Constrained aerodynamic shape optimization using neural networks and sequential sampling

Pavankumar Koratikere[1], Leifur Leifsson[1[0000−0001−5134−870X]], Slawomir Koziel[2,3[0000−0002−9063−2647]], and Anna Pietrenko-Dabrowska[3[0000−0003−2319−6782]]

[1] School of Aeronautics and Astronautics, Purdue University, West Lafayette, Indiana 47907, USA
{pkoratik@purdue.edu, leifur@purdue.edu}

[2] Engineering Optimization & Modeling Center, Department of Engineering, Reykjavík University, Menntavegur 1, 102 Reykjavík, Iceland
koziel@ru.is

[3] Faculty of Electronics Telecommunications and Informatics, Gdansk University of Technology, Narutowicza 11/12, 80-233 Gdansk, Poland
anna.dabrowska@pg.edu.pl

Abstract. Aerodynamic shape optimization (ASO) involves computational fluid dynamics (CFD)-based search for an optimal aerodynamic shape such as airfoils and wings. Gradient-based optimization (GBO) with adjoints can be used efficiently to solve ASO problems with many design variables, but problems with many constraints can still be challenging. The recently created efficient global optimization algorithm with neural network (NN)-based prediction and uncertainty (EGONN) partially alleviates this challenge. A unique feature of EGONN is its ability to sequentially sample the design space and continuously update the NN prediction using an uncertainty model based on NNs. This work proposes a novel extension to EGONN that enables efficient handling of nonlinear constraints and a continuous update of the prediction and prediction uncertainty data sets. The proposed algorithm is demonstrated on constrained airfoil shape optimization in transonic flow and compared against state-of-the-art GBO with adjoints. The results show that the proposed constrained EGONN algorithm yields comparable optimal designs as GBO at a similar computational cost.

Keywords: Aerodynamic shape optimization · global surrogate modeling · neural networks · sequential sampling

1 Introduction

The goal of aerodynamic design is to find a shape (or adjusting an existing one) in airflow (such as as airfoils, wings, helicopter rotor blades, wind turbine blades, and the external shape of aircraft) that improves a given quantity of interest (QoI) (such as the drag force), while adhering to appropriate constraints (such as a specified lift force) [10,25,21]. Computational aerodynamic

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_33

design involves the use of computer simulations of the airflow past the shape, using computational fluid dynamics (CFD), to numerically evaluate the QoI and the associated constraints [22]. Aerodynamic shape optimization (ASO) is the automation of computational aerodynamic design by embedding the computer simulations within an optimization framework to search for the constrained optimal shape [8]. Key challenges of ASO include (1) time-consuming simulations, (2) a large number of design variables, (3) a large number of constraints, and (4) many model evaluations.

The state-of-the-art ASO is gradient-based optimization (GBO) with adjoints [7]. The main advantage of the adjoint approach is that the cost of a gradient calculation can be made nearly independent of the number of design variables. In the context of solving ASO problems with GBO and adjoints, this means that for each design evaluation the objective function and constraints need to be computed, which involves one primal computer simulation and one adjoint simulation for the objective and each constraint. For example, if there is one objective and two constraints, GBO with adjoints needs one primal computer solution and three adjoint solutions. Typically, the time per one primal CFD solution is roughly the same as one adjoint CFD solution [23]. This means, for the given example, that each design evaluation involves four simulations which yields the objective function value, the constraint function values, and the gradients of the objective and constraints. This is independent of the number of design variables, which renders the approach scalable to high-dimensional problems. It should be noted, however, that the computational cost grows quickly with the number of constraints.

Another way to solve ASO problems is to use surrogate-based optimization (SBO) where a surrogate replaces the time-consuming simulations in the computation of the objective and constraint functions (as well as its gradients, if needed) [26]. SBO has been around for a long time. A widely used approach is the efficient global optimization (EGO) algorithm [9]. In EGO, the design space is sampled initially using design of experiments [26], such as Latin hypercube sampling, and an initial surrogate model is constructed using kriging [2]. The kriging surrogate is then iteratively improved by sequentially sampling the design space using both the prediction and prediction variance. The key advantage of using kriging is that it can improve the surrogate accuracy for a given number samples since the samples can be assigned to regions where the surrogate shows poor accuracy (exploration) or where a local minimum is found (exploitation). A key disadvantage of kriging is that the computational cost grows quickly with the number of samples [15].

Neural network (NN) regression modeling [5], on the other hand, scales more efficiently for large data sets [20,29]. A major limitation, however, is that uncertainty estimates are not readily available for a single prediction [20], and it is necessary to make use of an ensemble of NNs with a range of predictions [18,32,4] or use dropout to represent model uncertainty [3]. These algorithms are, however, computationally very intensive.

A recently created EGO algorithm with neural network (NN)-based prediction and uncertainty (called EGONN) partially alleviates these challenges [14]. In EGONN, a NN model is utilized to approximate nonlinear high-dimensional objective functions. The unique feature of EGONN is its ability to sequentially sample the design space and continuously update the NN-based prediction with a prediction uncertainty which is modeled by a second NN.

