

Adjoint based Sensitivity Analysis for Navier-Stokes Equations

Srikanth Sathyanarayana ^{*}, Anil Nemili [†],

Department of Mathematics, BITS-Pilani, Hyderabad Campus, Hyderabad, 500078

Ashish Bhole [‡]

Université de Nice Sophia-Antipolis, C.N.R.S. U.M.R. & Inria Sophia-Antipolis, France

Aman Saxena [§] and Praveen Chandrashekar [¶]

TIFR Centre for Applicable Mathematics, Bengaluru, 560065

This paper presents the development of a discrete adjoint approach for accurate computation of shape sensitivities in steady laminar flows governed by the compressible Navier-Stokes equations. The adjoint laminar flow solver is generated by Algorithmically Differentiating the underlying primal solver. It is well-known that the adjoint solvers based on black-box differentiation demand significant computational resources, both in terms of memory and run-time. To enhance the computational efficiency, various advanced algorithmic differentiation techniques are employed. The performance of the adjoint solver in accurate computation of shape sensitivities is assessed by applying it to the test cases of laminar flow around NACA 0012 airfoil and analytical 3D body of revolution. Sensitivities based on the adjoint code are compared with the values obtained from direct methods such as finite differences and corresponding tangent linear code. Shape sensitivity plots are presented to analyse the regions that contribute to the desired objective function.

Keywords: Aerodynamic optimisation, shape sensitivities, discrete adjoints, Algorithmic Differentiation, laminar flows, Navier-Stokes equations.

I. Introduction

Aerodynamic optimisation involving Navier-Stokes equations has been an area of significant relevance to aerospace industry. Many optimisation problems in this field of research can be characterised by an objective function that is to be maximised or minimised, a set of flow field and geometry related constraints and a large number of control variables. The optimal set of control variables that maximises or minimises the desired objective function is then obtained by employing the gradient based optimisation algorithms.

Central to the success of gradient algorithms is accurate and efficient computation of the sensitivity gradients of the given objective function with respect to the control variables. Once evaluated, the sensitivities can then be used to drive an optimisation procedure until a desired convergence in the objective function is achieved. In general, the solution techniques for the evaluation of sensitivities can be classified into two groups, namely, direct methods and adjoint methods. The direct methods include finite differences, complex Taylor series expansion (CTSE) method and tangent linear mode differentiation (also known as

^{*}srikanth.c.sathyanarayana@gmail.com

[†]anil@hyderabad.bits-pilani.ac.in

[‡]ashishbhole07@gmail.com

[§]amansaxena.iitg@gmail.com

[¶]praveen@tifrbng.res.in

forward mode differentiation). Although easy to implement, a major drawback of these methods is that they give sensitivity information with respect to only one control variable. In order to find the complete gradient vector, numerical simulations need to be performed separately for each control variable. Thus the computational costs of evaluating sensitivities grow linearly with the number of control variables. For design optimisation problems involving a large number of control variables, these costs can become prohibitively expensive. On the other hand, the adjoint methods compute the complete gradient vector at an expense that is independent of the number of control variables. Due to this advantage, the use of adjoint methods in gradient based optimisation has received much attention.

The adjoint methods can be classified into continuous [1] or discrete [2] versions, depending on the order in which the linearisation and discretisation of the governing PDEs is performed. The discrete approach is preferred over the continuous as it yields an adjoint solver that is robust and consistent to the primal solver [3]. During its initial years, the discrete adjoint solvers were developed using the *hand-discrete* approach, where the linearisation of the adjoint equations is performed by hand. With the continuous development of Algorithmic Differentiation (AD) techniques [4], one can develop the discrete adjoint solvers by directly differentiating the underlying primal flow solvers. An advantage of this approach is that AD tools can perform the exact differentiation of all terms in the primal solver with much ease. Therefore, the derived adjoint solver is always consistent to the primal and hence gives accurate sensitivities at any residual level achieved by the primal solver. Furthermore, the adjoint solver inherits the asymptotic convergence and robustness of the primal solver [4]. In [5–7], this approach has been successfully employed to develop robust discrete adjoint solvers for steady and unsteady flows.

