

# Adjoint based aerodynamic shape optimisation using kinetic meshfree method

Keshav S Malagi<sup>a,\*</sup>, Nischay R. Mamidi<sup>b</sup>, N Anil<sup>b</sup>, V Ramesh<sup>a</sup>, S M Deshpande<sup>c</sup>

<sup>a</sup>Computational and Theoretical Fluid Dynamics Division, CSIR-NAL, Bengaluru, India

<sup>b</sup>Department of Mathematics, BITS Pilani-Hyderabad Campus, Hyderabad, India

<sup>c</sup>FASc, FNAE, FAeSI Fellow, Bengaluru, India

\*Corresponding author: [keshav@nal.res.in](mailto:keshav@nal.res.in)

## Abstract

The gradient based optimisation algorithms combined with the finite volume or element based adjoint approaches have been very successful in aerodynamic shape optimization (ASO). The meshfree least squares kinetic upwind method (LSKUM), which works on a cloud of points and its connectivity, has the advantage in terms of flexibility of generating a cloud of points and having an arbitrary number of points in the connectivity. The LSKUM based meshfree solvers have been successfully applied to problems involving multi-body configurations and moving boundaries. In the present work, the LSKUM based primal and discrete adjoint meshfree solvers have been applied to perform aerodynamic shape optimisation. It is well-known that the raw shape sensitivities of a defined objective function are highly oscillatory. In this research, we have used the Sobolev gradient smoothing. An approach is developed to find a suitable choice for the smoothing parameter in the Sobolev gradient algorithm. Numerical results are presented for the shape optimisation of the NACA0012 airfoil at transonic and supersonic flows. The advantages of using LSKUM in aerodynamic shape optimisation have been shown.

**Keywords:** aerodynamic shape optimisation; discrete adjoints; gradient smoothing; meshfree methods; LSKUM.

## 1 Introduction

The gradient based optimisation algorithms combined with the adjoint approaches, which can handle large design spaces, have become popular in aerodynamic shape optimisation (ASO) [1; 2; 3; 4]. An optimisation algorithm consists of perturbing the shape in the direction of the gradient of the cost function with respect to the control variables, also known as shape variables. The shape sensitivities are determined by solving the adjoint problem derived from the governing equations of fluid dynamics, whose CFD solution for the given boundary conditions is called the primal solution. The adjoint solvers developed till now use finite volume or finite element formulations. To our knowledge, the meshfree formulation has not been employed in adjoint based aerodynamic shape optimisation.

A meshfree scheme of particular interest to us is the Least Squares Kinetic Upwind Method (LSKUM)

[5; 6]. LSKUM operates on the distribution of points, known as a point cloud. Each point in the cloud requires connectivity or neighbourhood information to obtain the least-squares approximation of the spatial derivatives in the governing equations. LSKUM requires a connectivity preprocessor operating on the cloud of points. There is a lot of flexibility in choosing the connectivity set, as the number of points in the neighbourhood need not be fixed. The only restriction is that the number of neighbours must be at least 3 in two-dimensions (2D) and 4 in three-dimensions (3D). One important difference between the finite volume method (FVM) and meshfree LSKUM is worth emphasising. The FVM requires a disjoint cover of volumes for the computational domain. The LSKUM operates on an overlapping cover of connectivity sets. Points in the connectivity can vary over a domain. This flexibility of LSKUM is expected to be useful in performing shape optimisation. As mentioned before, the number of points in the connectivity is arbitrary. Hence during optimisation cycles, as the shape changes, points may move out or in the connectivity of the body points or nearby points without affecting the least-squares approximation of the spatial derivatives.

Over the past two decades, the LSKUM based meshfree solvers have been successfully used for computing flows around realistic multi-body configurations, problems involving moving bodies, investigating store separation dynamics, strongly rotating viscous flows, and flutter prediction in turbomachines [7; 8; 9; 10; 11]. In this paper, an attempt has been made to explore aerodynamic shape optimisation using meshfree LSKUM adjoints. The adjoint meshfree solver that computes the shape sensitivities is developed by algorithmically differentiating (AD) the primal LSKUM solver. This approach has been successfully employed in developing robust and accurate adjoint solvers in steady and unsteady flows [12; 13; 14].

This paper is organised as follows. Section 2 presents the basic theory of the kinetic meshfree solver based on LSKUM. The mathematical formulation of the optimisation problem and the details related to developing a discrete adjoint meshfree solver for accurate computation of shape sensitivities are presented in Section 3. Sobolev gradient smoothing of non-smooth shape sensitivities is shown in Section 4. Furthermore, an approach to finding the smoothing parameter in the Sobolev gradient algorithm that yields sufficiently smoothed sensitivities is presented. Section 5 presents the optimisation results for the NACA0012 airfoil at transonic and supersonic flows. Finally, conclusions are drawn in Section 6.

## 2 Primal meshfree LSKUM solver

The Least Squares Kinetic Upwind Method (LSKUM) is a kinetic theory based meshfree upwind scheme for the numerical solution of compressible fluid flows [5; 6]. It uses the Kinetic Flux Vector Split (KFVS) fluxes [15]. The KFVS scheme is based on the well-known fact that the governing equations of fluid dynamics can be obtained as the suitable moments of the Boltzmann equation with appropriate velocity distribution functions [16]. In this section, we present the basic theory of LSKUM for two-dimensional (2D) Euler equations.

In 2D, the Boltzmann equation is given by

$$\frac{\partial f}{\partial t} + v_1 \frac{\partial f}{\partial x} + v_2 \frac{\partial f}{\partial y} = J(f, f) \quad (2.1)$$

where  $f$  is the molecular velocity distribution function,  $J(f, f)$  is the collision term,  $v_1$  and  $v_2$  are the molecular velocities along the coordinate directions  $x$  and  $y$ , respectively. When the velocity distribution function,  $f$  is Maxwellian ( $F$ ), the collision term vanishes and  $\Psi$ -moments of the eqn. (2.1) lead to the

Euler equations. In the compact form, the relation between these equations can be expressed as

$$\left\langle \Psi, \frac{\partial f}{\partial t} + v_1 \frac{\partial f}{\partial x} + v_2 \frac{\partial f}{\partial y} \right\rangle = \frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{GX}}{\partial x} + \frac{\partial \mathbf{GY}}{\partial y} = 0, \quad f = F \quad (2.2)$$

where,  $\mathbf{U}$  is the conserved vector,  $\mathbf{GX}$  and  $\mathbf{GY}$  are the flux vectors in  $x$  and  $y$  directions, respectively. These vectors are defined by

