

Aerodynamic Optimisation Using CAD Parameterisations in SU2

Hewitt, P., Marques, S., & Robinson, T. T. (2014). Aerodynamic Optimisation Using CAD Parameterisations in SU2. Paper presented at 4th EASN Workshop on Flight Physics and Aircraft Design, Aachen, Germany.

Queen's University Belfast - Research Portal:

[Link to publication record in Queen's University Belfast Research Portal](#)

Publisher rights © the authors.

General rights

Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.

Aerodynamic Optimisation Using CAD Parameterisations in SU^2

4th EASN Flight Physics Workshop
Aachen, Germany

Philip Hewitt
phewitt04@qub.ac.uk

Simão Marques
s.marques@qub.ac.uk

Trevor Robinson
t.robinson@qub.ac.uk

School of Mechanical and Aerospace Engineering
Queen's University Belfast



Outline

Outline

1 Overview

2 Motivation

3 Gradient Calculation

4 Test Cases

5 Conclusions

Contents

- Overview
- Motivation
- SU^2
- CST Transformation
- Results
- Conclusions

Outline

1 Overview

2 Motivation

3 Gradient Calculation

4 Test Cases

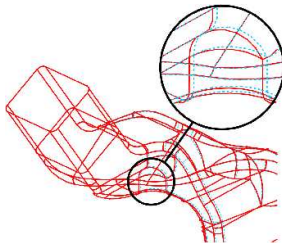
5 Conclusions

Motivation

- Increase flexibility of Adjoint Based Optimisation

Motivation

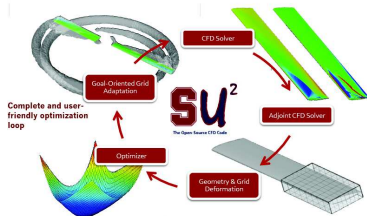
- Increase flexibility of Adjoint Based Optimisation
- Efficient calculation of any parametric sensitivity



SU²

SU² is an open-source CFD/Adjoint optimisation framework¹

- Developed at Stanford University

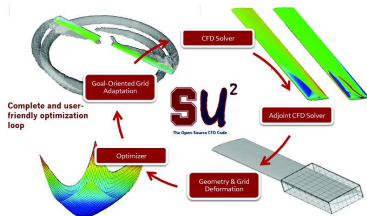


SU2
The Open-Source CFD Code

SU²

SU² is an open-source CFD/Adjoint optimisation framework¹

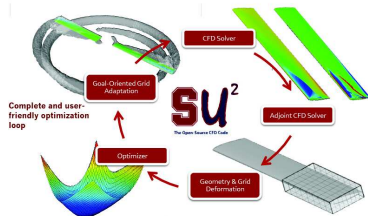
- Developed at Stanford University
- General purpose PDE solution methods



SU²

SU² is an open-source CFD/Adjoint optimisation framework¹

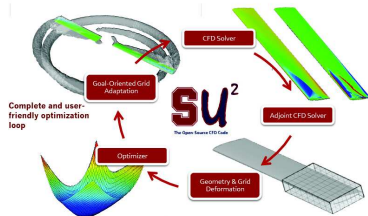
- Developed at Stanford University
- General purpose PDE solution methods
- Range of numerical schemes available (JST, ROE, MG, Euler-Implicit, . . .)



SU²

SU² is an open-source CFD/Adjoint optimisation framework¹

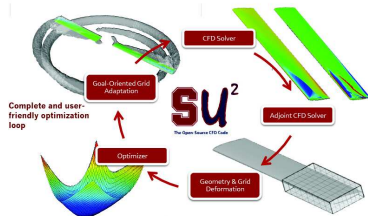
- Developed at Stanford University
- General purpose PDE solution methods
- Range of numerical schemes available (JST, ROE, MG, Euler-Implicit, . . .)
- Mesh deformation/adaptation



SU²

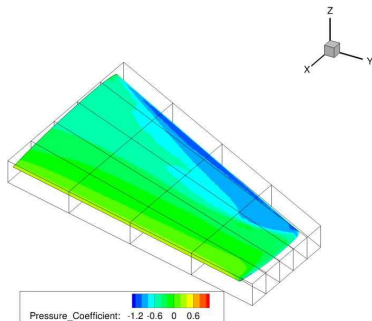
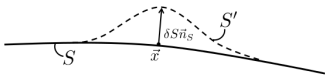
SU² is an open-source CFD/Adjoint optimisation framework¹

- Developed at Stanford University
- General purpose PDE solution methods
- Range of numerical schemes available (JST, ROE, MG, Euler-Implicit, . . .)
- Mesh deformation/adaptation
- **Continuous Adjoint Solver**



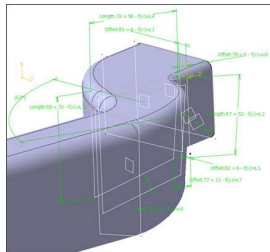
SU²

Native parameterisations in SU² 2



²images taken from <http://su2.stanford.edu/>

How to link any parameterisation to SU^2 ?