In this paper, we develop a discrete adjoint approach based on AD for accurate computation of shape sensitivities in laminar flows governed by compressible Navier-Stokes equations. The rest of the paper is organised as follows. Section II presents the details pertaining to the construction of an efficient and accurate discrete adjoint approach. In section III, numerical results are shown for standard test cases to verify the accuracy of the adjoint sensitivities. Furthermore, a detailed analysis of shape sensitivities for several objective functions of aerodynamic importance are presented. Finally, conclusions and future research plans are presented in Section IV.

II. A discrete adjoint approach for shape sensitivities

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

$$\begin{aligned} \text{max/min} \quad & J(\mathbf{U}, \boldsymbol{\alpha}) \\ \text{subject to} \quad & \mathbf{C}(\mathbf{U}, \boldsymbol{\alpha}) = 0 \end{aligned} \tag{1}$$

where J is a scalar objective function like the aerodynamic lift or drag coefficients, \mathbf{U} is the conserved vector and $\boldsymbol{\alpha}$ is the vector of control variables comprising the shape coordinates (x, y, z) that define the geometry of the configuration. In the present work, $\mathbf{C}(\mathbf{U}, \boldsymbol{\alpha}) = 0$ represents the compressible Navier-Stokes equations along with boundary conditions. In the semi-discrete form, the constraints can be written as

$$\frac{d\mathbf{U}}{dt} + \mathbf{R}_s(\mathbf{U}, \boldsymbol{\alpha}) = 0 \tag{2}$$

where, $\mathbf{R}_s(\mathbf{U}, \boldsymbol{\alpha})$ is the discrete residual vector obtained after the finite volume discretisation of the spatial derivatives in the governing equations. Approximating the temporal derivative using the first order forward

difference formula, the state-update formula can be written as

$$\mathbf{U}^{n+1} = \mathbf{U}^n - \Delta t \mathbf{R}_s(\mathbf{U}^n, \boldsymbol{\alpha}) \quad (3)$$

Here, n represents a pseudo-time iteration, since we are interested in the steady-state solution. The state-update formula can be written in the fixed point form as

$$\mathbf{U}^{n+1} = G(\mathbf{U}^n, \boldsymbol{\alpha}) \quad (4)$$

Here, G represents an iteration of the finite volume scheme employed for the numerical solution of the Navier-Stokes equations. The above fixed point iteration converges to the steady state solution \mathbf{U} , given by

$$\mathbf{U} = G(\mathbf{U}, \boldsymbol{\alpha}) \quad (5)$$

In the discrete form, the optimisation problem defined in eq. (1) can be formulated as

$$\begin{aligned} \max/\min \quad & J(\mathbf{U}, \boldsymbol{\alpha}) \\ \text{subject to} \quad & \mathbf{U} = G(\mathbf{U}, \boldsymbol{\alpha}) \end{aligned} \quad (6)$$

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

$$L(\mathbf{U}, \boldsymbol{\Psi}, \boldsymbol{\alpha}) = J(\mathbf{U}, \boldsymbol{\alpha}) - \boldsymbol{\Psi}^T \{\mathbf{U} - G(\mathbf{U}, \boldsymbol{\alpha})\} \quad (7)$$

Here $\boldsymbol{\Psi}$ is the adjoint state vector or the Lagrange multiplier. The first order necessary conditions (KKT conditions) for optimality of the Lagrangian function are given by

$$\frac{\partial L}{\partial \boldsymbol{\Psi}} = 0 \quad (\text{State equations}) \quad (8a)$$

$$\frac{\partial L}{\partial \mathbf{U}} = 0 \quad (\text{Adjoint equations}) \quad (8b)$$

$$\frac{\partial L}{\partial \boldsymbol{\alpha}} = 0 \quad (\text{Control equation}) \quad (8c)$$