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho E \end{bmatrix}, \quad \mathbf{GX} = \begin{bmatrix} \rho u_1 \\ \rho u_1^2 + p \\ \rho u_1 u_2 \\ (\rho E + p)u_1 \end{bmatrix}, \quad \mathbf{GY} = \begin{bmatrix} \rho u_2 \\ \rho u_1 u_2 \\ \rho u_2^2 + p \\ (\rho E + p)u_2 \end{bmatrix} \quad (2.3)$$

where,  $\rho$  is fluid density,  $p$  is pressure,  $E$  is total energy per unit mass,  $u_1$  and  $u_2$  are fluid velocities in  $x$  and  $y$  directions, respectively. The Maxwellian distribution function  $F$  and the moment function vector  $\Psi$  are defined by,

$$F = \frac{\rho}{I_0} \left( \frac{\beta}{\pi} \right) \exp \left[ -\beta \{ (v_1 - u_1)^2 + (v_2 - u_2)^2 \} - \frac{I}{I_0} \right] \quad (2.4)$$

$$\Psi = \left[ 1, v_1, v_2, I + \frac{v_1^2 + v_2^2}{2} \right]^T$$

where,  $I_0$  and  $I$  represent equilibrium internal energy and internal energy due to non-translational degrees of freedom and  $\beta = 1/2RT$ . The moment relations to obtain the conserved and the flux vectors are given as

$$\mathbf{U} = \langle \Psi, F \rangle = \int_{\mathbb{R}^+ \times \mathbb{R}^2} \Psi F dv_1 dv_2 dI \quad (2.5)$$

$$\mathbf{GX} = \langle \Psi, v_1 F \rangle = \int_{\mathbb{R}^+ \times \mathbb{R}^2} \Psi v_1 F dv_1 dv_2 dI, \quad \mathbf{GY} = \langle \Psi, v_2 F \rangle = \int_{\mathbb{R}^+ \times \mathbb{R}^2} \Psi v_2 F dv_1 dv_2 dI$$

To obtain the KFVS fluxes, consider the CIR split [17] Boltzmann equation with the distribution function being the Maxwellian,

$$\frac{\partial f}{\partial t} + v_1^+ \frac{\partial f}{\partial x} + v_1^- \frac{\partial f}{\partial x} + v_2^+ \frac{\partial f}{\partial y} + v_2^- \frac{\partial f}{\partial y} = 0, \quad f = F, \quad v_i^\pm = \frac{v_i \pm |v_i|}{2} \quad (2.6)$$

$\Psi$ -moments of the above equation leads to the KFVS Euler equations

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{GX}^+}{\partial x} + \frac{\partial \mathbf{GX}^-}{\partial x} + \frac{\partial \mathbf{GY}^+}{\partial y} + \frac{\partial \mathbf{GY}^-}{\partial y} = 0 \quad (2.7)$$

where,  $\mathbf{GX}^\pm$  and  $\mathbf{GY}^\pm$  are KFVS Euler fluxes [15]. The expressions for the split fluxes can be obtained by using the moment relations in eqn. (2.5) with suitable limits of integration that respect the corresponding velocity space.

In LSKUM, the flux derivatives in the KFVS Euler equations are discretised using the least-squares principle [6]. To derive the least-squares approximation of the spatial derivatives, consider the Taylor series expansion of  $f$  at a point  $P_i$  around the point  $P_0$ ,

$$f_i = f_0 + f_{x_0} \Delta x_i + f_{y_0} \Delta y_i + O(\Delta x^2, \Delta y^2), \quad i = 1, \dots, n \quad (2.8)$$

where,  $\Delta x_i = x_i - x_0$ ,  $\Delta y_i = y_i - y_0$  and  $n$  represents the number of neighbours of the point  $P_0$ . The set of points in the neighbourhood of  $P_0$  defines its connectivity set (stencil),  $C(P_0) = \{P_i : d(P_i, P_0) < h\}$ . Here,  $d(P_i, P_0)$  is the Euclidean distance between the points  $P_i$  and  $P_0$ , and  $h$  is a characteristic length to be chosen by the user. For  $n \geq 3$ , the above equation represents an over-determined linear system

for the unknowns  $f_{x_0}$  and  $f_{y_0}$ . To find the first-order accurate least-squares approximation of the spatial derivatives, we neglect the terms of order greater than  $O(\Delta x, \Delta y)$  in eqn. (2.8). Consider the sum of the squares of the error in the approximation of  $f_i$ ,

$$E = \sum e_i^2 = \sum (\Delta f_i - f_{x_0} \Delta x_i - f_{y_0} \Delta y_i)^2, \quad \Delta f_i = f_i - f_0 \quad (2.9)$$

Minimising  $E$  with respect to  $f_{x_0}$  and  $f_{y_0}$  gives the system

$$\begin{bmatrix} \sum \Delta x_i^2 & \sum \Delta x_i \Delta y_i \\ \sum \Delta x_i \Delta y_i & \sum \Delta y_i^2 \end{bmatrix} \begin{bmatrix} f_{x_0} \\ f_{y_0} \end{bmatrix} = \begin{bmatrix} \sum \Delta f_i \Delta x_i \\ \sum \Delta f_i \Delta y_i \end{bmatrix} \quad (2.10)$$

The solution of the above linear system yields the first-order accurate approximations to  $f_{x_0}$  and  $f_{y_0}$ . Taking the  $\Psi$ -moments of eqn. (2.6) with the least-squares formulae for  $f_{x_0}$  and  $f_{y_0}$  results in the semi-discrete form of LSKUM. To enforce upwinding, the spatial derivatives of the split fluxes  $\mathbf{GX}^\pm$  and  $\mathbf{GY}^\pm$  are approximated using sub-stencils. For example, the sub-stencil for approximating the spatial derivative of  $\mathbf{GX}^+$  is defined as

$$C_1(P_0) = \{P_i \in C(P_0) | \Delta x_i < 0\} \quad (2.11)$$

This approach is known as upwinding by stencil sub-division [6; 18]. The second-order accurate approximations to the spatial derivatives can be obtained by using the defect correction approach with entropy variables [19]. Suitable time marching scheme leads to the state update formula for LSKUM.

It is pertinent to note that the mathematical formulation of the method does not place any condition on the point cloud. Thus, given a cloud of points and connectivity set (stencil) at each point, we can obtain the spatial derivatives without the need for topological information of the point cloud. However, the connectivity generation for any point should ensure that the least-squares matrix shown in eqn. (2.10) must be well-conditioned. This is crucial in ensuring the accuracy and robustness of least-squares approximation, thereby avoiding code divergence.

### 3 Adjoint meshfree LSKUM solver

Consider the problem of finding an optimal shape that maximises or minimises an objective function of particular interest in aerodynamics. This amounts to a PDE-constrained optimisation problem, which, in its general form, can be defined as

$$\begin{aligned} \max/\min \quad & J(\mathbf{U}, \boldsymbol{\alpha}) \\ \text{subject to} \quad & \mathbf{C}(\mathbf{U}, \boldsymbol{\alpha}) = 0 \end{aligned} \quad (3.12)$$

where  $J$  is a scalar objective function like the aerodynamic lift or drag coefficients and  $\mathbf{U}$  is the conserved vector.  $\boldsymbol{\alpha}$  is the vector of shape coordinates  $(x, y)$  that define the geometry of the configuration in two-dimensions. In the present work,  $\mathbf{C}(\mathbf{U}, \boldsymbol{\alpha}) = 0$  represents the compressible Euler equations with boundary conditions. After discretisation, the constraints can be written as

$$\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} + \mathbf{R}_s(\mathbf{U}, \boldsymbol{\alpha}) = 0 \quad (3.13)$$