In this paper, a novel extension to EGONN is proposed that enables efficient handling of nonlinear constraints. Furthermore, the utilization of data for the prediction and prediction uncertainty is made more efficient by a continously updating all data sets. The EGONN algorithm only sequentially updated the prediction data set. The proposed constrained EGONN (cEGONN) algorithm is demonstrated on an airfoil shape optimization problem in transonic flow involving one objective, two constraints, and thirteen design variables. The proposed algorithm is compared against state-of-the-art GBO with adjoints.

The next section describes ASO using state-of-the-art GBO and the proposed cEGONN algorithm. The following section presents the numerical results of the constrained airfoil shape optimization problem using those algorithms. The last section provides concluding remarks and possible next steps in this work.

2 Aerodynamic shape optimization

This section states the ASO problem formulation, and then describes its solution with GBO and adjoints and with the proposed cEGONN algorithm.

2.1 Problem formulation

ASO involves minimization of the drag coefficient (C_d) of a baseline airfoil at a fixed free-stream Mach number1 (M_∞) and Reynolds number (Re_∞) with respect to design variables ($\mathbf{x}$) controlling the shape, subject to inequality constraints on the lift coefficient (C_l), and the airfoil cross-sectional area (A). Specifically, the constrained nonlinear minimization problem is formulated as:

$$\min_{\mathbf{x}} C_d(\mathbf{x})$$

subject to

$$\begin{aligned} C_l(\mathbf{x}) &\geq C_{l_{ref}} \\ A(\mathbf{x}) &\geq A_{ref} \end{aligned} \tag{1}$$

and

$$\mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u$$

where $\mathbf{x}_l$ and $\mathbf{x}_u$ are the lower and upper bounds, respectively, $C_{l_{ref}}$ is a reference lift coefficient, and A_{ref} is the cross-sectional area of the baseline airfoil nondimensionalized with the square of the chord length.

The design variable vector usually consists of the shape parameterization variables and the angle of attack of the free-stream to the airfoil chordline. The

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_33

shape of the airfoil can be parameterized using various methods such as free form deformation (FFD) [27], class-shape transformation (CST) [16], PARSEC [30], and hicks-henne bump function [6]. In this work, the CST parametrization method is used with a total of twelve variables, six for the upper surface and six for the lower surface.

The next two subsections describe the algorithms to solve the ASO problem, specifically, the GBO algorithm with adjoints, and proposed cEGONN algorithm.

2.2 Constrained GBO algorithm with adjoints

The extended design structure matrix (XDSM) [17] diagram shown in figure 1 outlines the GBO with adjoints algorithm implemented in this work. There are five major modules in the process which are arranged in the diagonal of the matrix. The input to the process (topmost row) are airfoil coordinates, volume mesh, and starting point, and output (leftmost column) is the optimized design. The coordinate file consists of points describing shape of the baseline airfoil in selig format. A structured surface mesh is created using the coordinate file and pyHyp [28] is used for creating an o-mesh grid by extruding the generated surface mesh outwards using hyperbolic mesh marching method.

The first module in the process is an optimizer which drives the entire process. The starting point for optimizer is essentially CST coefficients for baseline airfoil and angle of attack at which lift constraint is satisfied. In this work, sequential quadratic programming is used as optimizer which is implemented in pyOptSparse [33].

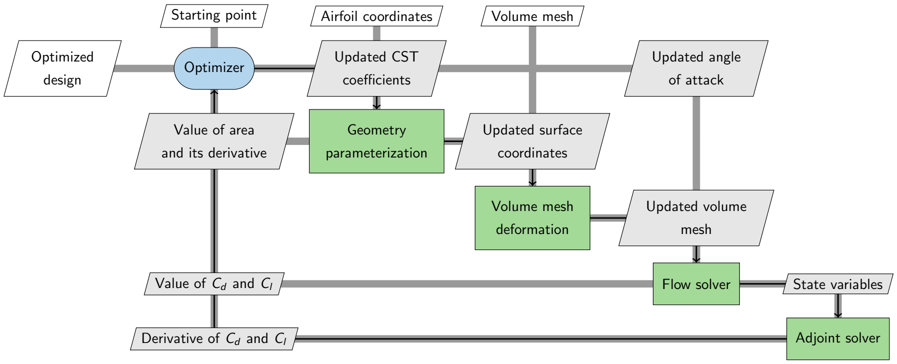


Fig. 1: Extended design structure matrix for gradient-based optimization with adjoints.

