The 9th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2024
3–7 June 2024, Lisboa, Portugal

ONE-SHOT OPTIMIZATION FOR THE INVERSE DESIGN OF A QUASI-1D DE LAVAL NOZZLE

LUCA ZAMPINI[1], VASSILIS GEORGOPOULOS[1] , GRÉGORY COUSSEMENT[2] AND TOM VERSTRAETE[1]

[1] Von Karman Institute for Fluid Dynamics (VKI)
Waterloosesteenweg 72, 1640 Sint-Genesius-Rode, Belgium
email: luca.zampini@vki.ac.be
email: v.georgopoulos@tudelft.nl
email: tom.verstraete@vki.ac.be

[2] Université de Mons
Rue du Joncquois, 53 Mons Belgium
email: gregory.coussement@umons.ac.be

Key words: Inverse Problems, Gradient-Based Optimization, One-Shot Acceleration

Summary. Modern component design often takes place in virtual environments, which are essential for both modeling and optimizing performance. Optimization involves solving complex Partial Differential Equations (PDEs), and the computational cost can be prohibitive, especially with many design variables. Gradient-based methods are preferred for their convergence efficiency, and the adjoint method is commonly used to reduce gradient evaluation costs. However, adjoint-based optimization can still be expensive. To address this, we propose a one-shot acceleration technique that solves PDEs and the optimization problem simultaneously in a coupled iteration. Using a Newton method for the optimality conditions, the inverse design of a nozzle has been performed and the solution has been achieved in just 6 Newton iterations.

1 INTRODUCTION

In recent decades, numerical simulations and Computer-Aided Design (CAD) software have become important to design turbomachinery components. Modeling components in a virtual environment allows for risk reduction and cost-effective testing of various designs through simulations. These simulations are usually performed by solving Partial Differential Equations (PDEs). Computational Fluid Dynamics (CFD) offers advanced tools [1, 2] that accurately model complex fluid phenomena, determining state variables u (e.g., velocity, density, temperature) for given design variables α.

In a virtual environment, it is possible not only to model and simulate the component but also to optimize its performance. During the optimization, the design variables α are adjusted to maximize or minimize an objective function $J(u, \alpha)$. The problem is typically formulated as follows:

$$\min_{u,\alpha} J(u, \alpha) \quad \text{s.t.} \quad R(u, \alpha) = 0 \tag{1}$$

where $R(u, \alpha)$ represents the PDE constraints. This problem can be applied to shape optimization, inverse design, or optimal control. For example, during the aerodynamic shape optimization, a usual goal may be to find an airfoil shape that minimizes drag while satisfying the Navier-Stokes equations.

Among the various optimization methods, gradient-based methods are often chosen for problems with many design parameters [3, 4]. These methods iteratively update the design based on the sensitivity of the objective function. Using gradient information they converge quickly to a local optimum [5]. However, computing gradients can become expensive for high-dimensional design spaces. To reduce the computational cost of the gradient calculation, adjoint techniques [6, 7] have been developed. They allow the calculation of the derivatives with costs nearly independent of the number of design variables.

1.1 Adjoint-based optimization

During shape optimization, the optimal geometry is identified by adjusting the design variables to maximize or minimize a performance index. The system state u is assumed to be a function of the design variables α, implying that each design variable configuration uniquely determines a state that satisfies the governing equations $R(u(\alpha), \alpha) = 0$.

By forming the Lagrangian function associated with this problem, we obtain:

$$\mathcal{L}(u, \alpha, \lambda) = J(u, \alpha) + \lambda^T R(u, \alpha) \tag{2}$$

Here, λ represents the Lagrange multipliers, also called adjoint variables. A Gradient-based optimizer aims at finding the optimum of a given function. The optimum can be mathematically formulated by the following conditions, also known as Karush-Kuhn-Tucker (KKT) conditions:

$$\begin{aligned} \partial_u \mathcal{L} &= \partial_u J(u, \alpha) + \lambda^T \partial_u R(u, \alpha) = 0 \\ \partial_\alpha \mathcal{L} &= \partial_\alpha J(u, \alpha) + \lambda^T \partial_\alpha R(u, \alpha) = 0 \\ \partial_\lambda \mathcal{L} &= R(u, \alpha) = 0 \end{aligned} \tag{3}$$