Here,  $\mathbf{R}_s(\mathbf{U}, \boldsymbol{\alpha})$  is the discrete residual vector obtained after the meshfree discretisation of the spatial derivatives in eqn. (2.7). Since we are interested in the steady-state solution, the superscript  $n$  represents a pseudo-time iteration. The state-update formula can be written in the fixed point form as

$$\mathbf{U}^{n+1} = G(\mathbf{U}^n, \boldsymbol{\alpha}) \quad (3.14)$$

Here,  $G$  represents an iteration of the meshfree solver based on LSKUM for the numerical solution of the Euler equations. The above fixed point iteration converges to the steady state solution  $\mathbf{U}$ , given by

$$\mathbf{U} = G(\mathbf{U}, \boldsymbol{\alpha}) \quad (3.15)$$

In the discrete form, the optimisation problem defined in eqn. (3.12) can be formulated as

$$\begin{aligned} \max/\min \quad & J(\mathbf{U}, \boldsymbol{\alpha}) \\ \text{subject to} \quad & \mathbf{U} = G(\mathbf{U}, \boldsymbol{\alpha}) \end{aligned} \quad (3.16)$$

The Lagrangian functional associated with the above constrained optimisation problem is given by

$$L(\mathbf{U}, \boldsymbol{\Psi}, \boldsymbol{\alpha}) = J(\mathbf{U}, \boldsymbol{\alpha}) - \boldsymbol{\Psi}^T \{\mathbf{U} - G(\mathbf{U}, \boldsymbol{\alpha})\} \quad (3.17)$$

Here  $\boldsymbol{\Psi}$  is the adjoint state vector or the Lagrange multiplier. From the first order necessary conditions (KKT conditions) for optimality of the Lagrangian function, the discrete adjoint equations can be derived in the fixed point form as

$$\boldsymbol{\Psi}^{n+1} = \left( \frac{\partial G}{\partial \mathbf{U}} \right)^T \boldsymbol{\Psi}^n + \left( \frac{\partial J}{\partial \mathbf{U}} \right)^T \quad (3.18)$$

A general notation for the adjoint fixed point iterative scheme can be written as

$$\boldsymbol{\Psi}^{n+1} = \overline{G}(\boldsymbol{\Psi}^n, \mathbf{U}, \boldsymbol{\alpha}) \quad (3.19)$$

where  $\overline{G}$  represents a pseudo-time iteration of the discrete adjoint LSKUM solver for Euler equations. Finally, the converged primal and adjoint solutions are used to evaluate the sensitivities of the objective functional  $L(\mathbf{U}, \boldsymbol{\Psi}, \boldsymbol{\alpha})$  with respect to the vector of shape variables  $\boldsymbol{\alpha}$  as

$$\frac{dL}{d\boldsymbol{\alpha}} = \frac{\partial J}{\partial \boldsymbol{\alpha}} + \boldsymbol{\Psi}^T \frac{\partial G}{\partial \boldsymbol{\alpha}} \quad (3.20)$$

In the present work, the adjoint LSKUM solver is constructed by Algorithmically Differentiating (AD) [20] the primal LSKUM solver. From eqns. (3.18) and (3.20), it is clear that accurate computation of the shape sensitivities require the exact differentiation of  $J$  and  $G$ . An advantage of AD is that it performs the exact differentiation of these terms and thus yields accurate shape sensitivities  $J_{\boldsymbol{\alpha}}$ . To ease the differentiation, the AD tool Tapenade [21] has been employed.

For the sake of simplicity, the mathematical formulation of the discrete optimisation problem is shown with first-order forward difference formula for the time derivative. For higher-order temporal schemes, the adjoint equations can be derived in a similar fashion. Furthermore, there could be aerodynamic and geometric constraints on the control vector. For example, the aerodynamic constraints are imposed on the lift and pitching moment coefficients. The geometric constraints could be on the thickness and trailing edge angle of the airfoils.

## 4 Gradient smoothing

In aerodynamic shape optimisation using free node shape representation, the points that define the surface of the geometry of interest are chosen as the control variables. This leads to a large number of control variables. An advantage of this approach is that the point-wise shape sensitivities give an insight into the regions that significantly influence the objective function and where incremental shape changes can occur. In general, the point-wise shape sensitivities thus obtained from the adjoint solver are non-smooth and may contain high frequency oscillations. The behavior of the sensitivities is almost tending to the Weierstrass function (continuous everywhere but differentiable nowhere). The direct use of raw sensitivities in the gradient descent algorithm leads to non-smooth and oscillatory shapes that are unphysical.

In order to ensure that the shapes obtained during optimisation cycles remain smooth, the sensitivities  $J_\alpha$  must be smoothed. Mathematically speaking, we must confine  $J_\alpha$  to a subspace of smooth differentiable functions such as Sobolev space instead of  $L_2$  space. The smoothed sensitivities can be obtained using the well-known Sobolev gradient smoothing [1; 22] or filtering techniques [23]. In the present work, we have used the Sobolev smoothing approach.

Using Sobolev smoothing, the smoothed shape sensitivities  $\bar{J}_\alpha$  along the arcual direction  $s$  of the boundary of a shape in two-dimensions can be obtained by solving

$$\bar{J}_\alpha - \frac{\partial}{\partial s} \varepsilon \frac{\partial \bar{J}_\alpha}{\partial s} = J_\alpha \quad (4.21)$$

where,  $\varepsilon$  is the smoothing coefficient, which can be non-dimensionalised by introducing a user defined parameter  $\sigma$  such that

$$\varepsilon = \sigma \Delta s^2 \quad (4.22)$$

Here  $\Delta s$  is the characteristic length along the arcual direction. The gradient smoothing in eqn. (4.21) can be mathematically explained in terms of confining the gradient to Sobolev space having an inner product of any two functions  $u$  and  $v$  as

$$(u, v)_{sob} = \int_C (uv + \varepsilon u'v') ds \quad (4.23)$$

Here  $C$  is a bounding curve. When applied to  $\bar{J}_\alpha$ , this induces the Sobolev norm,

$$\|\bar{J}_\alpha\|_{sob}^2 = \int (\bar{J}_\alpha^2 + \varepsilon \bar{J}_\alpha'^2) ds = \int \bar{J}_\alpha^2 ds + \int \varepsilon \bar{J}_\alpha'^2 ds \quad (4.24)$$

The terms on the right hand side of the above equation can be defined as,

$$\chi_1 = \int \bar{J}_\alpha^2 ds, \quad \chi_2 = \int \bar{J}_\alpha'^2 ds \quad (4.25)$$

Here  $\chi_1$  is a measure of  $\bar{J}_\alpha$  and  $\chi_2$  is a measure of the gradient of  $\bar{J}_\alpha$  along the arcual direction of the boundary of the shape. The ratio of these terms is defined as

$$\chi = \frac{\chi_2}{\chi_1} = \frac{\int \bar{J}_\alpha'^2 ds}{\int \bar{J}_\alpha^2 ds} \quad (4.26)$$

The ratio,  $\chi$  is a good indicator of the smoothness of  $\bar{J}_\alpha$  and therefore can be used to select  $\sigma$ . For a non-smooth highly oscillatory  $\bar{J}_\alpha$ , the values of  $\chi_2$  will be very large compared to  $\chi_1$ . For smooth  $\bar{J}_\alpha$ , the values of  $\chi$  will be small. Hence  $\chi$  is helpful in understanding the effect of  $\sigma$ . This expectation is borne out by our calculations in Section (5).