The second module consists of pyGeo [11] which provides CST parameterization and is initialized using the airfoil coordinate file. The third module performs mesh deformation using IDWarp [28] and uses volume mesh generated earlier for initialization. The flow solver module consists of ADflow [23] which is a finite-volume structured multiblock mesh solver. In this work, Approximate Newton Krylov method [34] is used to start the sovler and reduce the residual norm to a value of 10^{-6}, relative to initial value, and then full Newton Krylov method is employed to further reduce the residual norm to 10^{-14}, relative to initial norm. Last module consists of a jacobian free discrete adjoint (computed using algorithmic differentiation) method which is implemented within ADflow [12]. The generalized minimal residual method is used for solving the adjoint equations with termination criteria set to a value of 10^{-11}, relative to initial norm.

The process starts with updated design variables from pyOptSparse. pyGeo receives updated CST coefficients and updates the surface coordinates, it also returns the cross-sectional area of updated airfoil shape and its derivative with respect to the variables. The updated surface coordinates are then passed to IDWarp which deforms volume mesh and sends it to flow solver. With the updated angle of attack and mesh, ADflow computes various field variables like pressure, velocity, density, etc. It also computes integral quantities like C_d, C_l, C_m. The converged results are then passed on to adjoint solver which calculates the derivative of objectives and constraints with respect to the design variables. This process is continued until one of the convergence criteria is met within optimizer. Each iteration involves one primal solution (which gives C_d, C_l, C_m) and multiple adjoint solutions (which give derivatives of C_d, C_l, C_m with respect to the design variables) depending on the number of objectives and constraints.

2.3 Constrained EGONN algorithm

Algorithm 1 describes the proposed cEGONN algorithm for aerodynamic shape optimization. Two different data sets are generated using CFD which contain the objective function values $\mathbf{Y}$ and the constraint function values $\mathbf{G}$ for the sampling plan $\mathbf{X}$. In the first step, a neural network (NN_y) learns the mapping between $\mathbf{X}$ and $\mathbf{Y}$ in the first data set. Then, NN_y is used to get prediction $\widehat{\mathbf{Y}}$ and $\widehat{\mathbf{Y}}_u$ at $\mathbf{X}$ and $\mathbf{X_u}$, respectively. Following that the squared prediction error for both the predictions are computed as

$$\mathbf{S} = \sqrt{(\mathbf{Y} - \widehat{\mathbf{Y}})^2}, \tag{2}$$

and

$$\mathbf{S}_u = \sqrt{(\mathbf{Y}_u - \widehat{\mathbf{Y}}_u)^2}. \tag{3}$$

The samples, and the values of the prediction errors and constraints are combined to get a larger data set. A second neural network (NN_u) learns the mapping between the combined input $\widetilde{\mathbf{X}}$ and the prediction error $\widetilde{\mathbf{S}}$. A third neural network NN_g learns the mapping between the combined input $\widetilde{\mathbf{X}}$ and the combined constraint values $\widetilde{\mathbf{G}}$. Once the NN models are trained, the expected improvement, computed by

$$EI(\mathbf{x}) = \begin{cases} [y(\mathbf{x}^*) - \hat{y}(\mathbf{x})]\Phi(Z) + s(\mathbf{x})\phi(Z) & \text{if } s(\mathbf{x}) > 0 \\ 0 & \text{if } s(\mathbf{x}) = 0, \end{cases} \tag{4}$$

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_33

Algorithm 1 Constrained EGONN

```
Require: initial data sets $(\mathbf{X}, \mathbf{Y}, \mathbf{G})$ and $(\mathbf{X}, \mathbf{Y}, \mathbf{G})_u$
  repeat
    fit $NN_y$ to data $(\mathbf{X}, \mathbf{Y})$
    use $NN_y$ to get $\widehat{\mathbf{Y}}$ at $\mathbf{X}$ and $\widehat{\mathbf{Y}}_u$ at $\mathbf{X}_u$
    compute prediction error: $\mathbf{S} \leftarrow \sqrt{(\mathbf{Y} - \widehat{\mathbf{Y}})^2}$ and $\mathbf{S}_u \leftarrow \sqrt{(\mathbf{Y}_u - \widehat{\mathbf{Y}}_u)^2}$
    combine data: $\widetilde{\mathbf{X}} \leftarrow \mathbf{X} \cup \mathbf{X_u}$, $\widetilde{\mathbf{S}} \leftarrow \mathbf{S} \cup \mathbf{S_u}$ and $\widetilde{\mathbf{G}} \leftarrow \mathbf{G} \cup \mathbf{G_u}$
    fit $NN_u$ to data $(\widetilde{\mathbf{X}}, \widetilde{\mathbf{S}})$
    fit $NN_g$ to data $(\widetilde{\mathbf{X}}, \widetilde{\mathbf{G}})$
    $\mathbf{P} \leftarrow \arg\max EI(\mathbf{x})$ such that $\widehat{\mathbf{G}}(x) \leq 0$
    $\mathbf{X} \leftarrow \mathbf{X} \cup \mathbf{P}$
    get observations $Y$ and $G$
    $\mathbf{Y} \leftarrow \mathbf{Y} \cup Y$ and $\mathbf{G} \leftarrow \mathbf{G} \cup G$
  until convergence
  $y^* \leftarrow \min(\mathbf{Y})$ such that constraints are satisfied
  $\mathbf{x}^* \leftarrow \arg\min(\mathbf{Y})$
  return $(\mathbf{x}^*, y^*)$
```