The objective function J depends on u and α. Using the chain rule, the full gradient of J with respect to α is:

$$\frac{dJ}{d\alpha} = \frac{\partial J}{\partial \alpha} + \frac{\partial J}{\partial u}\frac{du}{d\alpha} \tag{4}$$

The calculation of the total derivative $\frac{du}{d\alpha}$ involves solving the state equation for each design parameter. This can become computationally expensive, especially when the number of design variables is high. However, the gradient can also expressed using the so-called adjoint variable λ. The residual equation $R(u, \alpha) = 0$ is linearized, obtaining the term $\frac{du}{d\alpha}$:

$$\frac{du}{d\alpha} = -\left(\frac{\partial R}{\partial u}\right)^{-1} \frac{\partial R}{\partial \alpha} \tag{5}$$

The term $\frac{du}{d\alpha}$ is then substituted in equation 4, obtaining equation 6:

$$\frac{dJ}{d\alpha} = \frac{\partial J}{\partial \alpha} - \frac{\partial J}{\partial u}\left(\frac{\partial R}{\partial u}\right)^{-1} \frac{\partial R}{\partial \alpha} = \frac{\partial J}{\partial \alpha} + \lambda^T \frac{\partial R}{\partial \alpha} \tag{6}$$

This requires the solution of the adjoint system of equations:

$$\frac{\partial J}{\partial u} + \lambda^T \frac{\partial R}{\partial u} = 0 \tag{7}$$

This linear adjoint system, which depends only on the objective function J, is typically solved once for every objective function of the problem. This makes the adjoint method efficient, especially when the design variables outnumber the objective functions. This is normally the case in aerodynamic shape optimization.

2 One-shot methods

Conventional adjoint-based optimization involves a sequential process: first solving the state equations, then the adjoint equations, and finally using the adjoint solution to compute gradients for updating design variables.

In contrast, the one-shot method solves the PDEs and optimization problem simultaneously by iterating on the state, adjoint, and design variables in a single coupled step. With an appropriate preconditioner, this approach converges efficiently to a stationary point [8].

Ta'asan [9, 10] introduced the one-shot method within a multi-grid structure, executing only a few iterations of state and adjoint solvers per optimization cycle. This method has since been adapted for aerodynamic shape optimization [11, 12].

The one-shot method can be applied to any PDE-constrained optimization problem. Walther et al. [13, 14] extended this approach to include additional equality constraints, while Munguía and Alonso [15] generalized it for both equality and inequality constraints. The method can be used with either continuous or discrete adjoint formulations, though this work employs the discrete adjoint [16].

Efforts to develop simultaneous optimization methods, where state and adjoint equations are not fully converged until optimality conditions are met, are reviewed in [17]. These include the all-at-once approach [18, 19], Simultaneous Optimization Approach (SOA) [16], and simultaneous analysis and design (SAND) [20]. SOA and SAND methods form and solve a nonlinear set of equations approximating the KKT matrix using Newton's method. Multi-grid techniques [18] similarly solve large KKT systems as a whole. Biros and Ghattas [21] proposed a PDE-constrained optimization approach using a space Newton solver with a reduced space quasi-Newton SQP preconditioner.

2.1 Objective of the research

While adjoint techniques are highly efficient for optimization, conventional adjoint-based methods can exhibit slow convergence. To address this, we propose a monolithic one-shot approach to accelerate the optimization process. Existing one-shot methods often use Reduced quasi-Newton sequential quadratic programming (SQP) to approximate the KKT system, which can slow the convergence for large-scale problems. Our approach aims to solve the exact KKT system using a full Newton method, leveraging Automatic Differentiation tools for the exact calculation of the KKT matrix.

Due to its simplicity, low computational cost, and industrial relevance, we will use the quasi 1-D de Laval nozzle as a test case. The inverse design of the nozzle will be performed and the convergence of the one-shot solver will be monitored. This test case will validate the proposed

method and serve as a proof of concept for the more general 2D and 3D optimization.

The paper is organized as follows: Section 3 covers the numerical schemes and methods supporting the one-shot approach. Section 4 details the proposed method and its implementation. Results are discussed in Section 5, with conclusions presented in Section 6.

3 PROBLEM DESCRIPTION AND NUMERICAL APPROACH

This section describes the mathematical models and numerical methods used for the inverse design of the nozzle. Figure 1 shows the geometry of the nozzle, as well as its parameterization. The geometry is parametrized with a Bezier curve and controlled by the cloud of control points, represented in red.

