Sensitivity Methods in SU2

Continuous Adjoint, Discrete Adjoint, & Finite Difference Methods

Dr. Heather L. Kline

National Institute of Aerospace

August 9^{th}, 2019

Introduction

National Institute of Aerospace

SU2 provides multiple techniques to calculate sensitivities, or gradients with respect to design variables. They all use the same design variable definitions.

- Finite differences - perturb each variable in sequence and re-evaluate the output.
- Continuous adjoint - solve the discretized adjoint of the continuous problem and project sensitivities onto the variables.
- Discrete adjoint - solve the adjoint of the discretized problem and project sensitivities onto the variables.

This presentation will discuss each of these methods, including description of the shape deformation techniques. Practical application of these techniques will be covered in a following presentation.

Table of Contents

Table of Contents

NATIONAL INSTITUTE OF AEROSPACE

Finite Difference Gradient Calculations

The finite difference method is the most straightforward - the geometry is deformed, the solution is re-evaluated, and the difference of the object function values is divided by the step size.

Downsides:

- Requires $n+1$ function evaluations for n design variables - a cost that becomes prohibitive for large numbers of design variables.
- Accuracy depends on the step size; too large and it will not capture the local gradient, too small and numerical error will effect the gradient accuracy.

Benefits:

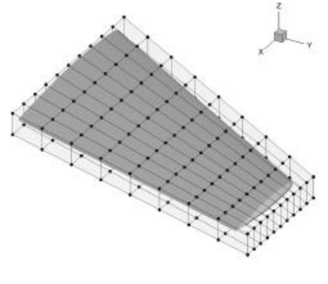
- Unlimited number of objective functions can be evaluated simultaneously.
- Simplicity - no additional derivations or automatic differentiation needed when addressing a new function.

Shape Deformations

A Free-Form Deformation (FFD) box technique is used to achieve smooth shape deformations (Samareh 2004). An initial box surrounding the object to be redesigned is parameterized as a Bézier solid, parameterized by Bernstein polynomials B^i:

$$X(u,v,w) = \sum_{i,j,k=0}^{l,m,n} P_{i,j,k} B_j^l(u) B_j^m(v) B_k^n(w),$$

where l, m, and n are the orders of the Bernstein polynomials, with one polynomial needed for each of the three dimensions. The control point indices are i, j, and k.



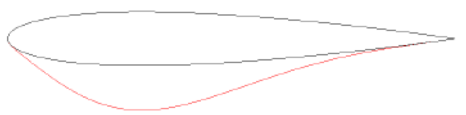
Onera M6 wing with FFD box from the SU2 tutorial on constrained shape design of a transonic inviscid wing at https://su2code.github.io/

Hicks-Henne Bump Functions

NATIONAL INSTITUTE OF AEROSPACE

Hicks-Henne bump functions are use for two-dimensional shape deformations, particularly for airfoil shapes(Hicks and Henne 1978). Hicks-Henne functions are defined in terms of the maximum location x_n, and result in smooth functions with zero deformation at the end points which can be superimposed to produce more complex deformations.

$$f_n(x) = \sin^3(\pi x^{e_n}), \;\; e_n = \frac{\log(0.5)}{\log(x_n)}, \;\; x \in [0, 1],$$



NACA 0012 airfoil with a single Hicks-Henne bump on the lower surface deformed by 0.1 with bump centered at 0.3 of the airfoil chord length.

Table of Contents

NATIONAL INSTITUTE OF AEROSPACE

Adjoints

Both the discrete and continuous adjoint methods use calculus of variations-based techniques to compute the sensitivity of a single objective function with respect to infinitesimal changes to the flow solution. These sensitivities are then projected onto specific deformations to the geometry in a post-processing step.

- Execution cost independent of the # design variables.
- Derive new PDE for new functionals.
- Sensitivity of one objective at a time.

A.k.a.: Lagrange multipliers, co-state problem, or dual problem.