is maximized to get a new infill point $\mathbf{P}$, where $\hat{y}(\mathbf{x})$ is the NN_y prediction and $s(\mathbf{x})$ is NN_u prediction. The Z is a standard normal variable, and Φ and ϕ are cumulative distribution function and probability density function of standard normal distribution, respectively. CFD analysis is performed at the new point $\mathbf{P}$ to obtain the objective function Y and constraint functions G, which are then appended to the first data set. This process is continued until the convergence criteria is met. Unlike GBO, there is no need for adjoint solutions. Compared to the original EGONN, two data sets are utilized in a more efficient manner in the process. Moreover, the algorithm is adapted to handle constrained optimization problems instead of an unconstrained one. The neural networks in cEGONN are implemented within Tensorflow [1].

The next section gives the results of applying GBO algorithm with adjoints and the proposed cEGONN algorithm to the ASO of an airfoil in transonic flow.

3 Results

The general ASO problem formulation is given in (1). In this work, the baseline airfoil is the RAE 2822 (shown in Fig. 2) with the nondimensional reference cross-sectional area $A_{ref} = 0.777$. The free-stream Mach number is fixed at 0.734 and the Reynolds number at 6.5×10^6. The reference lift coefficient is set to $C_{l_{ref}} = 0.824$.

The airfoil shape is parameterized using CST [16] with a total of twelve coefficients, six for the upper surface and six for the lower surface. Thus, the design variable vector $\mathbf{x}$ consists of 13 elements in total, the angle of attack and

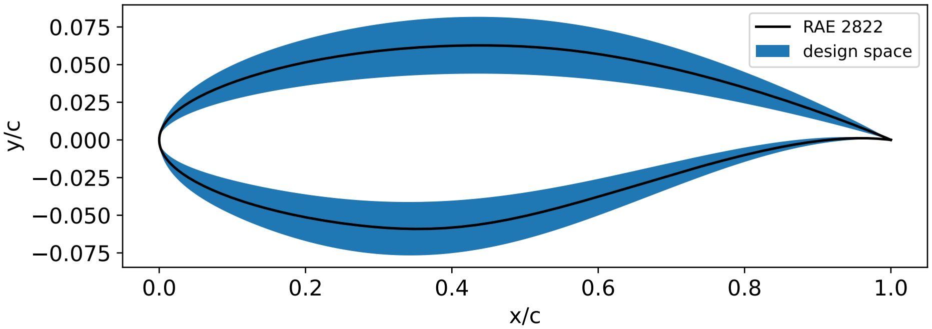
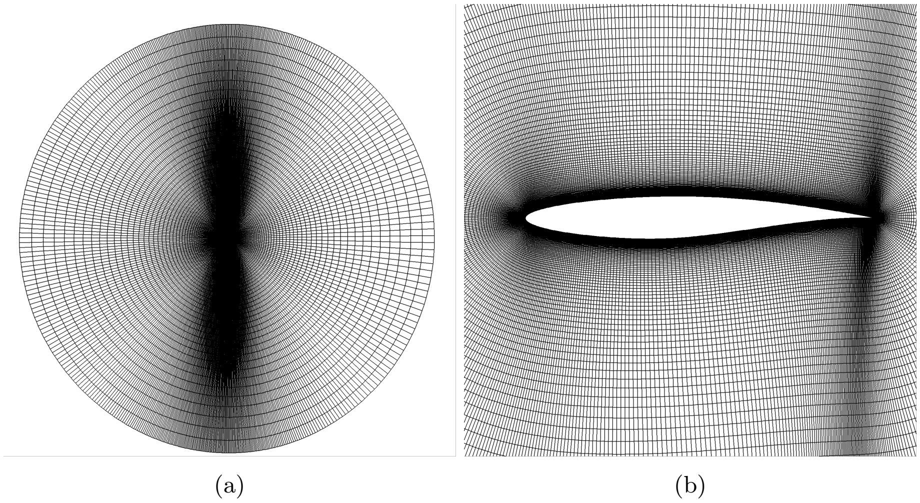


Fig. 2: The baseline RAE 2822 airfoil shape and the design space.

twelve shape variables. The upper and lower bounds for shape variables are set to +30% and -30% perturbations, respectively, of the fitted CST coefficients for the baseline RAE 2822 airfoil, and the angle of attack is bounded between 1.5° and 3.5°. Figure 2 shows the design space obtained with these bounds on the shape variables.

An O-mesh grid around the airfoil is created using pyHyp. Table 1 shows the result of grid convergence study for the mesh. The y^+ plus value for all the levels is less than 1 and the mesh is extruded until 100c. In this work, all the computations are performed using the L1 mesh. Figure 3 shows the generated far-field and surface L1 mesh.