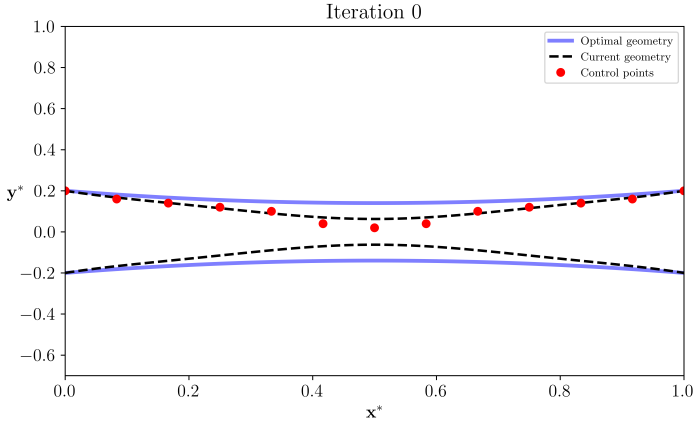


Figure 1: Description of the inverse design problem: optimal geometry (purple), initial geometry (dashed), and geometry parametrization (red dots). Inlet on the left.

3.1 Governing Equations

We consider the quasi-one-dimensional Euler equations for inviscid compressible flows, which describe the conservation of mass, momentum, and energy. In integral form for the region Ω with boundaries $\partial\Omega$, these are expressed as:

$$\frac{\partial}{\partial t}\int_{\Omega} u \,\mathrm{d}\Omega + \int_{\partial\Omega} F \cdot \hat{n} \,\mathrm{d}S = \int_{\Omega} Q \,\mathrm{d}\Omega, \tag{8}$$

where $\hat{n}$ is the outward unit normal vector. The vectors of conservative variables u, convective fluxes F, and source terms Q are:

$$u = \begin{bmatrix} \rho A \\ \rho v_x A \\ \rho E A \end{bmatrix}, F = \begin{bmatrix} \rho v_x A \\ (\rho v_x^2 + p) A \\ \rho v_x H \end{bmatrix}, Q = \begin{bmatrix} 0 \\ P\frac{\mathrm{d}A}{\mathrm{d}x} \\ 0 \end{bmatrix}. \tag{9}$$

For a calorically perfect gas, the pressure P and total enthalpy H are given by:

$$P = (\gamma - 1)\rho\left(E - \frac{v_x^2}{2}\right), \quad H = E + \frac{p}{\rho}. \tag{10}$$

In the quasi-one-dimensional case, the Euler equations are slightly modified to account for variations in the nozzle cross-sectional area. This introduces a source term Q in the momentum equation related to the pressure and axial variation of the cross-sectional area A.

To avoid round-off errors due to differing magnitudes of physical quantities, the flow variables are non-dimensionalized using the inlet total pressure $P_{0,\text{inlet}}$, total temperature $T_{0,\text{inlet}}$, and nozzle length L. The equations are discretized using the method of lines [22]. Space discretization employs a cell-centered finite volume scheme, dividing the domain into quasi-one-dimensional cells (Figure 2).

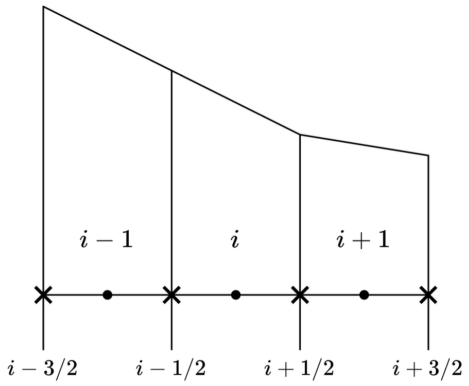


Figure 2: Discretization of the physical domain

For each cell Ω_i, the conservation law 8 is discretized into the semi-discrete form:

$$V_i \frac{\partial u_i}{\partial t} + R_i(u) = 0, \quad \forall i \in \{1, \ldots, N_{\text{cells}}\}, \tag{11}$$

where $R_i(u)$ includes flux balances and source terms and V_i is the volume of each computational cell. The residual R_i for cell Ω_i is given by summing the advective fluxes F and the volumetric source term:

$$R_i = F_{i+1/2} A_{i+1/2} + F_{i-1/2} A_{i-1/2} - Q_i V_i. \tag{12}$$

The numerical flux function F is calculated with the Roe scheme [23], which is chosen for its robustness. The Roe flux is given by:

$$F_{i+1/2} = \frac{1}{2}\left[(F(u_R) + F(u_L)) \cdot \hat{n} - \left| \tilde{\mathbf{A}}_{\text{Roe}} \right| (u_R - u_L) \right], \tag{13}$$

where u_R and u_L are the left and right states neighboring the face and $\left| \tilde{\mathbf{A}}_{\text{Roe}} \right|$ the matrix of the Roe state eigenvalues. Harten's entropy correction is also [24] is used to avoid the carbuncle phenomenon [25].