## 5 Results and discussion

The meshfree LSKUM primal and adjoint solvers have been applied to perform aerodynamic shape optimisation of the NACA0012 airfoil in transonic and supersonic flows. For all the test cases, a point cloud with 23,566 points and 936 points on the airfoil surface are used. The following test cases are considered.

- Case 1: Drag minimisation of the airfoil at  $M_\infty = 0.85$  and  $AoA = 1^\circ$ .
- Case 2: Drag minimisation of the airfoil at  $M_\infty = 0.85$  and  $AoA = 1^\circ$  with constant lift.
- Case 3: Drag minimisation of the airfoil at  $M_\infty = 1.2$  and  $AoA = 0^\circ$  with constant lift and trailing edge angle.

### 5.1 Case 1: Drag minimisation of the NACA0012 airfoil in transonic flow

The flow conditions for this case are given by the Mach number,  $M_\infty = 0.85$  and angle of attack,  $AoA = 1^\circ$ . The flow field on the baseline geometry is characterised by a strong shock on the upper surface and a weak shock on the lower surface. Hence the drag is due to the presence of shocks. The objective function is defined as the minimisation of the drag coefficient. The geometric constraints are that the chord length and the trailing edge point are fixed. This implies that the  $y$ -coordinates of the rest of the points on the airfoil are allowed to perturb, and therefore form the control vector.

Figure 1 shows the plot of shape sensitivities with and without smoothing along the arcual length of the baseline airfoil. For the numerical experiments, the characteristic length scale,  $\Delta s$  defined in eqn. (4.22) is chosen as the average surface point spacing on the airfoil. We observe that the raw shape sensitivities  $J_\alpha$  obtained from the adjoint solver are non-smooth and highly oscillatory. The smoothed sensitivities  $\bar{J}_\alpha$  becomes smoother with increasing values of  $\sigma$ . Figure 2a shows that the Sobolev norm of the smoothed sensitivities  $\bar{J}_\alpha$  reduces with increase in  $\sigma$ . Figure 2b shows the plot of the ratio  $\chi$ , defined in eqn. (4.26). This plot explains the reason for increased smoothness of  $\bar{J}_\alpha$  and the reduction in its Sobolev norm for large values of  $\sigma$ . For small values of  $\sigma$ ,  $\chi$  is large as  $\chi_2$ , a measure of non-smoothness is large. Note that the value of  $\chi$  computed from the non-smooth sensitivities  $J_\alpha$  is of the order of  $O(10^6)$ . On the other hand, large values of  $\sigma$  results in smaller gradients of  $\bar{J}_\alpha$  and thus reduces the contribution of  $\chi_2$  term. This in turn resulted in smaller values of the Sobolev norm. The ratio,  $\chi$  identifies the large value of  $\sigma$  that results in adequate smoothing of shape sensitivities. Figure 3 shows the Sobolev smoothed shape sensitivities on the airfoil with  $\sigma = 3 \times 10^5$ . Thus it can be inferred from the above discussion that the Sobolev smoothing attenuates the rapid variation of the raw shape sensitivities along the arcual direction.

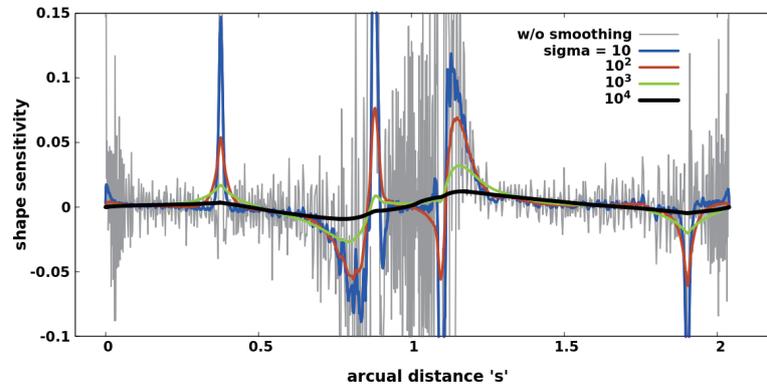


Figure 1: Variation of  $J_\alpha$  and  $\bar{J}_\alpha$  along the arcual distance of airfoil for various values of  $\sigma$ .

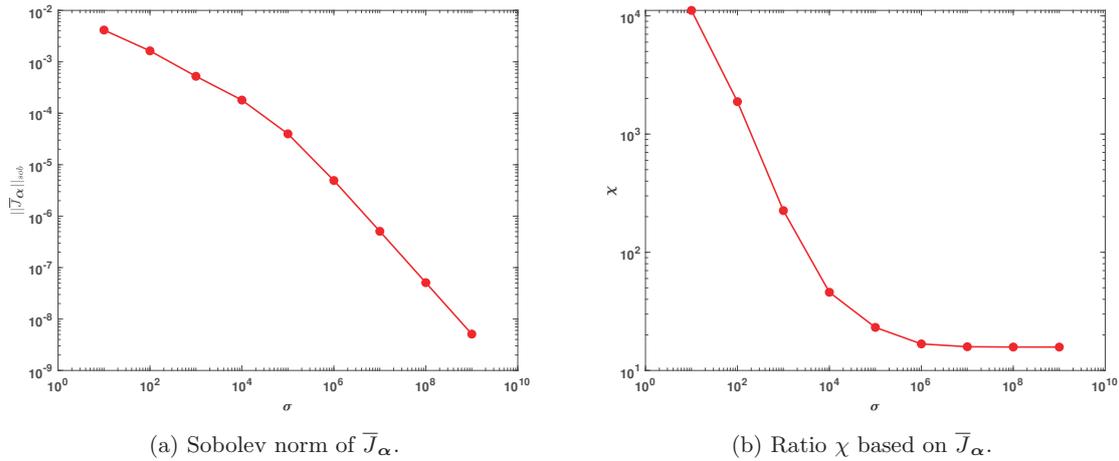


Figure 2: Case1: Drag minimisation of the NACA0012 airfoil at  $M = 0.85$  and  $AoA = 1^0$ .

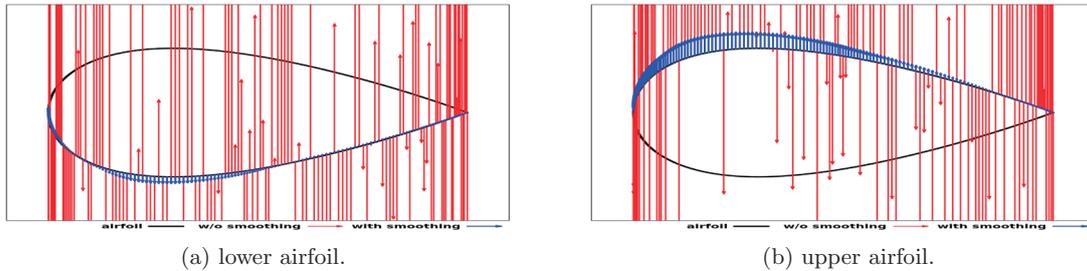


Figure 3: A comparison of non-smooth and smoothed shape sensitivities for the baseline airfoil.