(a)

(b)

Fig. 3: Computational mesh for the airfoil flow simulation: (a) the far-field, and (b) a zoom in near the airfoil surface

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_33

Table 1: Grid convergence study for the RAE 2822 at $M_\infty = 0.734$, $Re = 6.5 \times 10^6$, and $C_l = 0.824$.

Level	Number of cells	C_d (d.c.)	$C_{m,c/4}$	α(degree)
L0	512,000	195.58	-0.096	2.828
L1	128,000	200.55	-0.094	2.891
L2	32,000	213.26	-0.091	3.043
L3	8,000	235.05	-0.086	3.278

The initial data sets required for the cEGONN algorithm are generated using a Latin hypercube sampling plan [24] with the bounds described earlier. The first data set contains 50 samples and second data set contains 25 samples, hence, a total of 75 CFD simulations are performed initially. The convergence criteria is set to a maximum of 100 iterations. The NN_y consists of two hidden layers having eight and six neurons, respectively. The NN_u contains two hidden layers, each having eight neurons. The total number of neurons in NN_u is slightly higher than NN_y to avoid underfitting since it trains on a larger data set. The NN_g also contains two hidden layers but with four and three neurons, respectively. The area constraint is computed based on the airfoil shape using numerical integration, while NN_g provides values for the C_l constraint. In all the NNs, the hyperbolic tan is used as activation function and the number of epochs is set to 5000. The hyperparameters are tuned in such a manner that all the NNs slightly overfit the data since adding more data during sequential sampling will make the NN fit good. All the NNs are trained using the Adam optimizer [13] with a learning rate of 0.001. For maximizing the expected improvement, differential evolution [31] is used with a population size of 130. The mutation and recombination is set to 0.8 and 0.9, respectively, with a maximum of 100 generations. The constraints are handled explicitly using the strategy described in Lampinen [19].

Table 2 summarizes the result of optimization for GBO and cEGONN. Figure 4 shows the convergence history for GBO where a total of 37 iterations were needed for convergence. At each iteration, 1 primal solution and 2 adjoint solutions are computed which totals to 111 solutions.

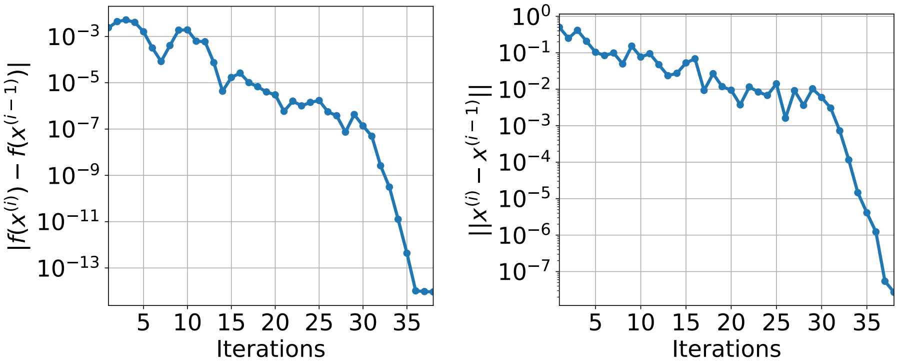
Figure 5 shows the convergence history for cEGONN. A green dot shows a feasible sample, a red dot shows an infeasible sample, and the gray region denotes the initial samples. The line shows variation of C_d with respect to iterations.

Figure 6 shows the baseline and the optimized airfoil shapes. In both the results, optimized shape has less curvature on the top which reduces the flow speed on upper surface. This decreases the strength of the shock which can be clearly noted in the pressure plot shown in Fig. 7. Due to the decrease in

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_33

Table 2: Characteristics of the baseline and optimized airfoil shapes and the number of CFD simulations for each algorithm.

	RAE 2822	GBO	cEGONN
Primal	-	37	175
Adjoint	-	74	0
C_d (d.c.)	200.55	110	114
C_l	0.824	0.824	0.830
α	2.89°	2.83°	2.80°
A	0.777	0.777	0.779



(a)

(b)

Fig. 4: Convergence history for gradient based optimization showing the change between iterations of: (a) the objective function, and (b) the design variable vector.

curvature, the lift generated also decreases which is compensated by an increase in the aft curvature of the lower surface.

In Fig. 5 it is observed that in the early stages of sequential sampling, many infeasible samples are added, but as the iteration progresses the number of feasible samples increase. This is attributed to the fact that the number of samples in the initial data set is low which leads to inaccurate constraint predictions. As more samples are added, the constraint fitting improves and the number of feasible infills increase. Figure 8 shows the Mach contours for the baseline and the optimized shapes at the given free-stream conditions. In the baseline contour Fig. 8a, it can seen that there is a strong shock on the upper surface, whereas in the contour plots of optimized shapes, Figs. 8b and 8c, shock is not present. This shows the capability of the proposed cEGONN for ASO.