The volumetric source term Q is discretized by multiplying the local pressure P with the area derivative $\frac{dA}{dx}$. The area derivative is calculated analytically from the Bezier function that describes the geometry.

3.2 Boundary Conditions

The Euler equations are discretized on a computational grid that represents only the internal physical domain. The physical domain is represented in Figure 3.

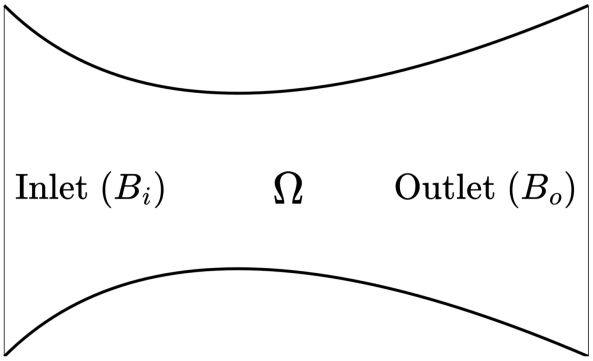


Figure 3: Geometrical representation of the Nozzle.

For the given problem, boundary conditions at the inlet and outlet are specified using the characteristic variables [26]. For the nozzle, the considered boundary conditions are:

- Inlet Boundary: Assumed to be connected to a stagnation chamber with zero velocity, the flow is always assumed subsonic. Total pressure P_{01} and total temperature T_{01} are specified at the inlet.
- Outlet Boundary: Both subsonic and supersonic outflows are considered. For subsonic flow, only the static back pressure P_2 is imposed. For supersonic flow, no requirement on the boundary is necessary.

The ghost cells approach is used, introducing a fictitious layer of dummy cells at the boundaries of the physical domain, as shown in Figure 4.

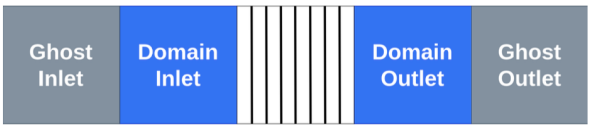


Figure 4: Ghost cells layer at each boundary of the computational domain.

Boundary conditions and the flow solution within the domain define the state $\mathbf{V}_B = [\rho_B, u_B, P_B]^T$ on the boundary, and the Ghost cell values are computed by linear extrapolation.

3.3 Temporal Discretization

Both explicit and implicit time integration schemes are considered for temporal discretization. The implicit time-marching method is derived from the semi-discrete Euler equations (Equation 11) using the backward Euler method:

$$\left(\frac{V}{\Delta t} + \left.\frac{\partial R}{\partial u}\right|_n\right) \Delta u_n = -R(u_n) \tag{14}$$

The main advantage of the implicit method over the explicit one is its ability to use larger time steps, which significantly speeds up the convergence for steady-state problems. As $\Delta t \to \infty$, Equation 14 approximates a Newton method, which convergences quadratically when the initialization is sufficiently close to the solution and the considered residual sensitivity is exact. The local time-stepping technique is also employed to improve the convergence rate of the solution.

4 ONE-SHOT METHOD

The one-shot method integrates the KKT equations 3, which identify the stationary points of the Lagrangian function $\mathcal{L}$.