The Continuous & Discrete Adjoint Method: Literature Review

- Optimal control of PDE systems by (Lions 1971) and (Pironneau 1984).
- Developed for aerodynamic optimization by (Jameson 1988).
- (Castro et al. 2007) developed the continuous adjoint for unstructured grids using a surface formulation. (Palacios et al. 2013; Economon et al. 2016) implemented many of these capabilities in SU2.
- Prior to the advent of automatic differentiation tools, analysis by (Nadarajah and Jameson 2000) indicated that the discrete adjoint has both a higher memory requirement and more difficult to implement accurately as compared to the continuous adjoint, while better able to accurately match finite-difference based.
- Details of the discrete adjoint in SU2 using automatic differentiation tools is provided by (Gauger et al. 2007; Albring, Sagebaum, and Gauger 2015; Mader et al. 2008; Zhou et al. 2015).

The Adjoint Method

J: Function of interest. R: Governing equations.
U: State variables (ex: conservative variables).
S: Design variables/independent variables (ex: surface shape).

$$J(U,S) \qquad \delta J/\delta S = ?$$

$$R(U,S) = 0$$

The Adjoint Method

J: Function of interest. R: Governing equations.
U: State variables (ex: conservative variables).
S: Design variables/independent variables (ex: surface shape).

$$J(U,S) \qquad \delta J/\delta S = ?$$

$$R(U,S) = 0$$

$$\delta J = \frac{\partial J}{\partial U}\delta U + \frac{\partial J}{\partial S}\delta S$$

$$\delta R = 0 = \frac{\partial R}{\partial U}\delta U + \frac{\partial R}{\partial S}\delta S$$

The Adjoint Method

J: Function of interest. R: Governing equations.
U: State variables (ex: conservative variables).
S: Design variables/independent variables (ex: surface shape).

$$J(U,S) \qquad \qquad \delta J/\delta S = ?$$

$$R(U,S) = 0$$

$$\delta J = \frac{\partial J}{\partial U}\delta U + \frac{\partial J}{\partial S}\delta S$$

$$\delta R = 0 = \frac{\partial R}{\partial U}\delta U + \frac{\partial R}{\partial S}\delta S$$

$$\delta \mathcal{J} = \delta J - \psi \delta R = \delta U \left(\frac{\partial J}{\partial U} - \psi \frac{\partial R}{\partial U} \right) + \delta S \left(\frac{\partial J}{\partial S} - \psi \frac{\partial R}{\partial S} \right)$$

The Adjoint Method

J: Function of interest. R: Governing equations.
U: State variables (ex: conservative variables).
S: Design variables/independent variables (ex: surface shape).

$$J(U,S) \qquad \delta J/\delta S = ?$$

$$R(U,S) = 0$$

$$\delta J = \frac{\partial J}{\partial U}\delta U + \frac{\partial J}{\partial S}\delta S$$

$$\delta R = 0 = \frac{\partial R}{\partial U}\delta U + \frac{\partial R}{\partial S}\delta S$$

$$\delta \mathfrak{J} = \delta J - \psi \delta R = \delta U \left(\frac{\partial J}{\partial U} - \psi \frac{\partial R}{\partial U} \right) + \delta S \left(\frac{\partial J}{\partial S} - \psi \frac{\partial R}{\partial S} \right)$$

$$\text{choose } \psi \text{ s.t. } \left(\frac{\partial J}{\partial U} - \psi \frac{\partial R}{\partial U} \right) = 0$$

J: Function of interest. R: Governing equations.
U: State variables (ex: conservative variables).
S: Design variables/independent variables (ex: surface shape).