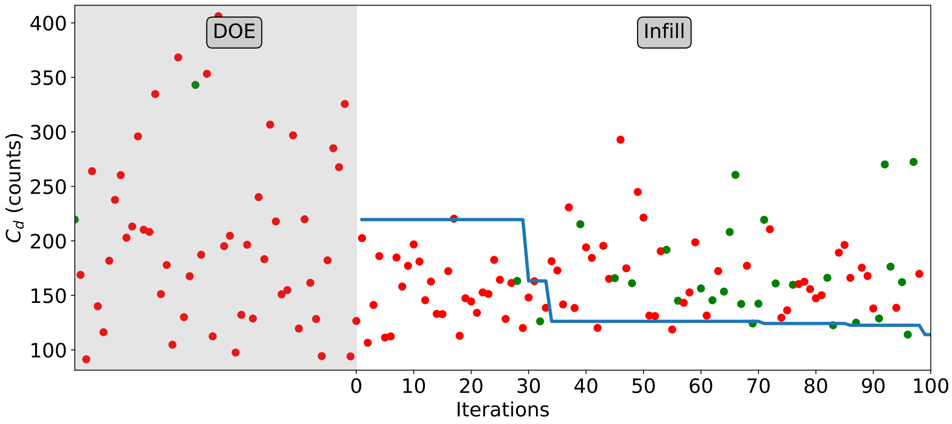


Fig. 5: Optimization convergence history for cEGONN showing the sampling and the drag coefficient values variation with the iterations.

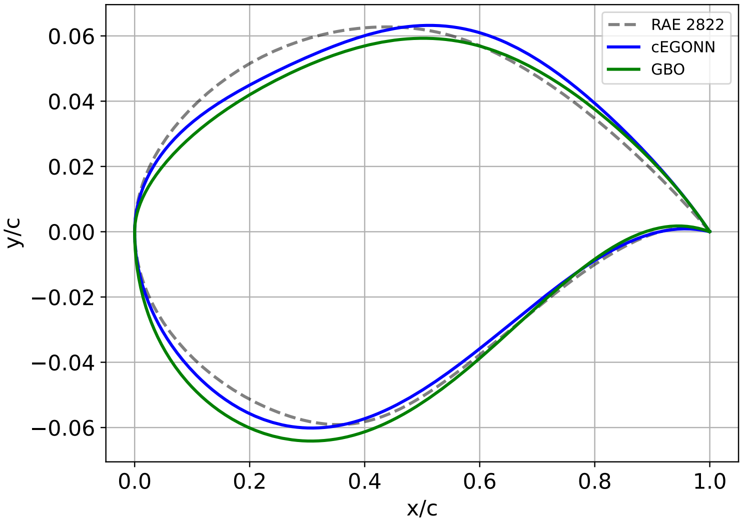


Fig. 6: The baseline and optimized airfoil shapes.

4 Conclusion

In this work, a novel extension to the efficient global optimization with neural network (NN)-based prediction and uncertainty (EGONN) algorithm is proposed which enables the handling of nonlinear constraints. A unique feature of the proposed constrained EGONN algorithm is its ability to perform sequential sampling of the design space and updating the NN predictions of nonlinear objective and constraint functions.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_33