From eq. (8b), the discrete adjoint equations can be derived in the fixed point form as

$$\boldsymbol{\Psi}^{n+1} = \left(\frac{\partial G}{\partial \mathbf{U}} \right)^T \boldsymbol{\Psi}^n + \left(\frac{\partial J}{\partial \mathbf{U}} \right)^T \quad (9)$$

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

$$\boldsymbol{\Psi}^{n+1} = \overline{G}(\boldsymbol{\Psi}^n, \mathbf{U}, \boldsymbol{\alpha}) \quad (10)$$

where \overline{G} represents a pseudo-time iteration of the discrete adjoint solver for compressible Navier-Stokes equations. The above fixed point iteration converges to the adjoint solution $\boldsymbol{\Psi}$, given by

$$\boldsymbol{\Psi} = \overline{G}(\boldsymbol{\Psi}, \mathbf{U}, \boldsymbol{\alpha}) \quad (11)$$

Finally, the primal and adjoint solutions from eqs. (5) and (11) are substituted in eq. (8c) to evaluate the sensitivities of the objective functional with respect to the shape variables as

$$\frac{dL}{d\boldsymbol{\alpha}} = \frac{\partial J}{\partial \boldsymbol{\alpha}} + \boldsymbol{\Psi}^T \frac{\partial G}{\partial \boldsymbol{\alpha}} \quad (12)$$

In the current work, the pseudo-time derivative in the governing equations is approximated with the four stage SSP-RK3 scheme [8]. However, for the sake of simplicity, the mathematical formulation of the discrete optimisation problem and the derivation of the discrete adjoint Navier-Stokes equations are shown with first order forward difference formula. For higher order temporal schemes, the adjoint equations can be derived in a similar fashion.

From eqs. (9) and (12), it is clear that accurate computation of the shape sensitivities require the exact differentiation of J and G . Note that the primal fixed point iterator G consists of the discrete residuals due to convective and viscous fluxes, terms related to higher order reconstruction scheme with limiters and also the Runge-Kutta scheme. The exact differentiation of these terms by hand is laborious and prone to errors. Any approximation made by neglecting the differentiation of these terms will result in inaccurate computation of sensitivities [9]. One way to circumvent this difficulty is by employing Algorithmic Differentiation (AD) techniques. An advantage of this approach is that AD tools can perform the exact differentiation of these terms with much ease.

In this research, a discrete adjoint Navier-Stokes solver is developed by algorithmically differentiating the underlying primal solver. To ease the process of differentiation, the AD tool Tapenade [10] has been employed. Note that, at the first step, the differentiation is performed in a black-box fashion by coupling the subroutine that computes the geometric quantities such as cell volumes, surface areas and normals with the routines that evaluate the objective function and perform the primal fixed point iteration. An advantage of integrating these subroutines is that the resulting adjoint solver directly yields the shape sensitivities along with adjoints of the flow field variables. However, a major drawback of this approach is that the discrete adjoint code demands expensive memory and computational time. To explain this in detail, Algorithms (1) and (2) show the general structure of the primal Navier-Stokes solver and the corresponding black-box AD based discrete adjoint solver. From Algorithm (2), it is clear that the black-box approach stores the primal flow solutions U^n for all N iterations that yield a desired convergence. The stored solutions are then used to compute the adjoints in the reverse sweep. Obviously, for numerical simulations on finer grids with large values of N , the storage costs can become very expensive. One way to circumvent the storage of primal solutions is by employing the reverse accumulation technique [11]. This approach makes use of the iterative structure of the adjoint fixed point scheme in eq. (10). From this equation, it is clear that the adjoint solver requires only the converged primal solution U . The structure of an efficient adjoint solver based on this technique is shown in Algorithm (3). Here, M represents the number of pseudo-time iterations required for the convergence of adjoint solution. Note that the resulting adjoint solver is still consistent to the primal and computes the shape sensitivities as in eq. (12). Numerical investigations have shown that the run time of the optimised adjoint code is around a factor of 6 compared to the primal code. The performance of the discrete adjoint Navier-Stokes solver in accurate computation of shape sensitivities is demonstrated in numerical results.