Figure 4 shows the baseline shape of the airfoil and the optimal shape obtained after 11 cycles of optimisation. Figure 5 shows a comparison of the Mach contours. We observe that the baseline shape has a strong shock on the upper surface (shock is bent more) and a weak shock on the lower surface. For the optimal shape, the strength of the shock on the upper surface is reduced (less bending) while on the lower surface the shock has almost vanished. Also notice the lower coalescence of Mach waves on the optimal shape as compared to baseline shape suggesting lower wave drag. The entropy contours in Figure 6 show that the loss in the entropy is much less for the optimal shape as compared to the baseline shape thereby causing less drag. Figure 7 shows the surface pressure distribution ( $C_p$ ). We can notice the reduction in strength of shocks and the location of shock on the upper surface has moved upstream. The changes in  $C_p$  near the leading edge that affect acceleration of the flow can also be seen clearly. Pressure distribution is approaching the well-known roof top distribution. The overall effect of these flow changes can be easily seen in Figure 8 that shows the reduction in drag as optimisation proceeds. Table 1 shows a comparison of the lift and drag coefficients for the baseline and optimal shapes. We observe that the optimisation reduced the drag by around 98%. Since there is no constraint on the lift during optimisation, the pressure differential between the pressure and suction sides is reduced, thus leading to the decrease in the lift coefficient.

During the optimisation, as the shape changes, the interior points in the vicinity of the airfoil may move close to it or fall inside the airfoil. In the meshfree framework, the points falling inside the airfoil can be blanked, and the connectivity of the wall and nearest interior points can be regenerated. However, as the interior points move closer to the wall boundary, the least-squares matrix of the wall points may

become ill-conditioned. This may affect the accuracy of the derivative or even result in code divergence.

Alternatively, each nearby interior point can be moved according to the perturbation of its nearest wall or the shape point. Note that the set of nearby interior points can be defined by choosing a bounding box around the airfoil. An advantage of this approach is that there is no need for point blanking, and the connectivity set obtained for the baseline shape can be used throughout the optimisation cycles. Furthermore, the least-squares matrices of the wall points remain well-conditioned. In this research, we pursued this approach to handle the point movement during optimisation.

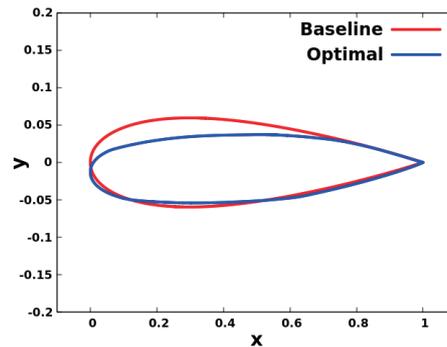
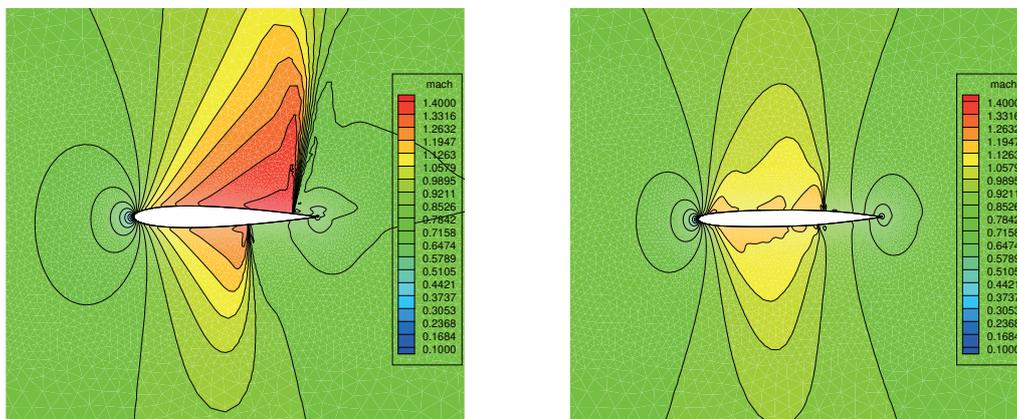


Figure 4: Case 1: Baseline and optimal airfoil shapes.



(a) Baseline shape

(b) Optimal shape

Figure 5: Case1: A comparison of the Mach contours for the baseline and optimal airfoils.

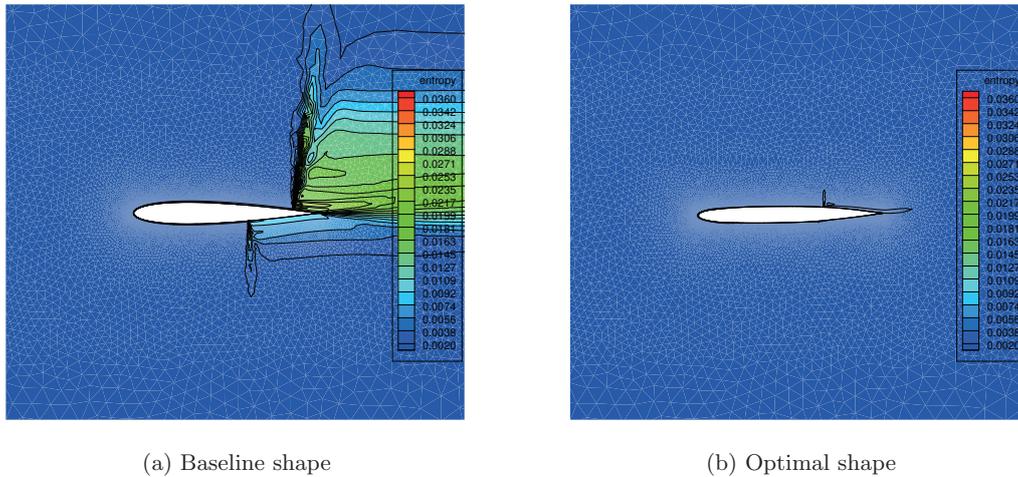


Figure 6: Case1: A comparison of the entropy contours for the baseline and optimal airfoils.

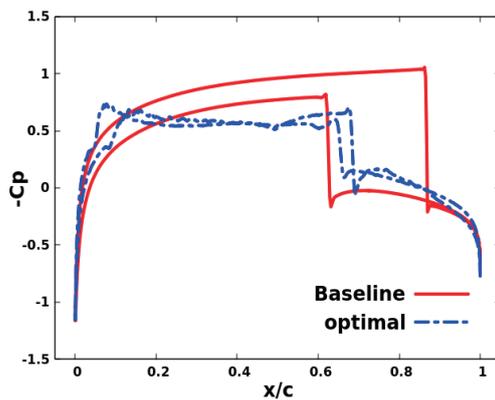
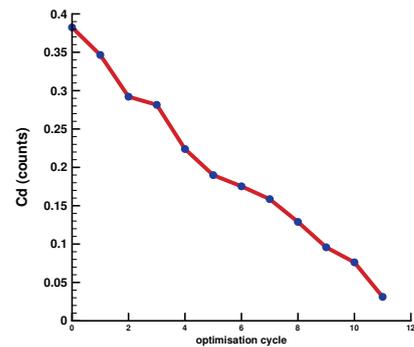
Figure 7: Case1: A comparison of the  $C_p$  distribution for the baseline and optimal airfoils.

Figure 8: Case1: Drag history.

