optimization method, which uses physical information in the measure used to weigh the possible actions which is employed for estimating the value function (1).

The effect of the adaptive technique

- improves the directness (reduces the length) of the mesosopic path towards the global minimum,
- can be formulated as a policy re-evaluation method within the RL framework allowing for conditioning on physically plausible solutions identifie.

The benefits of the physics-informed measure $P_{s,\,s'}$ are:

- It 'flattens' the function $R_0(s)$ inducing the possibility for larger radii (due to locally reduced total curvature).
- In the parts of the parameter space in which the additional cost function can be applied without a change of the optimum (homogeneity of the flow-field at the heat-exchanger), the pressure loss can be used advantageously to exclude geometric shapes which have physical effects that are not in favour of homogeneity at the exit-plane.

We conclude that the relationship between pressure difference and geometric parameters can be used even though its monotonicity holding only remotely from the precise optimum (Figure 2). In this way physical insight into the physics of the flow problem can be used to accelerate the efficiency of the optimization procedure without impairing the accurate prediction of the optimum.

Acknowledgements

The main contributor to this research is the FFG (Austrian Applied research fund) project "AIrFoil" (FFG-No. 915010). The research reported in this paper has also been partly funded by the Federal Ministry for Innovation, Mobility and Infrastructure (BMIMI), the Federal Ministry for Economy, Energy and Tourism (BMWET), and the State of Upper Austria in the frame of the SCCH competence center INTEGRATE [(FFG grant no. 892418)] in the COMET - Competence Centers for Excellent Technologies Programme managed by Austrian Research Promotion Agency FFG.

References

[1] Florian Sobieczky, Alfredo Lopez, Erika Dudkin, Christopher Lackner, Matthias Hochsteger, Bernhard Scheichl, Helmut Sobieczky. Reinforcement Learning for Accelerated Aerodynamic Shape Optimisation. arXiv:2507.17786 [cs.LG], 8686 (Contribution to AIFluids: https://www.aifluids.net/proceedings/S14P2.pdf)
[2] Philip L. Lederer, Xavn er Mooslechner, Joachim Schöberl, High-order projection-based upwind method for implicit large eddy simulation, Journal of Computational Physics, Volume 493, 2023, 112492, ISSN 0021-9991, https://doi.org/10.1016/j.jcp.2023.112492.
[3] Jay Gopalakrishnan, Philip L Lederer, Joachim Schöberl, A mass conserving mixed stress formulation for the Stokes equations, IMA Journal of Numerical Analysis, Volume 40, Issue 3, July 2020, Pages 1838–1874, https://doi.org/10.1093/imanum/drz022
[4] Joachim Schöberl, Christopher Lackner, Matthias Hochsteger. NGSolve – Multiphysics Simulation Software, https://www.researchgate.net/publication/266797024_C11_Implementation_of_Finite_Elements_in_NGSolve, and Mesh-generation: Netgen: https://link.springer.com/article/10.1007/s007910050004
[5] Dimitri P. Bertsekas, *Nonlinear Programming Vol. II: 3rd Edition*. 2016, Sect. 2.3, ISBN: 978-1-886529-05-2.
[6] Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction*, Second edition, Adaptive Computation and Machine Learning Series (Cambridge, Massachusetts: The MIT Press, 2018).
[7] Banerjee, C., Nguyen, K., Fookes, C., & Raissi, M. 2023. *A Survey on Physics Informed Reinforcement Learning: Review and Open Problems*. arXiv preprint arXiv:2309.01909. https://arxiv.org/abs/2309.01909
[8] Raissi, M., Perdikaris, P., & Karniadakis, G. E. 2019. *Physics-Informed Neural Networks: A deep learning framework for solving forward and inverse problems involving nonlinear PDEs*. Journal of Computational Physics, 378, 686-707. https://doi.org/10.1016/j.jcp.2018.10.045
[9] Lawal, Z. K., Yassin, H., Lai, D. T. C., & Che Idris, A. 2022. *Physics Informed Neural Network (PINN) Evolution and Beyond: A Systematic Literature Review and Bibliometric Analysis*. Big Data and Cognitive Computing, 6(4), 140. https://doi.org/10.3390/bdcc6040140
[10] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. 2021. *Physics-informed machine learning*. Nature Reviews Physics, 3, 422-440. https://doi.org/10.1038/s42254-021-00314-5
[11] Cuomo, S., Schiano di Cola, V., Giampaolo, F., Rozza, G., Raissi, M., & Piccialli, F. 2022. *Scientific Machine Learning through Physics-Informed Neural Networks: Where we are and What's next*. Journal of Scientific Computing, 92(1), 1-52. https://doi.org/10.1007/s10915-022-01939-z
[12] Cranmer, M., Greydanus, S., Hoyer, S., Battaglia, P., Spergel, D., & Ho, S. 2020. *Lagrangian Neural Networks*. arXiv preprint arXiv:2003.04630. https://arxiv.org/abs/2003.04630
[13] Ramesh, A., & Ravindran, B. 2023. *Physics Informed Model Based Reinforcement Learning*. In Proceedings of the 5th Annual Learning for Dynamics and Control Conference (L4DC), Proceedings of Machine Learning Research, 211, 26-37. https://proceedings.mlr.press/v211/ramesh23a.html
[14] Greydanus, S., Dzamba, M., & Yosinski, J. 2019. *Hamiltonian Neural Networks*. Advances in Neural Information Processing Systems, 32, 15379-15389. https://arxiv.org/abs/1906.01563
[15] Moradi, S., Jaensson, N., Tóth, R., & Schoukens, M. 2023. *Physics Informed Learning Using Hamiltonian Neural Networks with Output Error Noise Models*. arXiv preprint arXiv:2305.01338. https://arxiv.org/abs/2305.01338
[16] Bajaj, C., & Nguyen, M. 2021. *Physics Informed Neural Networks via Stochastic Hamiltonian Dynamics Learning*. arXiv preprint arXiv:2111.08108. https://arxiv.org/abs/2111.08108
[17] Chen, R. T. Q., Rubanova, Y., Bettencourt, J., & Duvenaud, D. 2018. *Neural Ordinary Differential Equations*. In Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018). https://arxiv.org/abs/1806.07366
[18] Zhou, K., Tang, Y., & Karniadakis, G. E. 2021. *Physics-informed neural networks for high-dimensional parametric PDEs with neural ODEs*. Journal of Computational Physics, 442, 110469. https://doi.org/10.1016/j.neunet.2025.107166
[19] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, Anima Anandkumar. Neural Operator: Learning Maps Between Function Spaces. The Journal of Machine Learning Research (2023), Volume 24, Issue 1, Article No 89, pp 4061-4157