Conventional adjoint-based optimizers solve these equations sequentially: first converging the state equation, then the adjoint equation, and finally updating the design variables iteratively. In contrast, the proposed one-shot method solves these equations in a coupled, monolithic process with a Newton method. The application of a Newton method to the KKT condition of Equation 3 leads to the system of equations of Equation 15.

$$\begin{bmatrix} \mathcal{L}_{uu} & \mathcal{L}_{u\alpha} & R_u^T \\ \mathcal{L}_{\alpha u} & \mathcal{L}_{\alpha\alpha} & R_\alpha^T \\ R_u & R_\alpha & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta \lambda \end{bmatrix} = - \begin{bmatrix} J_u + \lambda^T R_u \\ J_\alpha + \lambda^T R_\alpha^T \\ R \end{bmatrix} \tag{15}$$

The Right-Hand-Side of Equation 15 is the KKT condition of Equation 3. The subscript indicates the differentiation operation, used here to shorten the notation. This term is the residual of the one-shot problem and can also be referred to as the One-Shot Residual, R_{OS}. The set of variables u, α, and λ are also grouped in a unique vector of one-shot variables $u_{OS} = [u, \alpha, \lambda]^T$.

The Left-Hand-Side matrix includes the second derivatives of the Lagrangian function with respect to the variables u_{OS}. The appearance of these second derivatives is a natural consequence of the Newton method. While the KKT condition is indeed a first-derivative condition, the application of the Newton method introduces the derivatives of the objective function. As the first derivatives are differentiated further, the second derivatives naturally appear in the Newton matrix.

Solving this system provides the Newton step, updating the state, design, and adjoint variables all at once. In this work, we adopt a strongly coupled approach that includes all KKT matrix terms. This full Newton method computes all second-order terms and geometry sensitivities and the key for the full differentiation required by the Newton Method is a mix of analytical computation and Automatic Differentiation.

The calculation is divided into two separate steps, highlighted in equation 15. First, the Right-Hand-Side vector, also known as the One-Shot Residual, is calculated analytically. Then, the Left-Hand-Side matrix, also known as the KKT Matrix, is calculated with Automatic Differentiation tools. In Section 4.1 and 4.2 the calculation of these two components is described in more detail.

4.1 One-shot Residual

The one-shot residual is the collection of the adjoint residual, the gradient residual, and the flow residual, as reported in Equation 15. The calculation of the one-shot residual goes through the following calculations: flow residual, adjoint residual, and gradient residual. The calculation of the flow residual R is straightforward, as it just requires the evaluation of the residual using the solver described in section 3.

The calculation of the adjoint residual requires the evaluation of the term $\frac{\partial R}{\partial u}$. The derivation of the Roe flux is performed analytically and tested against Automatic Differentiation tools. The result of the derivation is the sensitivity of the 3 fluxes with respect to the primitive flow variables (density, velocity, and pressure) of Equation 16. The flux derivatives with respect to the primitive variables v are then transformed into derivatives with respect to the conservative variables u (density, momentum, and energy) by means of the chain rule:

$$\frac{\partial F}{\partial v} = \begin{bmatrix} \frac{\partial F_1}{\partial \rho} & \frac{\partial F_1}{\partial v_x} & \frac{\partial F_1}{\partial P} \\ \frac{\partial F_2}{\partial \rho} & \frac{\partial F_2}{\partial v_x} & \frac{\partial F_2}{\partial P} \\ \frac{\partial F_3}{\partial \rho} & \frac{\partial F_3}{\partial v_x} & \frac{\partial F_3}{\partial P} \end{bmatrix}, \quad \frac{\partial F}{\partial u} = \frac{\partial F}{\partial v}\frac{\partial v}{\partial u} \tag{16}$$

where F_1, F_2, and F_3 denote the mass, momentum, and energy fluxes, respectively, and ρ v_x and P denote the density, axial velocity, and pressure.

The derivation of the source term Q in Equation 9 involves only the pressure term. The derivation of this source term is trivial and its derivation is performed analytically as well.

Summing up all contributions from all faces and all volumes, the full derivative of the flow residual R with respect to the conservative variables u is obtained. This includes boundary conditions as well. Given the fact that the sensitivity of the residual $\frac{\partial R}{\partial u}$ is calculated exactly, the calculation of the adjoint residual is obtained by performing the transposed product between the Jacobian and the adjoint variable and adding to this function sensitivity $\frac{\partial J}{\partial u}$ according to Equation 7.