Shape	Lift coefficient ( $C_L$ )	Drag coefficient ( $C_D$ )
Baseline	0.3824	0.0573
Optimal	0.0312	0.0013

Table 1: Case1: Lift and drag coefficients for the baseline and optimal airfoils.

## 5.2 Case 2: Drag minimisation of the NACA0012 airfoil in transonic flow with lift constraint

The next test case under investigation is the drag minimisation of the NACA0012 airfoil by maintaining the lift of the baseline airfoil. The flow conditions, geometric constraints, and the control vector are the same as in Case 1. The lift constraint is imposed through the quadratic penalty function approach. The

objective function  $J$  with the penalty function is then given by

$$\min J = C_D + \frac{1}{2}\mu(C_{Lb} - C_L)^2 \quad (5.27)$$

where,  $\mu$  is the penalty constant,  $C_L$  is the computed lift and  $C_{Lb}$  is lift obtained from the baseline shape.

Figure 9 compares the baseline and optimal airfoil shapes. Figure 10 shows the Mach contours. It can be noticed that the reduction in strength of shock on the upper surface is not as high as the Case 1, while on the lower surface, the shock has almost vanished. The coalescence of Mach waves is less, suggesting less wave drag. The entropy contours in Figure 11 show the reduced shock strength. The shape changed in such a way that the pressure differential on top and bottom surfaces was maintained to achieve the desired lift. This can be seen from the  $C_p$  distribution plot in Figure 12. We can notice a small change in the strength of shock on the upper side, while shock has moved and reduced in strength on the lower side. Figure 13 shows the drag history, which indicates the oscillatory nature of convergence compared to Case 1. This is due to the penalty term corrections to maintain the required lift. Table 2 shows a comparison of the lift and drag coefficients. For the optimal shape, the drag is reduced by around 67% while maintaining the lift generated by the baseline airfoil.

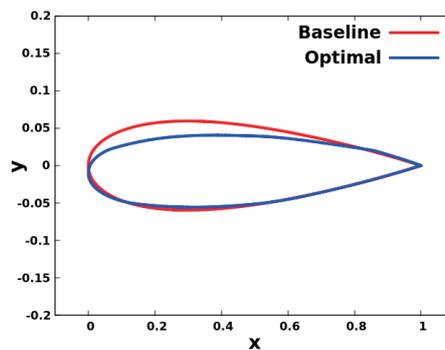
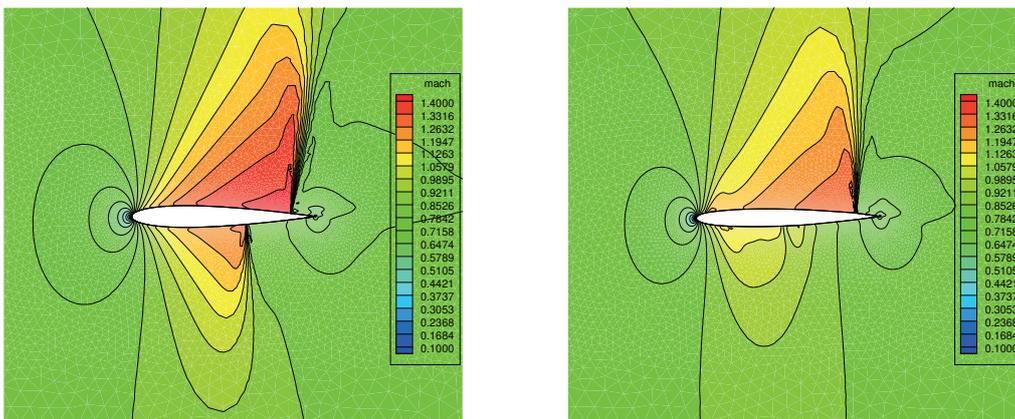


Figure 9: Case2: Baseline and optimal airfoil shapes.



(a) Baseline shape

(b) Optimal shape

Figure 10: Case2: A comparison of the Mach contours for the baseline and optimal airfoils.

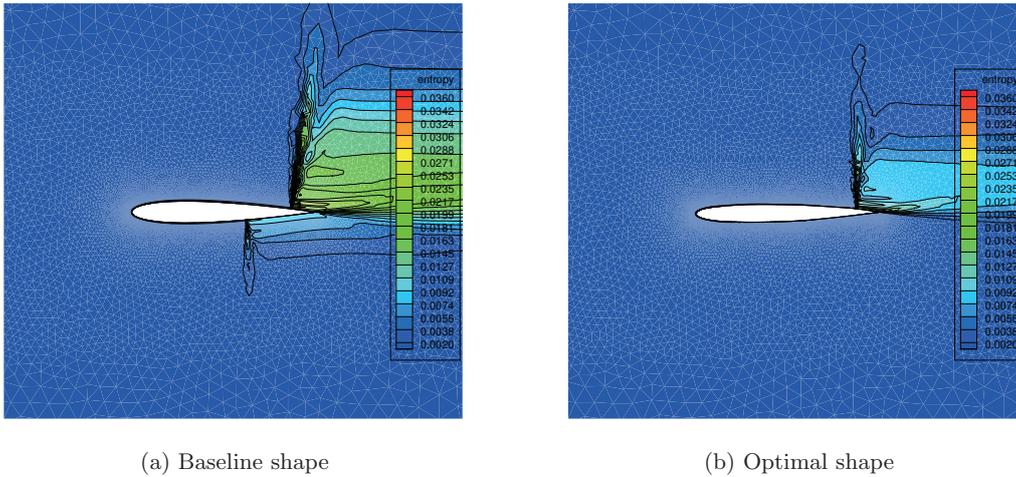


Figure 11: Case2: A comparison of the entropy contours for the baseline and optimal airfoils.

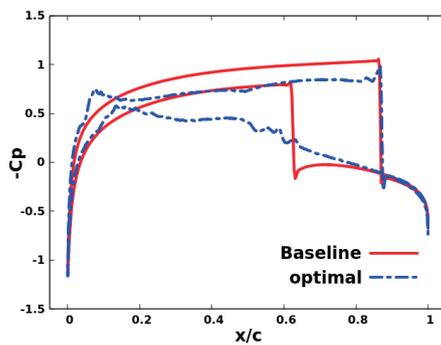
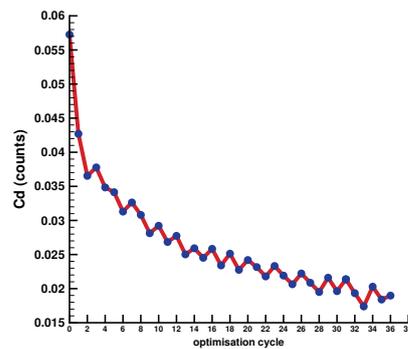
Figure 12: Case2: A comparison of the  $C_p$  distribution for the baseline and optimal airfoils.

Figure 13: Case2: Drag history.

Shape	Lift coefficient ( $C_L$ )	Drag coefficient ( $C_D$ )
Baseline	0.3824	0.0573
Optimal	0.3818	0.0190

Table 2: Case2: Lift and drag coefficients for the baseline and optimal airfoils.