$$J(U,S) \qquad \delta J/\delta S = ?$$

$$R(U,S) = 0$$

$$\delta J = \frac{\partial J}{\partial U}\delta U + \frac{\partial J}{\partial S}\delta S$$

$$\delta R = 0 = \frac{\partial R}{\partial U}\delta U + \frac{\partial R}{\partial S}\delta S$$

$$\delta \mathfrak{J} = \delta J - \psi \delta R = \delta U \left(\frac{\partial J}{\partial U} - \psi \frac{\partial R}{\partial U} \right) + \delta S \left(\frac{\partial J}{\partial S} - \psi \frac{\partial R}{\partial S} \right)$$

$$\text{choose } \psi \text{ s.t. } \left(\frac{\partial J}{\partial U} - \psi \frac{\partial R}{\partial U} \right) = 0$$

$$\rightarrow \frac{\delta J}{\delta S} = \left(\frac{\partial J}{\partial S} - \psi \frac{\partial R}{\partial S} \right)$$

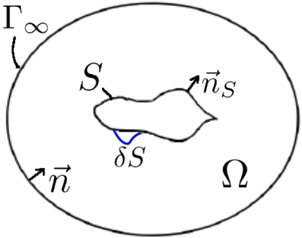
The Adjoint for Fluid Flow

$\mathcal{R}(U) = 0$ represents the Euler equations.

$$U = \left\{ \begin{array}{c} \rho \\ \rho\vec{v} \\ \rho E \end{array} \right\}, \Psi = \left\{ \begin{array}{c} \psi_\rho \\ \vec{\varphi} \\ \psi_{\rho E} \end{array} \right\}$$

$$\min_S J = \int_S j(U)ds$$

subject to: $\mathcal{R}(U) = 0$,



The Adjoint for Fluid Flow

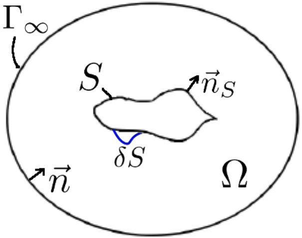
$\mathcal{R}(U) = 0$ represents the Euler equations.

$$U = \left\{ \begin{array}{c} \rho \\ \rho\vec{v} \\ \rho E \end{array} \right\}, \Psi = \left\{ \begin{array}{c} \psi_\rho \\ \vec{\varphi} \\ \psi_{\rho E} \end{array} \right\}$$

$$\min_S J = \int_S j(U)ds$$

subject to: $\mathcal{R}(U) = 0$,

$$\delta \mathcal{J} = \delta J - \int_\Omega \Psi^T \delta\mathcal{R}(U)d\Omega$$



The Adjoint for Fluid Flow

National Institute of Aerospace

$\mathcal{R}(U) = 0$ represents the Euler equations.

$$U = \left\{ \begin{array}{c} \rho \\ \rho\vec{v} \\ \rho E \end{array} \right\}, \Psi = \left\{ \begin{array}{c} \psi_\rho \\ \vec{\varphi} \\ \psi_{\rho E} \end{array} \right\}$$

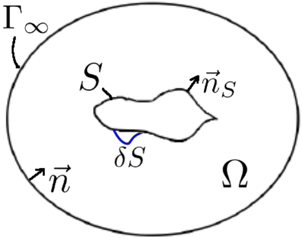
$$\min_S J = \int_S j(U) ds$$

subject to: $\mathcal{R}(U) = 0$,

$$\delta \mathcal{J} = \delta J - \int_\Omega \Psi^T \delta \mathcal{R}(U) d\Omega$$