The calculation of the gradient residual is performed analytically. This involves the propagation of the sensitivities from the control points of the geometry α up until the calculation of the flow residual R. The chain rule used for this operation is reported in Equation 17 and each of the aforementioned steps has been performed by hand and validated against Automatic Differentiation tools.

$$\frac{dR}{d\alpha} = \frac{\partial R}{\partial \alpha} + \frac{\partial R}{\partial A}\frac{\partial A}{\partial B_{ezier}}\frac{\partial B_{ezier}}{\partial \alpha} \tag{17}$$

4.2 KKT Matrix

The differentiation of the one-shot residual is performed using the complex-step method. This method, derived from Taylor series expansion, involves replacing real values in the source code with complex variables [27]. The first derivative can be approximated as:

$$\frac{\partial f}{\partial x} \approx \frac{\mathrm{Im}(f(x+ih))}{h} \tag{18}$$

By adding a small complex perturbation h (e.g., 10^{-12}) and calling the evaluation of the one-shot residual multiple times, it is possible to construct the KKT matrix column-by-column.

The main steps are the following:

1. Introduce a complex perturbation into u, α, or λ.
2. Calculate the one-shot residual R_{OS} using the perturbed variables.
3. Extract the complex derivatives and place the column of derivatives into the KKT matrix.

Perturbing the i-th variable in the one-shot vector u_{OS} allows for the calculation of the derivatives with respect to that specific variable, thereby determining the i-th column of the KKT matrix. This perturbation and column-by-column construction process is repeated for every variable in the vector u_{OS}, ensuring the complete formation of the KKT matrix.

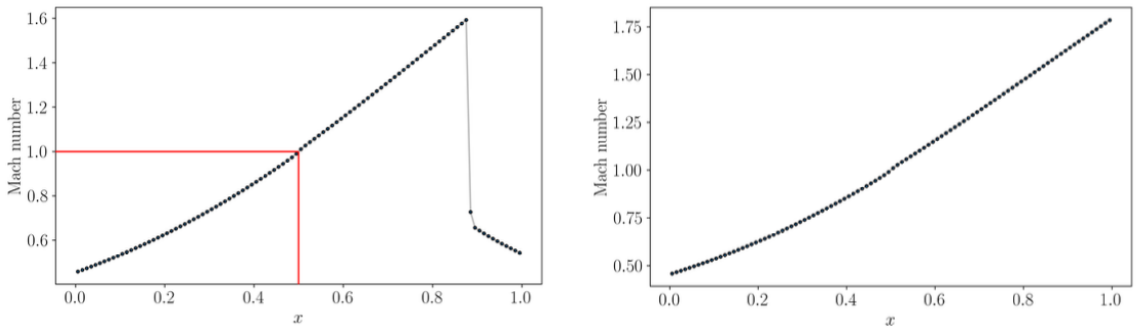
A key advantage of the full analytical calculation of the one-shot residual is that it enables the use of automatic differentiation tools. If even one term were calculated numerically, automatic differentiation would not be possible.

5 RESULTS

To validate the primal solver implementation some test runs were made. Inlet and outlet areas were set equal to 0.2 while the throat area was set equal to 0.14. Then, simulations with different pressure ratios $Pr = P_2/P_{01}$ were carried out.

It is known that for an under-expanded pressure ratio $Pr = 0.729$, sonic conditions are expected at the throat and a shock is expected near the exit of the nozzle. Figure 5a shows the Mach-number distribution for a Pressure Ratio $Pr = 0.729$ and the solver manages to capture quite well the resulting shock that occurs inside the divergent part of the nozzle.

It is also known that for the critical pressure ratio $Pr = Pc = 0.528$, the shock is expected to occur at the exit plane of the nozzle. The computational domain consists of the internal area of the nozzle and hence the solver should not see any shock. Indeed, in Figure 5b no shock is observed within the nozzle and the flow is supersonic at the outlet. These two observations suggest that the implementation of the 1D solver is performed correctly.



(a) (b)

Figure 5: a Underexpanded nozzle, b Adapted nozzle

5.1 One-shot optimizer

After validating the framework, the optimization problem is solved, and the convergence of the solution is monitored. Figure 6 shows the comparison between the initial and target geometries. The dashed line represents the initial geometry, while the solid line represents the target geometry. The final geometry closely matches the target geometry and the L_2 norm of the error is below 10^{-12}. This proves that the optimizer was indeed able to solve the optimization problem.