Algorithm 1: Primal Navier-Stokes Solver

```

Compute geometric quantities
Initialize  $U^0$ 
for  $n \leftarrow 0$  to  $n < N$  do
    | perform the primal iteration  $U^{n+1} = G(U^n, \alpha)$ 
end
return Objective function  $J(U^N, \alpha)$ 

```

Algorithm 2: Black-box AD based adjoint Navier-Stokes solver

```
Compute geometric quantities
Initialize  $\mathbf{U}^0$ 
for  $n \leftarrow 0$  to  $n < N$  do
    | store ( $\mathbf{U}^n$ )
    | perform the primal iteration  $\mathbf{U}^{n+1} = G(\mathbf{U}^n, \alpha)$ 
end
compute the objective function  $J(\mathbf{U}^N, \alpha)$ 
Initialize  $\Psi^N$ 
for  $n \leftarrow N - 1$  to  $n \geq 0$  do
    | perform the adjoint iteration  $\Psi^n = \overline{G}(\Psi^{n+1}, \mathbf{U}^n, \alpha)$ 
    | retrieve ( $\mathbf{U}^n$ )
end
return Shape sensitivities  $\frac{dL}{d\alpha} = \frac{\partial J}{\partial \alpha} + \Psi^T \frac{\partial G}{\partial \alpha}$ 
```

Algorithm 3: Efficient adjoint Navier-Stokes solver

```
Compute geometric quantities
Initialize  $\mathbf{U}^0$ 
for  $n \leftarrow 0$  to  $n < N$  do
    | perform the primal iteration  $\mathbf{U}^{n+1} = G(\mathbf{U}^n, \alpha)$ 
end
compute the objective function  $J(\mathbf{U}^N, \alpha)$ 
Initialize  $\Psi^0$ 
for  $n \leftarrow 0$  to  $n < M$  do
    | perform the adjoint iteration  $\Psi^{n+1} = \overline{G}(\Psi^n, \mathbf{U}^N, \alpha)$ 
end
return Shape sensitivities  $\frac{dL}{d\alpha} = \frac{\partial J}{\partial \alpha} + \Psi^T \frac{\partial G}{\partial \alpha}$ 
```

III. Numerical Results

In this section, we present the numerical results to demonstrate the performance of the discrete adjoint Navier-Stokes solver in accurate computation of shape sensitivities. The test cases under investigation are: Laminar flow past NACA 0012 airfoil and analytical 3D body of revolution. The primal and adjoint laminar flow simulations are performed with the CFD codes UG3 and UG3A respectively [12]. UG3 is an unstructured higher-order accurate code for compressible flows, developed at the Centre for Applicable Mathematics, TIFR, Bengaluru. Its discrete version, UG3A is developed at the BITS Pilani, Hyderabad Campus. Since the adjoint code is generated by exactly differentiating the primal code, UG3A retains all features of UG3. The primal and adjoint solvers are parallelised using MPI with PETSc libraries.

A. Laminar flow past NACA 0012 airfoil

Transonic laminar flow over the NACA 0012 airfoil is computed with Mach number, $M = 0.8$, Reynolds number, $Re = 500$ and angle of attack, $AoA = 10^\circ$. The computational domain is a C-grid with 32,706 hexahedra. The airfoil boundary consists of 337 grid points. The flow results in a steady state solution with a small supersonic region just above the shoulder region and a separation zone covering considerable part on the suction side of the airfoil. Furthermore, it also leads to two recirculation zones near the trailing edge. The

computed lift and drag coefficients are shown in Table 1. These coefficients are in good agreement with the reference values in [13].

Code	Coefficient of lift (C_l)	Coefficient of Drag (C_d)
UG3	0.4328	0.2689
Swanson	0.4364	0.2750

Table 1 Laminar flow past NACA 0012 airfoil. Comparison of the lift and drag coefficients.