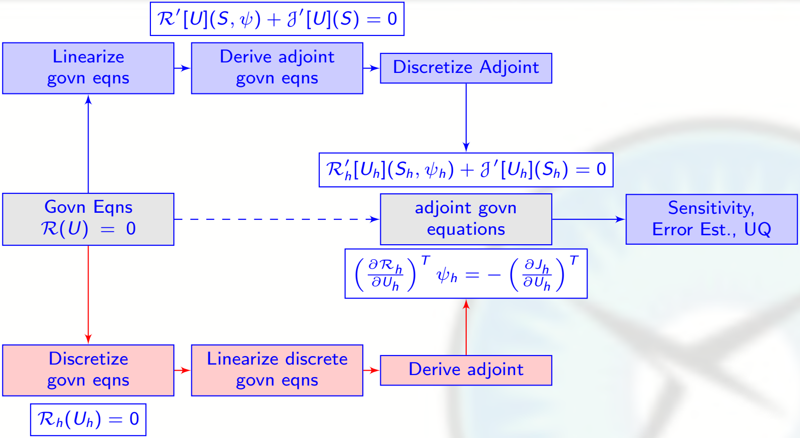
Find Ψ s.t. $\delta \mathcal{J}$ independent of all unknown δU.

$$\delta \mathcal{J} = \int_S \left(\frac{\partial J}{\partial S}(\Psi, U) \right) \delta S \, ds$$

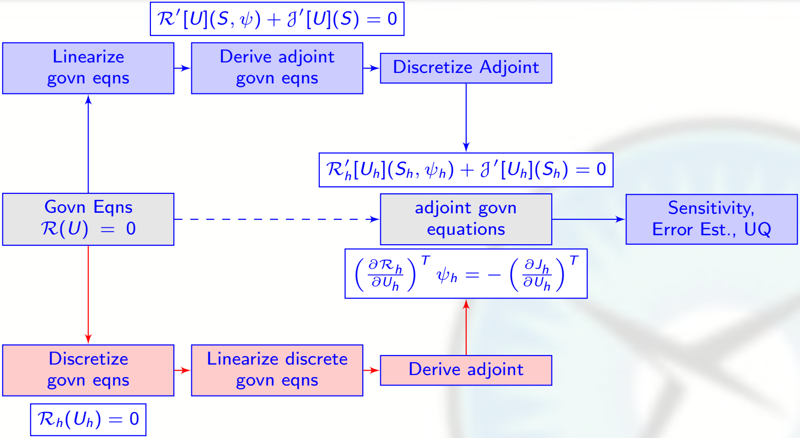


Continuous vs. Discrete Adjoint

NATIONAL INSTITUTE OF AEROSPACE



Continuous vs. Discrete Adjoint



Further detail of continuous adjoint surface formulation included in backup slides

Table of Contents

NATIONAL INSTITUTE OF AEROSPACE

Gradient Methods: Pros and Cons

Method	Costs	New Functionals	Accuracy
Finite Differences	Scales with the number of variables.	Just add the new output.	Depends on step size.
Continuous Adjoint	Scales with the number of functionals	Requires implementation of new boundary conditions.	Dependent on well-refined mesh, good implementation.
Discrete Adjoint	Higher memory cost relative to continuous adjoint.	Requires careful coding and recompilation for new functions.	Discretely consistent.

Table of Contents

References I

NATIONAL INSTITUTE OF AEROSPACE

Albring, T., M. Sagebaum, and N. R. Gauger (2015). "Development of a Consistent Dis-
crete Adjoint Solver in an Evolving Aerodynamic Design Framework". In: 16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference.

Boggs, P. and J. Tolle (1995). "Sequential Quadratic Programming". In: Acta numerica 4, pp. 1–51.

Castro, C. et al. (Sept. 2007). "Systematic Continuous Adjoint Approach to Viscous Aerodynamic Design on Unstructured Grids". In: AIAA Journal 45.9, pp. 2125–2139.

Economon, T. et al. (2016). "SU2 : An Open-Source Suite for Multiphysics Simulation and Design". In: AIAA Journal 54.3, pp. 828–846.

Gauger, N. R. et al. (2007). "Automatic Differentiation of an Entire Design Chain for Aerodynamic Shape Optimization". In: New Results in Numerical and Experimental Fluid Mechanics VI. Springer, pp. 454–461.