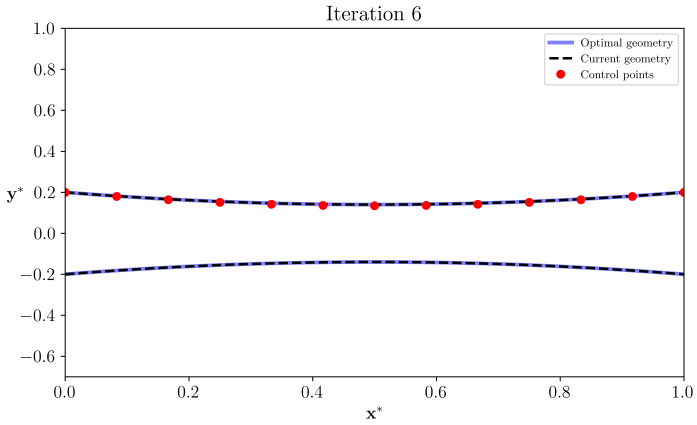


Figure 6: Final configuration after 6 one-shot iterations

The convergence history of the one-shot method is illustrated in Figure 7, where the norms of the CFD, adjoint, and design residuals are plotted against the iteration number. In this context, a single iteration refers to solving the KKT conditions once. This should not be confused with the iteration of the CFD or adjoint solver, as these are implicitly resolved within the one-shot solver. The residuals converge simultaneously to zero and the convergence rate of the residuals is quadratic. This fast convergence rate is the key advantage of the proposed one-shot method. Remarkably, the optimization problem was resolved in just 6 coupled iterations.

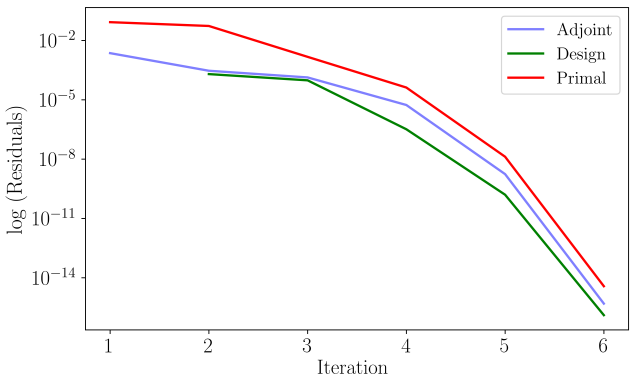


Figure 7: Simultaneous convergence of Flow, Adjoint and Gradient equations

6 CONCLUSIONS

A novel one-shot acceleration technique has been applied to the inverse design of a quasi-1D nozzle, demonstrating remarkable effectiveness in reducing the number of iterations required to achieve the optimal solution.

Initially, a quasi-1D implicit flow solver was developed and validated against established solutions. Subsequently, a coupled one-shot method was formulated for the nozzle geometry. This method leverages a full-Newton approach applied to the optimality conditions, commonly referred to as the one-shot residual. The one-shot residual was derived analytically, enabling its differentiation using automatic tools. The complex-step method was successfully employed to compute the derivatives of the one-shot residual, forming the Newton matrix necessary for the simultaneous update of the flow, adjoint variable, and geometry. During the optimization phase, the method exhibited significant convergence advantages. The final nozzle geometry closely matched the target configuration, indicating the method's effectiveness in minimizing discrepancies in the pressure distribution. The observed quadratic convergence of the residuals further underscores the efficiency of the one-shot method, with the optimum achieved in just 6 Newton iterations.

In summary, the coupled one-shot acceleration technique presents a promising approach to solving optimization problems in computational fluid dynamics, offering both rapid convergence and precise solutions.

References

[1] Norbert Kroll, Stefan Langer, and Axel Schwöppe. "The DLR flow solver TAU - Status and recent algorithmic developments". In: Jan. 2014. ISBN: 978-1-62410-256-1. DOI: 10.2514/6.2014-0080.

[2] E.M. Lee-Rausch et al. "Application of the FUN3D solver to the 4th AIAA drag prediction workshop". In: *Journal of Aircraft* 51 (July 2014), pp. 1149–1160. DOI: 10.2514/1.C032558.

[3] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. 2e. New York, NY, USA: Springer, 2006.

[4] Jacques-Louis Lions. "Optimal Control of Systems Governed by Partial Differential Equations". In: 1971.

[5] Ji-chao Li and Mengqi Zhang. "Data-based approach for wing shape design optimization". In: *Aerospace Science and Technology* 112 (2021).

[6] O. Pironneau. "On optimum profiles in Stokes flow". In: *Journal of Fluid Mechanics* 59.1 (1973). DOI: 10.1017/S002211207300145X.

[7] Antony Jameson. "Aerodynamic design via control theory". In: *Journal of Scientific Computing* 3.3 (Sept. 1988). ISSN: 1573-7691. DOI: 10.1007/BF01061285.