In order to verify the adjoint solver, the objective functions for the present case are defined as the lift and drag coefficients. The control variables are the x and y coordinates of the grid points on the airfoil. We then have 674 control variables.

Tables (2) and (3) show a comparison of sensitivities with respect to x and y coordinates at 3 points on the suction side of the airfoil. Note that point P_1 is located near the leading edge, P_2 near the mid chord and P_3 near the trailing edge of the airfoil. From these tables it can be observed that the shape sensitivities based on the discrete adjoint code are in excellent agreement with the values obtained from the second order finite differences with a step size of $\delta = 10^{-7}$. Furthermore, the adjoint sensitivities match even better with the values obtained from the tangent linear code. Note that the tangent linear or the so-called forward derivative code is generated by differentiating the primal laminar code in tangent mode. For large number of control variables, this approach is not feasible as the computational costs grow linearly. However, it is very useful in the development and verification of the adjoint code.

Figures 1 and 2 respectively show the vector plots of the sensitivity gradients of the lift and drag coefficients with respect to the shape variables. These plots show that for both the objective functions, maximum sensitivities are observed near the trailing edge of the airfoil. Apart from this region, significant sensitivities are noticed on the upper portion of the airfoil starting from the stagnation point till the region of attachment, which is about 0.36 chords. We can also notice that the sensitivities in the separation zone are either small or negligible. The sensitivity information can be used to perturb the shape in the descending direction of the gradient to minimise the drag coefficient.

Point	Objective function	Finite Differences	Tangent linear code	Discrete adjoint code
P_1	C_l	0.0058443281	0.005845318435566	0.005845318435892
	C_d	-0.0056312158	-0.005631378641497	-0.005631378643819
P_2	C_l	0.0059151730	0.005914341734872	0.005914341734252
	C_d	-0.0004255176	-0.000425123350878	-0.000425123350820
P_3	C_l	-0.0833677216	-0.083367330021174	-0.083367330021475
	C_d	0.0033024670	0.003302091528395	0.003302091520875

Table 2 Comparison of the sensitivities with respect to the shape variable x .

Point	Objective function	Finite Differences	Tangent linear code	Discrete adjoint code
P_1	C_l	0.0119872105	0.011981339551730	0.011981339551420
	C_d	0.0084285364	0.008425489132079	0.008425489132419
P_2	C_l	0.0130814907	0.013081373405311	0.013081373405304
	C_d	-0.0008615843	-0.000869588688824	-0.000869588688070
P_3	C_l	-0.5028925409	-0.50289753508449	-0.50289753508184
	C_d	-0.0364802861	-0.036487124781490	-0.036487124781374

Table 3 Comparison of the sensitivities with respect to the shape variable y .