References II

- Hicks, R. M. and P. A. Henne (1978). "Wing design by numerical optimization". In: Journal of Aircraft 15.7, pp. 407–412.
- Jameson, A. (1988). "Aerodynamic Design via Control Theory". In: Journal of Scientific Computing 3.3, pp. 233–260.
- Lions, J. (1971). Optimal Control of Systems Governed by Partial Differential Equations. New York: Springer-Verlag.
- Mader, C. A. et al. (2008). "ADjoint: An Approach for the Rapid Development of Discrete Adjoint Solvers". In: AIAA Journal 46.4, pp. 863–873.
- Nadarajah, S. and A. Jameson (2000). "A Comparison of the Continuous and Discrete Adjoint Approach to Automatic Aerodynamic Optimization". In: AIAA Paper 2000-0667.
- Palacios, F. et al. (Jan. 2013). "Stanford University Unstructured (SU2): An open-source integrated computational environment for multi-physics simulation and design". In: 51st AIAA Aerospace Sciences Meeting and Exhibit. Grapevine, TX.

References III

- Pironneau, O. (1984). Optimal Shape Design for Elliptic Systems. New York: Springer-Verlag.
- Powell, M. (1978). “Algorithms for Nonlinear Constraints that use Lagrangian Functions”. In: Mathematical programming 14.1, pp. 224–248.
- Samareh, J. A. (2004). “Aerodynamic Shape Optimization Based on Free-Form Deformation”. In: AIAA Paper 2004-4630.
- Wilson, R. (1963). “A Simplicial Algorithm for Concave Programming”. PhD thesis. Harvard University.
- Zhou, B. et al. (2015). “A Discrete Adjoint Framework for Unsteady Aerodynamic and Aeroacoustic Optimization”. In: The 16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conferenc

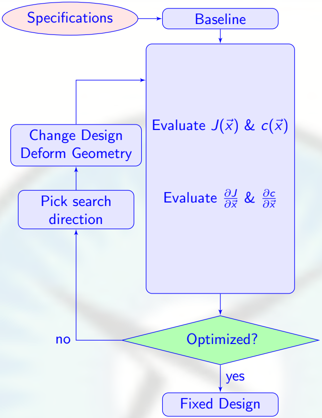
Questions?

Optimization

Non-Linear Program:

$$\begin{array}{lll} \text{minimize} & J(\vec{x}) & \\ \text{with respect to} & \vec{x} \in \mathbb{R}^n & \\ \text{subject to} & \hat{c}_j(\vec{x}) = 0, & j = 1, \ldots, \hat{m} \\ & c_k(\vec{x}) \geq 0, & k = 1, \ldots, m \end{array}$$

Optimization algorithms have been developed by (Powell 1978), (Wilson 1963), (Boggs and Tolle 1995) and others.



Continuous Adjoint Method Derivation

Expanding the Lagrangian: $\delta \mathcal{J} = \delta J - \int_{\Omega} \Psi^T \delta \mathcal{R}(U) d\Omega$, with the assumption that Γ_e is undeformed:

Continuous Adjoint Method Derivation

Expanding the Lagrangian: $\delta\mathcal{J} = \delta J - \int_\Omega \Psi^T \delta\mathcal{R}(U) d\Omega$, with the assumption that Γ_e is undeformed:

$$\delta J = \int_{\delta\Gamma_e} j(U)ds + \int_{\Gamma_e} \frac{\partial j}{\partial U}\delta U ds = \cancel{\int_{\Gamma_e'} j(U)ds} - \cancel{\int_{\Gamma_e} j(U)ds} + \int_{\Gamma_e} \frac{\partial j}{\partial U}\delta U ds = \int_{\Gamma_e} \frac{\partial j}{\partial U}\delta U ds$$