[8] Adel Hamdi and Andreas Griewank. "Properties of an augmented Lagrangian for design optimization". In: *Optimization Methods and Software* 25 (2010).

[9] Shlomo Ta'asan. "One shot methods for optimal control of distributed parameter systems 1: Finite dimensional control". In: 1991.

[10] Shlomo Ta'asan, G. Kuruvila, and M. Salas. "Aerodynamic design and optimization in one shot". In: *30th Aerospace Sciences Meeting and Exhibit*. DOI: 10.2514/6.1992-25.

[11] Nicolas R. Gauger et al. "Automated Extension of Fixed Point PDE Solvers for Optimal Design with Bounded Retardation". In: *Constrained Optimization and Optimal Control for Partial Differential Equations*. 2012.

[12] Angelo Carnarius et al. "Optimal Control of Unsteady Flows Using a Discrete and a Continuous Adjoint Approach". In: *System Modeling and Optimization*. Ed. by Dietmar Hömberg and Fredi Tröltzsch. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. ISBN: 978-3-642-36062-6.

[13] Andrea Walther et al. "On an extension of one-shot methods to incorporate additional constraints". In: *Optimization Methods and Software* 31.3 (2016). DOI: 10.1080/10556788.2016.1146268.

[14] Andrea Walther, Lisa Kusch, and Nicolas Gauger. "New Results for the Handling of Additional Equality Constraints in One-Shot Optimization". In: *Vietnam Journal of Mathematics* 46 (Sept. 2018). DOI: 10.1007/s10013-018-0306-y.

[15] Brian C. Munguía and Juan J. Alonso. "One Shot Optimization with Generalized Constraints". In: *AIAA Scitech 2020 Forum*. DOI: 10.2514/6.2020-0886.

[16] S.B. Hazra et al. "Aerodynamic shape optimization using simultaneous pseudo-timestepping". In: *Journal of Computational Physics* 204.1 (2005). ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2004.10.007.

[17] Torsten Bosse et al. "One-Shot Approaches to Design Optimzation". In: 2014.

[18] Alfio Borzì and Volker Schulz. "Multigrid Methods for PDE Optimization". In: *SIAM Review* 51 (May 2009). DOI: 10.1137/060671590.

[19] J. Carsten Ziems and Stefan Ulbrich. "Adaptive Multilevel Inexact SQP Methods for PDE-Constrained Optimization". In: *SIAM J. Optim.* 21 (2011).

[20] Raphael T. Haftka and Zafer Gürdal. *Elements of structural optimization.* 3. revised and expanded ed. Vol. 1. Solid Mech. Appl. Dordrecht: Kluwer Academic Publishers, 1992. ISBN: 0-7923-1504-9.

[21] George Biros and Omar Ghattas. "Parallel Lagrange–Newton–Krylov–Schur Methods for PDE-Constrained Optimization. Part I: The Krylov–Schur Solver". In: *SIAM Journal on Scientific Computing* 27.2 (2005). DOI: 10.1137/S106482750241565X.

[22] Robert D Richtmyer and Keith W Morton. "Difference methods for initial-value problems". In: *New York: Interscience* (1967).

[23] P.L Roe. "Approximate Riemann solvers, parameter vectors, and difference schemes". In: *Journal of Computational Physics* 43.2 (1981), pp. 357–372. ISSN: 0021-9991. DOI: https://doi.org/10.1016/0021-9991(81)90128-5. URL: https://www.sciencedirect.com/science/article/pii/0021999181901285.

[24] Amiram Harten, Peter Lax, and Bram van Leer. "On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws". In: *SIAM Rev* 25 (Jan. 1983), pp. 35–61.

[25] K. Peery and Scott T. Imlay. "Blunt-body flow simulations". In: 1988.

[26] D. Whitfield and Jonathan Janus. "Three-dimensional unsteady Euler equations solution using flux vector splitting". In: *AIAA Paper* 84 (July 1984).

[27] Joaquim R. R. A. Martins, Peter Sturdza, and Juan J. Alonso. "The Complex-Step Derivative Approximation". In: *ACM Trans. Math. Softw.* 29.3 (Sept. 2003), pp. 245–262. ISSN: 0098-3500. DOI: 10.1145/838250.838251. URL: https://doi.org/10.1145/838250.838251.