### 5.3 Case 3: Drag minimisation of the NACA0012 airfoil in supersonic flow

The flow conditions for this test case are supersonic, with  $M_\infty = 1.2$  and  $AoA = 0^\circ$ . The flow features are characterised by a bow shock that appears ahead of the airfoil and the trailing edge fishtail shock. The symmetry of the airfoil with zero angle of attack causes the lift to be zero. For this test case, the objective function is defined as minimising the drag coefficient by maintaining zero lift. Apart from the geometric constraints defined in Case 1, the maximum thickness of the airfoil section is allowed to halve. Additionally, the trailing edge angle is constrained to be the same as the baseline airfoil angle.

The constraints on the lift and trailing edge angle are imposed by the quadratic penalty functions. The objective function is then given by

$$\min J = C_D + \frac{1}{2}\mu_{C_L}(C_{L_t} - C_L)^2 + \frac{1}{2}\mu_{TE}(TE_b - TE)^2 \quad (5.28)$$

where,  $\mu_{C_L}$  and  $\mu_{TE}$  are respectively the penalty constants for the lift and trailing edge angle.  $C_L$  is the computed lift and  $C_{L_t}$  is the target lift, which is zero.  $TE$  is the trailing edge angle of the shape at any cycle and  $TE_b$  is the trailing edge angle corresponding to the baseline shape.

This test case is interesting as the bow shock, which contributes maximally to the objective function, is off the body. We notice from Figure 14 that the shock stand-off distance reduces as a consequence of optimisation. This is due to the change in geometry near the leading edge. Also, the oblique shock away from the line of symmetry becomes weaker, thus reducing the wave drag. Entropy contours from Figure 15 show that the tail shock almost disappears. Figure 16 compares the baseline and optimal airfoil shapes. Figure 17 shows the zoomed view of the airfoils near the trailing edge. We observe that the trailing edge angle is preserved over the optimisation cycles. Figure 18 shows the objective function history. Table 3 shows a comparison of the lift and drag coefficients, the maximum thickness of the section, and the trailing edge angle for the baseline and optimal shapes. Optimisation reduced the drag by around 43% while maintaining the lift close to zero. The maximum thickness of the airfoil section is reduced by 25%. Throughout the optimisation, the interior points near the wall boundary are moved so that no point comes too close to the wall or falls inside the airfoil. Similar to the earlier test cases, the point connectivity remained the same.

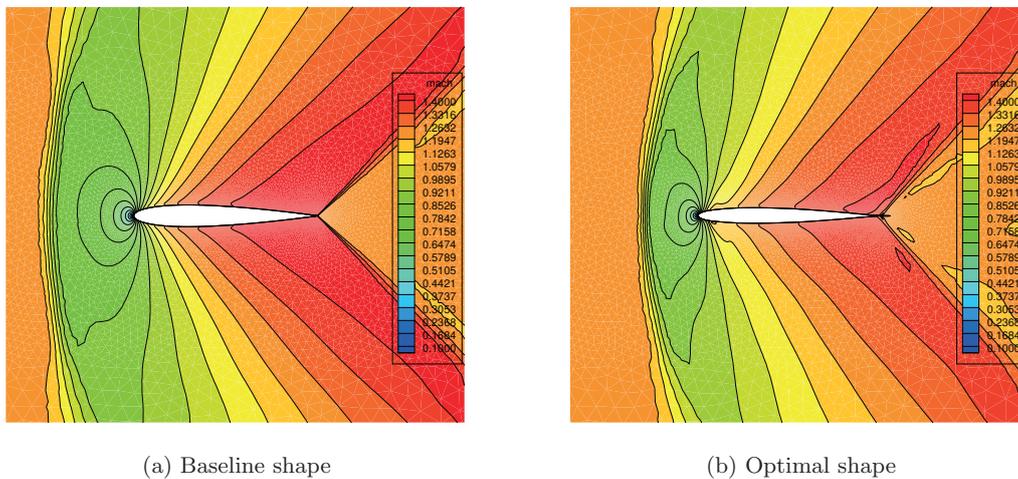


Figure 14: Case3: A comparison of the Mach contours for the baseline and optimal airfoils.

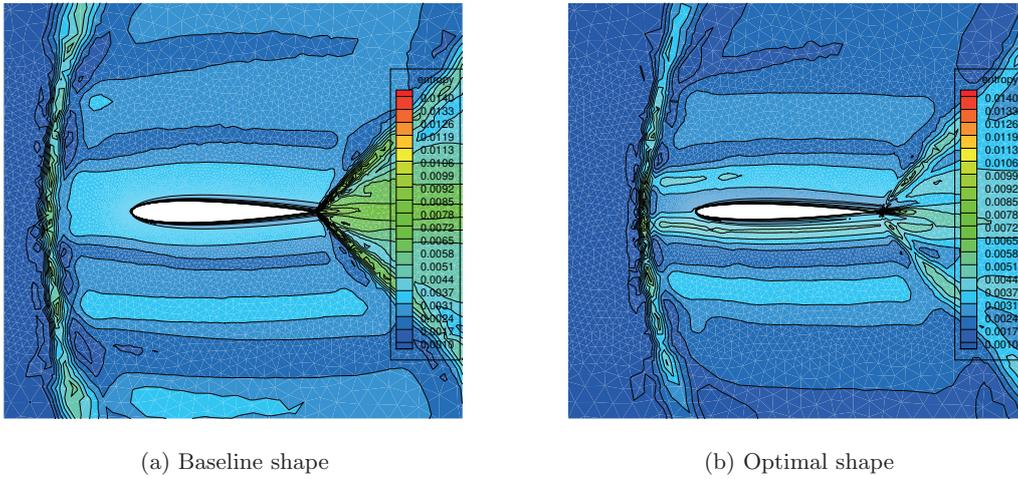


Figure 15: Case3: A comparison of the entropy contours for the baseline and optimal airfoils.

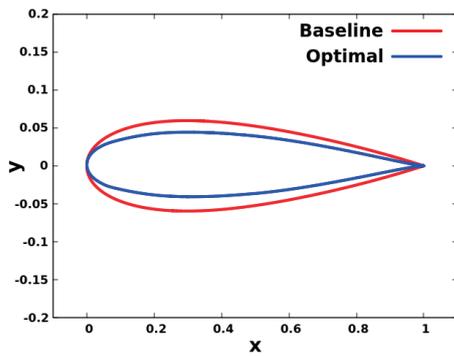


Figure 16: Case3: Baseline and optimal airfoil shapes.

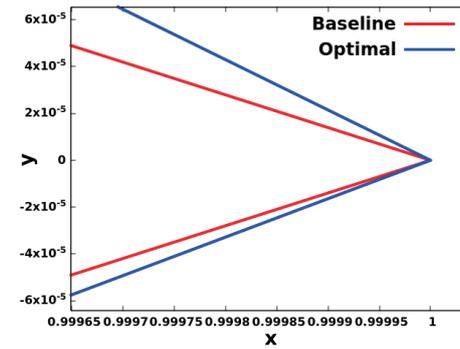


Figure 17: Case3: Zoomed view of the trailing edge.

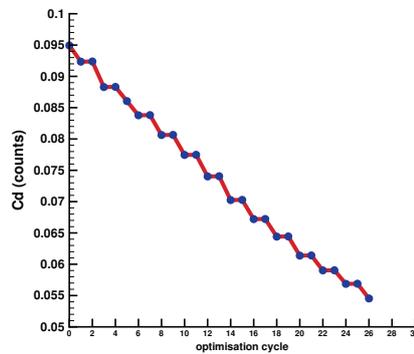


Figure 18: Case3: Drag history.