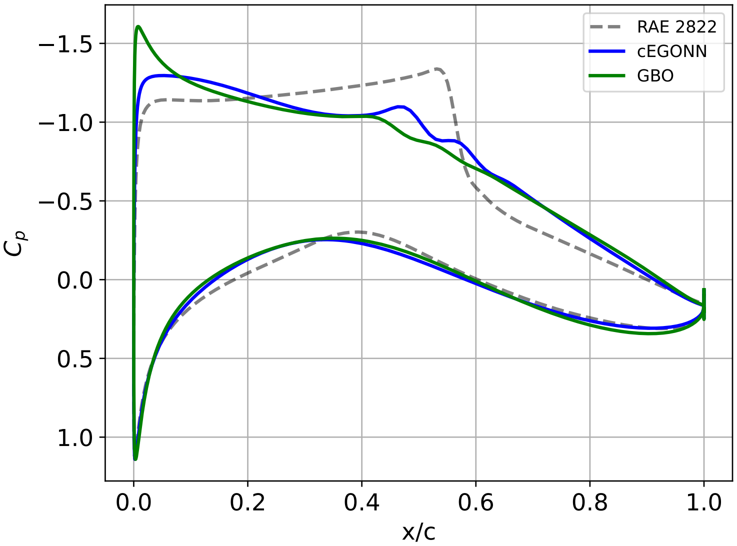


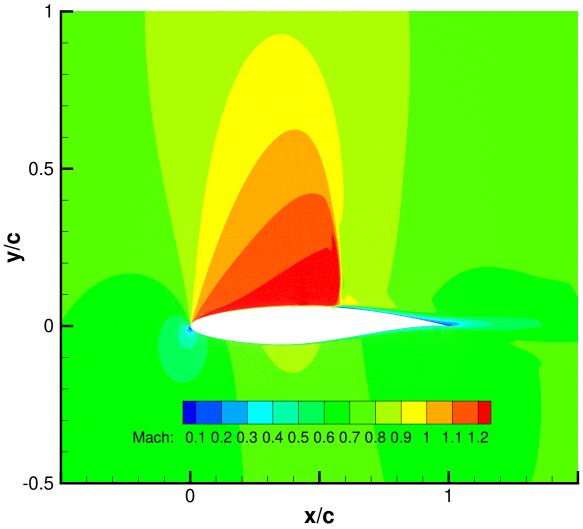
Fig. 7: Coefficient of pressure for the baseline and optimized airfoils.

A demonstration example involving airfoil shape optimization in transonic flow with one objective, two constraints, and twelve design variables shows that the proposed algorithm can obtain comparable optimal designs as gradient-based optimization with adjoints with similar computational cost.

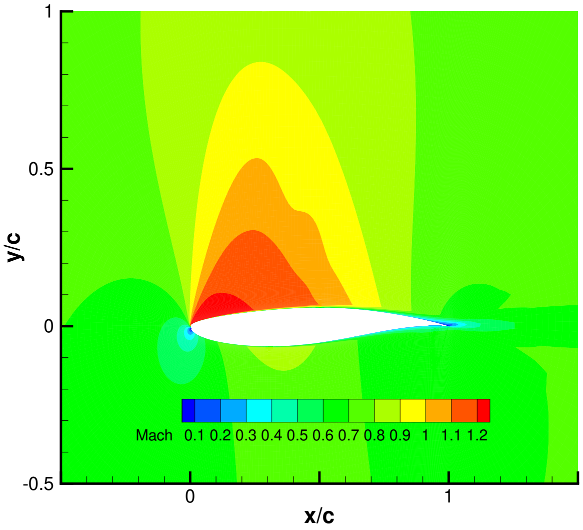
The future steps in this work include extending the algorithm to automatically tune the NNs when additional samples are being gathered. This will be important because the optimal hyperparameters of the NN architecture may change as the number of training points increases. Another important future step is demonstrate and characterize the proposed algorithm on aerodynamic shape optimization problems with a large number of design variables, and a large number of constraints.

References

1. Abadi, M., et al.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), `https://tensorflow.org/`, software available from tensorflow.org
2. Forrester, A.I.J., Keane, A.J.: Recent Advances in Surrogate-Based Optimization. Progress in Aerospace Sciences **45**(1-3), 50–79 (2009)
3. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: Proceedings of the 33rd International Conference on Machine Learning. pp. 1050–1059 (2016)
4. Goan, E., Fookes, C.: Bayesian neural networks: An introduction and survey. In: Mengersen, K., Pudlo, P., Robert, C. (eds) Case Studies in Applied Bayesian Data Science. Lecture Notes in Mathematics. pp. 45–87 (2020)

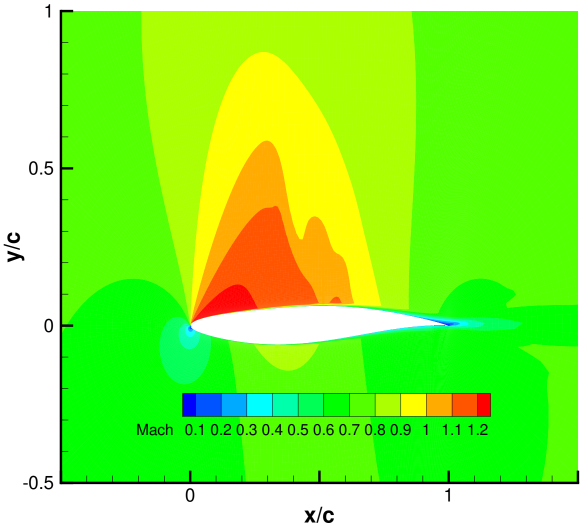


(a)



(b)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36024-4_33



(c)

Fig. 8: Flow field Mach contours for (a) the baseline, (b) the GBO optimum, and (c) the cEGONN optimum.

5. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. The MIT Press, Cambridge, MA (2016)
6. Hicks, R.M., Henne, P.A.: Wing design by numerical optimization. Journal of Aircraft **15**(7), 407–412 (1978)
7. Jameson, A.: Aerodynamic design via control theory. Journal of Scientific Computing **3**, 233–260 (1988). https://doi.org/10.1.1.419.9280
8. Jameson, A., Leoviriyakit, K., Shankaran, S.: Multi-point aero-structural optimization of wings including planform variations. In: 45th Aerospace Sciences Meeting and Exhibit (Reno, Nevada, January 8–11, 2007). https://doi.org/10.2514/6.2007-764
9. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient Global Optimization of Expensive Black-Box Functions. Journal of Global Optimization **13**(4), 455–492 (1998)
10. Kenway, G., Martins, J.R.R.A.: Aerostructural shape optimization of wind turbine blades considering site-specific winds. In: 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (Victoria, British Columbia, Canada, September 10-12, 2008). https://doi.org/10.2514/6.2008-6025
11. Kenway, G., Kennedy, G., Martins, J.R.: A CAD-free approach to high-fidelity aerostructural optimization. In: 13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (2010)
12. Kenway, G.K., Mader, C.A., He, P., Martins, J.R.: Effective adjoint approaches for computational fluid dynamics. Progress in Aerospace Sciences **110**, 100542 (2019)
13. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv:1412.6980 (2014)