Continuous Adjoint Method Derivation

Expanding the Lagrangian: $\delta \mathcal{J} = \delta J - \int_\Omega \Psi^T \delta \mathcal{R}(U) d\Omega$, with the assumption that Γ_e is undeformed:

$$\delta J = \int_{\delta\Gamma_e} j(U)ds + \int_{\Gamma_e} \frac{\partial j}{\partial U}\delta U ds = \int_{\Gamma_e'} j(U)ds - \int_{\Gamma_e} j(U)ds + \int_{\Gamma_e} \frac{\partial j}{\partial U}\delta U ds = \int_{\Gamma_e} \frac{\partial j}{\partial U}\delta U ds$$

Applying the divergence theorem to the second term:

$$\int_\Omega \Psi^T \delta\mathcal{R}(U) d\Omega = \int_\Gamma \Psi^T \vec{A}\cdot\vec{n}\delta U ds + \int_S \Psi^T \vec{A}\cdot\vec{n}\delta U ds + \int_S \Psi^T \vec{A}\cdot\vec{n}\delta S ds - \int_\Omega \nabla\Psi^T \vec{A}\delta U d\Omega$$

Continuous Adjoint Method Derivation

Expanding the Lagrangian: $\delta\mathcal{J} = \delta J - \int_\Omega \Psi^T \delta\mathcal{R}(U)d\Omega$, with the assumption that Γ_e is undeformed:

$$\delta J = \int_{\delta\Gamma_e} j(U)ds + \int_{\Gamma_e} \frac{\partial j}{\partial U}\delta U ds = \cancel{\int_{\Gamma_e'} j(U)ds} - \cancel{\int_{\Gamma_e} j(U)ds} + \int_{\Gamma_e} \frac{\partial j}{\partial U}\delta U ds = \int_{\Gamma_e} \frac{\partial j}{\partial U}\delta U ds$$

Applying the divergence theorem to the second term:

$$\int_\Omega \Psi^T \delta\mathcal{R}(U)d\Omega = \int_\Gamma \Psi^T \vec{A}\cdot\vec{n}\delta U ds + \int_S \Psi^T \vec{A}\cdot\vec{n}\delta U ds + \int_S \Psi^T \vec{A}\cdot\vec{n}\delta S ds - \int_\Omega \nabla\Psi^T \vec{A}\delta U d\Omega$$

Combining the terms above:

$$\delta\mathcal{J} = \int_{\Gamma_e} \frac{\partial j}{\partial U}\delta U ds - \int_\Gamma \Psi^T \vec{A}\cdot\vec{n}\delta U ds - \int_S \Psi^T \vec{A}\cdot\vec{n}\delta U ds - \int_S \Psi^T \vec{A}\cdot\vec{n} U \delta S ds + \int_\Omega \nabla\Psi^T \cdot \vec{A}\delta U d\Omega$$

Continuous Adjoint Method Derivation

NATIONAL INSTITUTE OF AEROSPACE

Expanding the Lagrangian: $\delta \mathcal{J} = \delta J - \int_\Omega \Psi^T \delta \mathcal{R}(U) d\Omega$, with the assumption that Γ_e is undeformed:

$$\delta J = \int_{\delta\Gamma_e} j(U)ds + \int_{\Gamma_e} \frac{\partial j}{\partial U}\delta U ds = \cancel{\int_{\Gamma'_e} j(U)ds} - \cancel{\int_{\Gamma_e} j(U)ds} + \int_{\Gamma_e} \frac{\partial j}{\partial U}\delta U ds = \int_{\Gamma_e} \frac{\partial j}{\partial U}\delta U ds$$

Applying the divergence theorem to the second term:

$$\int_\Omega \Psi^T \delta\mathcal{R}(U) d\Omega = \int_\Gamma \Psi^T \vec{A}\cdot\vec{n}\delta U ds + \int_S \Psi^T \vec{A}\cdot\vec{n}\delta U ds + \int_S \Psi^T \vec{A}\cdot\vec{n}\delta S ds - \int_\Omega \nabla\Psi^T \vec{A}\delta U d\Omega$$

Combining the terms above:

Terms that lead to boundary conditions

$$\delta\mathcal{J} = \int_{\Gamma_e} \frac{\partial j}{\partial U}\delta U ds - \int_\Gamma \Psi^T \vec{A}\cdot\vec{n}\delta U ds - \int_S \Psi^T \vec{A}\cdot\vec{n}\delta U ds - \int_S \Psi^T \vec{A}\cdot\vec{n} U\delta S ds + \int_\Omega \nabla\Psi^T \cdot \vec{A}\delta U d\Omega$$

Continuous Adjoint Method Derivation

Expanding the Lagrangian: $\delta \mathcal{J} = \delta J - \int_\Omega \Psi^T \delta \mathcal{R}(U) d\Omega$, with the assumption that Γ_e is undeformed:

$$\delta J = \int_{\delta\Gamma_e} j(U) ds + \int_{\Gamma_e} \frac{\partial j}{\partial U} \delta U ds = \int_{\Gamma_e'} j(U) ds - \int_{\Gamma_e} j(U) ds + \int_{\Gamma_e} \frac{\partial j}{\partial U} \delta U ds = \int_{\Gamma_e} \frac{\partial j}{\partial U} \delta U ds$$

Applying the divergence theorem to the second term:

$$\int_\Omega \Psi^T \delta \mathcal{R}(U) d\Omega = \int_\Gamma \Psi^T \vec{A} \cdot \vec{n} \delta U ds + \int_S \Psi^T \vec{A} \cdot \vec{n} \delta U ds + \int_S \Psi^T \vec{A} \cdot \vec{n} \delta S ds - \int_\Omega \nabla \Psi^T \vec{A} \delta U d\Omega$$

Combining the terms above:

$$\delta \mathcal{J} = \int_{\Gamma_e} \frac{\partial j}{\partial U} \delta U ds - \int_\Gamma \Psi^T \vec{A} \cdot \vec{n} \delta U ds - \int_S \Psi^T \vec{A} \cdot \vec{n} \delta U ds - \int_S \Psi^T \vec{A} \cdot \vec{n} U \delta S ds + \int_\Omega \nabla \Psi^T \cdot \vec{A} \delta U d\Omega$$

Terms that lead to surface sensitivity

Continuous Adjoint Method Derivation

Expanding the Lagrangian: $\delta\mathcal{J} = \delta J - \int_\Omega \Psi^T \delta\mathcal{R}(U) d\Omega$, with the assumption that Γ_e is undeformed:

$$\delta J = \int_{\delta\Gamma_e} j(U) ds + \int_{\Gamma_e} \frac{\partial j}{\partial U}\delta U ds = \cancel{\int_{\Gamma_e'} j(U) ds} - \cancel{\int_{\Gamma_e} j(U) ds} + \int_{\Gamma_e} \frac{\partial j}{\partial U}\delta U ds = \int_{\Gamma_e} \frac{\partial j}{\partial U}\delta U ds$$

Applying the divergence theorem to the second term:

$$\int_\Omega \Psi^T \delta\mathcal{R}(U) d\Omega = \int_\Gamma \Psi^T \vec{A}\cdot\vec{n}\delta U ds + \int_S \Psi^T \vec{A}\cdot\vec{n}\delta U ds + \int_S \Psi^T \vec{A}\cdot\vec{n}\delta S ds - \int_\Omega \nabla\Psi^T \vec{A}\delta U d\Omega$$

Combining the terms above:

$$\delta\mathcal{J} = \int_{\Gamma_e} \frac{\partial j}{\partial U}\delta U ds - \int_\Gamma \Psi^T \vec{A}\cdot\vec{n}\delta U ds - \int_S \Psi^T \vec{A}\cdot\vec{n}\delta U ds - \int_S \Psi^T \vec{A}\cdot\vec{n} U \delta S ds + \boxed{\int_\Omega \nabla\Psi^T \cdot \vec{A}\delta U d\Omega}$$

Terms that lead to the adjoint governing equation