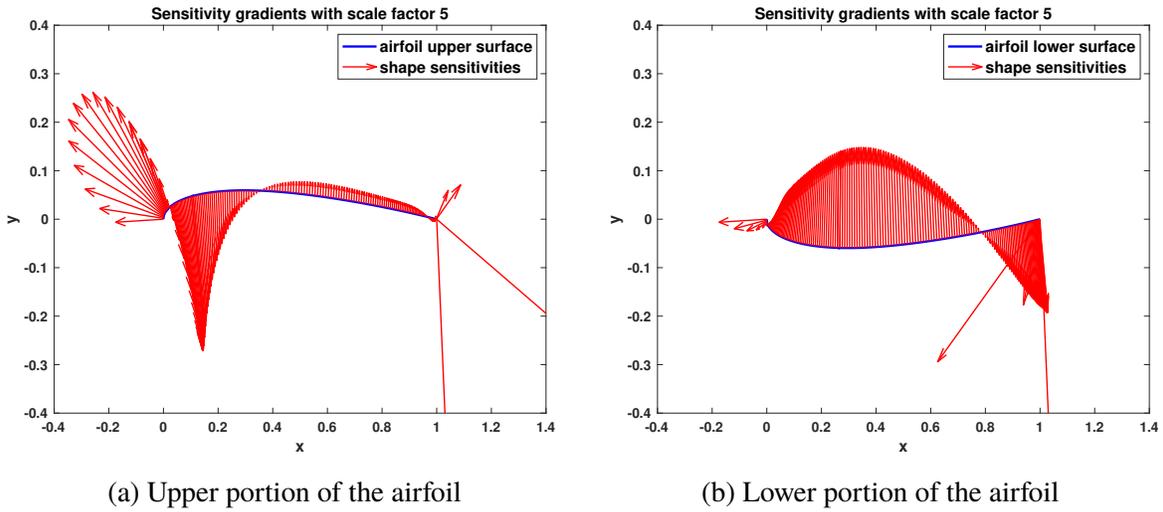


Fig. 1 Laminar flow past NACA 0012 airfoil. Sensitivities of the lift coefficient with respect to the shape variables.

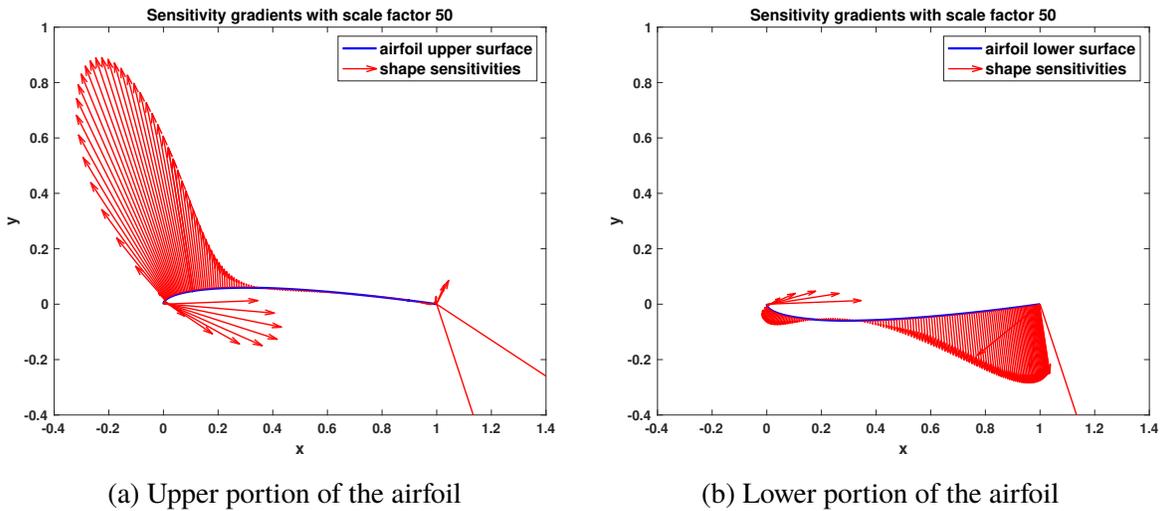


Fig. 2 Laminar flow past NACA 0012 airfoil. Sensitivities of the drag coefficient with respect to the shape variables.

B. Laminar flow past analytical 3D half-body of revolution

The next test case under investigation is the subsonic laminar flow past the analytical 3D half-body of revolution [14]. Numerical simulations are performed with Mach number, $M = 0.5$, Reynolds number, $Re = 5000$ and angle of attack, $AoA = 1^\circ$. The computational domain consists of around 3.6 million grid points. The surface of the body is resolved with 8,783 grid points. For this test case, the objective function is considered as the drag coefficient. One set of control variables are the x , y and z coordinates of the grid points that define the surface of the body of revolution. Apart from the shape variables, we are also interested in the sensitivities of the objective function with respect to the surface pressure distribution. This results in 35,132 control variables.

In order to demonstrate the accuracy of the adjoint solver on this test case, Table 4 shows a comparison of shape sensitivities at two grid points near the stagnation point. It can be observed that the sensitivities based on the adjoint solver are in very good agreement with the values obtained from direct methods.