Shape	$C_L$	$C_D$	Maximum thickness	Trailing edge angle
Baseline	$1.6154 \times 10^{-3}$	0.0949	0.1190	15.90
Optimal	$8.6722 \times 10^{-4}$	0.0546	0.0893	15.35

Table 3: Case 3: A comparison of the lift and drag coefficients, maximum thickness and the trailing edge angle for the baseline and optimal airfoils.

## 6 Conclusions

In this paper, we performed the aerodynamic shape optimisation of the NACA0012 airfoil using the mesh-free LSKUM based primal and discrete adjoint solvers. The adjoint meshfree solver to compute the shape sensitivities is constructed by algorithmically differentiating the underlying primal LSKUM solver. The shape sensitivities using the free node shape parametric representation provide insight into the regions of the shape that significantly influence the objective function. Equivalently, the sensitivity map shows where incremental shape changes can occur. The raw non-smooth shape sensitivities are smoothed using the Sobolev gradient smoothing. The ratio of two terms ( $\chi_2/\chi_1$ ) in the Sobolev norm is used to obtain a suitable smoothing parameter value, which yields adequately smooth shape sensitivities.

Three test cases for flow past the NACA0012 airfoil were chosen with aerodynamic and geometric constraints. Numerical results have shown that the optimisation yielded desired results. During optimisation, the movement of points on the surface of the airfoil led to minimal changes in the point cloud. Furthermore, there was no need to regenerate the connectivity. Hence the point cloud alteration was easier to implement than mesh deformation, presumably due to the flexibility of the LSKUM solver. Further studies are underway to exploit shape optimisation using meshfree LSKUM.

## References

- [1] A. Jameson, "Aerodynamic design via control theory," *J. Sci. Comput.*, vol. 3, pp. 233–260, 1988.
- [2] J. Elliott and J. Peraire, "Practical three-dimensional aerodynamic design and optimization using unstructured meshes," *AIAA Journal*, vol. 35, no. 9, pp. 1479–1485, 1997.
- [3] E. J. Nielsen and W. K. Anderson, "Aerodynamic design optimization on unstructured meshes using the navier-stokes equations," *AIAA Journal*, vol. 37, no. 11, pp. 185–191, 1999.
- [4] J. Peter and R. Dwight, "Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches and applications," *Computers & Fluids*, vol. 39, pp. 373–391, 2010.
- [5] A. K. Ghosh and S. M. Deshpande, "Least squares kinetic upwind method for inviscid compressible flows," *AIAA paper 1995-1735*, 1995.
- [6] S. M. Deshpande, P. S. Kulkarni, and A. K. Ghosh, "New developments in kinetic schemes," *Computers Math. Applic.*, vol. 35, no. 1, pp. 75–93, 1998.
- [7] C. Praveen, A. K. Ghosh, and S. M. Deshpande, "Positivity preservation, stencil selection and applications of LSKUM to 3-D inviscid flows," *Computers and Fluids*, vol. 38, no. 8, pp. 1481–1494, 2009.
- [8] A. Mahendra, R. Singh, and G. Gouthaman, "Meshless kinetic upwind method for compressible, viscous rotating flows," *Computers & Fluids*, vol. 46, no. 1, pp. 325–332, Jul. 2011. [Online]. Available: <https://doi.org/10.1016/j.compfluid.2010.10.015>
- [9] K. Anandhanarayanan, K. Arora, V. Shah, R. Krishnamurthy, and D. Chakraborty, "Separation dynamics of air-to-air missile using a grid-free euler solver," *Journal of Aircraft*, vol. 50, no. 3, pp. 725–731, 2013.
- [10] V. Ramesh and S. M. Deshpande, "Kinetic mesh-free method for flutter prediction in turbomachines," *Sadhana*, vol. 39, no. 1, pp. 149–164, 2014.
- [11] K. Anandhanarayanan, R. Krishnamurthy, and D. Chakraborty, "Development and validation of a grid-free viscous solver," *AIAA Journal*, vol. 54, no. 10, pp. 3312–3315, 2016.
- [12] E. Özkaya and N. Gauger, "Automatic transition from simulation to one-shot shape optimization with navier-stokes equations," *GAMM-Mitteilungen*, vol. 33, no. 2, pp. 133–147, 2010.
- [13] A. Nemili, E. Özkaya, N. R. Gauger, F. Kramer, and F. Thiele, "Accurate discrete adjoint approach for optimal active separation control," *AIAA Journal*, vol. 55, no. 9, pp. 3016–3026, 2017.
- [14] E. Özkaya, J. A. Hay, N. R. Gauger, and F. Thiele, "Development of an adjoint CAA solver for design optimization of acoustic liners," *AIAA paper 2016-2778*, 2016.
- [15] J. C. Mandal and S. M. Deshpande, "Kinetic flux vector splitting for Euler equations," *Comp. & Fluids*, vol. 23, no. 2, pp. 447–478, 1994.
- [16] S. M. Deshpande, "A second-order accurate kinetic-theory-based method for inviscid compressible flows," *NASA-TP-2613*, 1986.
- [17] R. Courant, E. Issacson, and M. Rees, "On the solution of nonlinear hyperbolic differential equations by finite differences," *Comm. Pure Appl. Math.*, vol. 5, pp. 243–255, 1952.
- [18] S. Deshpande, V. Ramesh, K. Malagi, and K. Arora, "Least squares kinetic upwind mesh-free method," *Defence Science Journal*, vol. 60, no. 6, pp. 583–597, Sep. 2010. [Online]. Available: <https://publications.drdo.gov.in/ojs/index.php/dsj/article/view/579>
- [19] S. M. Deshpande, K. Anandhanarayanan, C. Praveen, and V. Ramesh, "Theory and application of 3-D LSKUM based on entropy variables," *Int. J. Numer. Meth. Fluids*, vol. 40, pp. 47–62, 2002.
- [20] A. Griewank and A. Walther, *Evaluating derivatives: Principles and techniques of Algorithmic Differentiation*. SIAM, 2008.
- [21] L. Hascoët and V. Pascual, "The Tapenade Automatic Differentiation tool: Principles, Model, and Specification," *ACM Transactions On Mathematical Software*, vol. 39, no. 3, 2013. [Online]. Available: <http://dx.doi.org/10.1145/2450153.2450158>
- [22] A. Jaworski and J.-D. Müller, "Toward modular multigrid design optimisation," in *Advances in Automatic Differentiation*, C. H. Bischof, H. M. Bücker, P. D. Hovland, U. Naumann, and J. Utke, Eds. Springer, 2008, pp. 281–291.

- [23] A. Stuck and T. Rung, "Filtered gradients for adjoint-based shape optimisation," *AIAA paper 2011-3072*, 2011.

## Acknowledgements

The third author gratefully acknowledges the Aeronautics Research and Development Board (AR&DB), DRDO, for funding this research under the project number 1961. He also acknowledges the computing time provided on the high performance computing facility, Sharanga, at the Birla Institute of Technology and Science - Pilani, Hyderabad Campus.