14. Koratikere, P., Leifsson, L.T., Barnet, L., Bryden, K.: Efficient global optimization algorithm using neural network-based prediction and uncertainty. In: AIAA SCITECH 2023 Forum. p. 2683 (2023)
15. Koziel, S., Echeverria-Ciaurri, D., Leifsson, L.: "Surrogate-Based Methods" in S. Koziel and X. S. Yang (Eds.) Computational Optimization, Methods and Algorithms, Series: Studies in Computational Intelligence. Springer-Verlag, Berlin Heidelberg (2011)
16. Kulfan, B.M.: Universal parametric geometry representation method. Journal of aircraft **45**(1), 142–158 (2008)
17. Lambe, A.B., Martins, J.R.R.A.: Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes. Structural and Multidisciplinary Optimization **46**, 273–284 (2012)
18. Lampinen, J., Vehtari, A.: Bayesian approach for neural networks - review and case studies. Neural Networks **14**(3), 257–274 (2001)
19. Lampinen, J.: A constraint handling approach for the differential evolution algorithm. In: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600). vol. 2, pp. 1468–1473. IEEE (2002)
20. Lim, Y.F., Ng, C.K., Vaitesswar, U.S., Hippalgaonkar, K.: Extrapolative Bayesian Optimization with Gaussain Process and Neural Network Ensemble Surrogate Models. Advanced Intelligent Sytems **3**, 2100101 (2021)
21. Lyu, Z., Kenway, G.K.W., Martins, J.R.R.A.: Aerodynamic shape optimization investigations of the common research model wing benchmark. AIAA Journal **53**(4), 968–984 (2015). https://doi.org/10.2514/1.J053318
22. Mader, C.A., Martins, J.R.R.A.: Derivatives for time-spectral computational fluid dynamics using an automatic differentiation adjoint. AIAA Journal **50**(12), 2809–2819 (2012). https://doi.org/10.2514/1.J051658
23. Mader, C.A., Kenway, G.K., Yildirim, A., Martins, J.R.: ADflow: An open-source computational fluid dynamics solver for aerodynamic and multidisciplinary optimization. Journal of Aerospace Information Systems **17**(9), 508–527 (2020)
24. McKay, M.D., Beckman, R.J., Conover, W.J.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics **21**(2), 239–245 (1979)
25. Mousavi, A., Nadarajah, S.: Heat transfer optimization of gas turbine blades using an adjoint approach. In: 13th AIAA/ISSMO Multidisciplinary Analysis and Optimizaiton Conference (Fort Worth, Texas, Sep 13-15, 2010). https://doi.org/10.2514/6.2010-9048
26. Queipo, N.V., Haftka, R.T., Shyy, W., Goel, T., Vaidyanathan, R., Tucker, P.K.: Surrogate-based analysis and optimization. Progress in Aerospace Sciences **21**(1), 1–28 (2005)
27. Samareh, J.: Aerodynamic shape optimization based on free-form deformation. In: 10th AIAA/ISSMO multidisciplinary analysis and optimization conference. p. 4630 (2004)
28. Secco, N.R., Kenway, G.K., He, P., Mader, C., Martins, J.R.: Efficient mesh generation and deformation for aerodynamic shape optimization. AIAA Journal **59**(4), 1151–1168 (2021)
29. Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M.A., Adams, R.P.: Scalable bayesian optimization using deep neural networks. In: Proceedings of the 32nd International Conference on Machine Learning. pp. 2171–2180 (2015)
30. Sobieczky, H.: Parametric airfoils and wings. Recent development of aerodynamic design methodologies: inverse design and optimization pp. 71–87 (1999)

31. Storn, R., Price, K.: Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization **11**, 341–359 (1997)
32. Titterington, D.M.: Bayeisan methods for neural networks and related models. Statistical Science **19**(1), 128–139 (2004)
33. Wu, N., Kenway, G., Mader, C.A., Jasa, J., Martins, J.R.: pyoptsparse: A python framework for large-scale constrained nonlinear optimization of sparse systems. Journal of Open Source Software **5**(54), 2564 (2020)
34. Yildirim, A., Kenway, G.K., Mader, C.A., Martins, J.R.: A jacobian-free approximate newton–krylov startup strategy for rans simulations. Journal of Computational Physics **397**, 108741 (2019)