Figure (3) shows the sensitivity gradients of the drag coefficient with respect to the shape variables x and y . Figure (4) shows the primal pressure contours and the adjoint pressure contours. Here, adjoint pressure implies the sensitivities of the drag coefficient with respect to the pressure distribution on the surface of the body. These plots clearly show the regions that significantly contribute to the defined objective function.

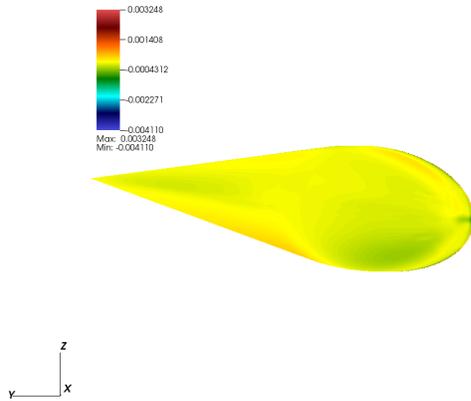
Point	Sensitivity	Finite Differences	Tangent linear code	Discrete adjoint code
P_1	$\frac{\partial C_d}{\partial x}$	-0.000111425821314	-0.000111713290849	-0.000111713252207
	$\frac{\partial C_d}{\partial y}$	0.001898371622554	0.001890016076619	0.001890016079188
	$\frac{\partial C_d}{\partial z}$	-0.000475473840471	-0.000470129023309	-0.000470129021143
P_2	$\frac{\partial C_d}{\partial x}$	0.000174027866295	0.000172374358211	0.000172374357936
	$\frac{\partial C_d}{\partial y}$	-0.001016720876650	-0.001016847747203	-0.001016847719231
	$\frac{\partial C_d}{\partial z}$	-0.000114856483818	-0.000114025201481	-0.000114025200578

Table 4 Comparison of the sensitivities with respect to the shape variables x , y , z .

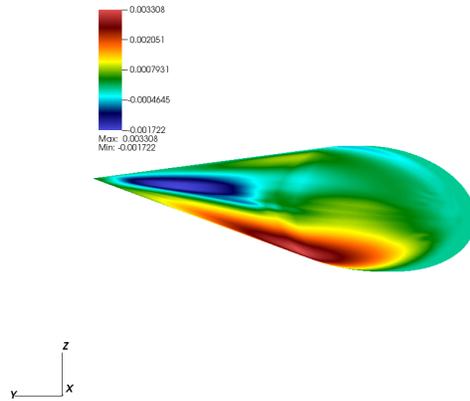
IV. Conclusions

This paper outlined the development of a discrete adjoint method for accurate computation of shape sensitivities in compressible laminar flows. The adjoint Navier-Stokes solver was developed by algorithmically differentiating the underlying primal solver. The performance of the adjoint solver in accurate computation of sensitivities was assessed by applying to the test cases of transonic flow past NACA 0012 airfoil and subsonic flow over the analytical half-body of revolution. Numerical results have shown that the adjoint solver is consistent to the primal as the shape sensitivities based on the adjoint code were in excellent agreement with the values obtained from finite differences and tangent linear code. This can be attributed to the exact differentiation of all terms in the primal solver.

Further research is under progress to extend the discrete adjoint solver from laminar to turbulent flows governed by the Reynolds-averaged Navier Stokes (RANS) equations. Efforts are also underway to extend the adjoint solvers to unsteady flows.

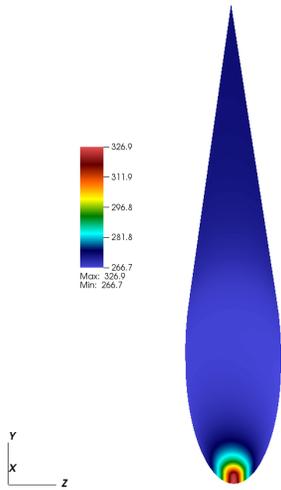


(a) Sensitivities w.r.t. the shape variable x

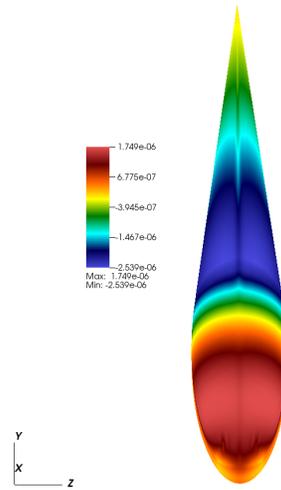


(b) Sensitivities w.r.t. the shape variable y

Fig. 3 Laminar flow past the half-body of revolution. Sensitivities of the drag coefficient with respect to the shape variables x and y .



(a) Pressure contours



(b) Adjoint pressure contours

Fig. 4 Laminar flow past the half-body of revolution. Sensitivities of the drag coefficient with respect to the surface pressure.

Acknowledgements

The authors gratefully acknowledge the financial support from the Science and Engineering Research Board, Department of Science and Technology, Government of India, under the project number EMR/2016/003182.

References

- [1] A. Jameson. Aerodynamic design via control theory. *J. Sci. Comput.*, 3:233–260, 1988.
- [2] J. Elliott and J. Peraire. Practical three-dimensional aerodynamic design and optimization using unstructured meshes. *AIAA Journal*, 35(9):1479–1485, 1997.
- [3] A. Carnarius, F. Thiele, E. Özkaya, A. Nemili, and N. Gauger. Optimal control of unsteady flows using a discrete and a continuous adjoint approach. *IFIP Advances in Information and Communication Technology*, 391:318–327, 2013.
- [4] A. Griewank and A. Walther. *Evaluating derivatives: Principles and techniques of Algorithmic Differentiation*. SIAM, 2008.
- [5] T. Albring, M. Sagebaum, and N. Gauger. A consistent and robust discrete adjoint solver for the Stanford University Unstructured (SU2) Framework - Validation and Application. *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, 132:77–86, 2016.
- [6] Junis Abdel Hay, E. Özkaya, N. Gauger, and Norbert Schönwald. Application of an adjoint CAA solver for design optimization of acoustic liners. *AIAA Paper 2016–2775*, 2016.
- [7] Anil Nemili, Emre Özkaya, Nicolas R. Gauger, Felix Kramer, and Frank Thiele. Accurate discrete adjoint approach for optimal active separation control. *AIAA Journal*, 55(9):3016–3026, 2017.
- [8] J. F. B. M. Kraaijevanger. Contractivity of runge-kutta methods. *BIT Numerical Mathematics*, 31(3):482–528, 1991.
- [9] Eric J. Nielsen and W. Kyle Anderson. Aerodynamic design optimization on unstructured meshes using the navier-stokes equations. *AIAA Journal*, 37(11):185–191, 1999.
- [10] L. Hascoët and V. Pascual. The Tapenade Automatic Differentiation tool: Principles, Model, and Specification. *ACM Transactions On Mathematical Software*, 39(3), 2013.
- [11] B. Christianson. Reverse accumulation of attractive fixed points. *Optimization Methods and Software*, 3:311–326, 1994.
- [12] C.S.Srikanth, Anil Nemili, Ashish Bhole, and Praveen Chandrashekar. An adjoint approach for accurate shape sensitivities in 3d compressible flows. In *8th Symposium on Applied Aerodynamics and Design of Aerospace Vehicles (SAROD)*, December 2018.
- [13] R. C. Swanson and S. Langer. Comparison of naca 0012 laminar flow solutions: Structured and unstructured grid methods. Technical report, NASA Langley Research Center, January 2016.
- [14] N. Kroll, H. Bieler, H. Deconinck, V. Couaillier, H. van der Ven, and K. Sorensen. ADIGMA – A European initiative on the development of adaptive higher-order variational methods for aerospace applications. *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, 113, 2010.