Outline

- 1 Overview
- 2 Motivation
- 3 Gradient Calculation**
- 4 Test Cases
- 5 Conclusions

Adjoint Based Optimisation

- Gradients required for optimisation
- Finite differences not feasible for complex shapes with multiple parameters
- Adjoint method provides an efficient alternative

Gradient Calculation

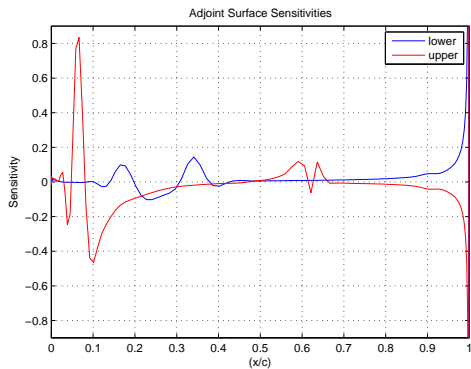
$$\underbrace{\begin{bmatrix} \frac{\partial f}{\partial A_1} \\ \frac{\partial f}{\partial A_2} \\ \vdots \\ \frac{\partial f}{\partial A_n} \end{bmatrix}}_{\text{Gradients}} = \underbrace{\begin{bmatrix} \frac{\partial x_1}{\partial A_1} & \cdots & \frac{\partial x_m}{\partial A_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_1}{\partial A_n} & \cdots & \frac{\partial x_m}{\partial A_n} \end{bmatrix}}_{\text{Geometric Sensitivities}} \underbrace{\begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_m} \end{bmatrix}}_{\text{Surface Sensitivities}}$$

- $\frac{\partial f}{\partial A_i}$ - Gradient
- $\frac{\partial x_j}{\partial A_i}$ - Geometric Sensitivities
- $\frac{\partial f}{\partial x_j}$ - Surface Sensitivities

Gradient Calculation

$$\underbrace{\begin{bmatrix} \frac{\partial f}{\partial A_1} \\ \frac{\partial f}{\partial A_2} \\ \vdots \\ \frac{\partial f}{\partial A_n} \end{bmatrix}}_{\text{Gradients}} = \underbrace{\begin{bmatrix} \frac{\partial x_1}{\partial A_1} & \cdots & \frac{\partial x_m}{\partial A_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_1}{\partial A_n} & \cdots & \frac{\partial x_m}{\partial A_n} \end{bmatrix}}_{\text{Geometric Sensitivities}} \underbrace{\begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_m} \end{bmatrix}}_{\text{Surface Sensitivities}}$$

- $\frac{\partial f}{\partial A_i}$ - Gradient
- $\frac{\partial x_j}{\partial A_i}$ - Geometric Sensitivities
- $\frac{\partial f}{\partial x_j}$ - Surface Sensitivities

SU^2 Computed by SU^2

Gradient Calculation

$$\underbrace{\begin{bmatrix} \frac{\partial f}{\partial A_1} \\ \frac{\partial f}{\partial A_2} \\ \vdots \\ \frac{\partial f}{\partial A_n} \end{bmatrix}}_{\text{Gradients}} = \underbrace{\begin{bmatrix} \frac{\partial x_1}{\partial A_1} & \dots & \frac{\partial x_m}{\partial A_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_1}{\partial A_n} & \dots & \frac{\partial x_m}{\partial A_n} \end{bmatrix}}_{\text{Geometric Sensitivities}} \underbrace{\begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_m} \end{bmatrix}}_{\text{Surface Sensitivities}}$$

- $\frac{\partial f}{\partial A_i}$ - Gradient
- $\frac{\partial x_j}{\partial A_i}$ - Geometric Sensitivities
- $\frac{\partial f}{\partial x_j}$ - Surface Sensitivities

SU^2

Gradient Calculation

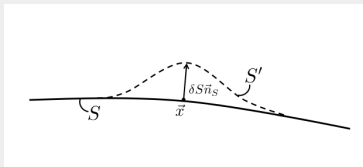
Use SU^2 native
parameterisations



SU^2

Gradient Calculation

Use SU^2 native
parameterisations



Gradient Calculation

Or add you own parametric
sensitivities:

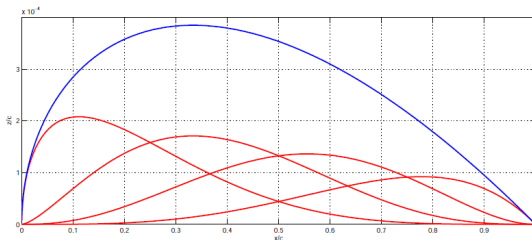
$$\Delta J = \int \phi V_n dS$$

CST Parameterisation

$$\zeta(\phi) = C_{N2}^{N1}(\phi)S(\phi) + \phi\Delta\zeta_{TE}$$

$$C_{N2}^{N1}(\phi) = \phi^{N1}(1-\phi)^{N2} \quad \text{Class Function}$$

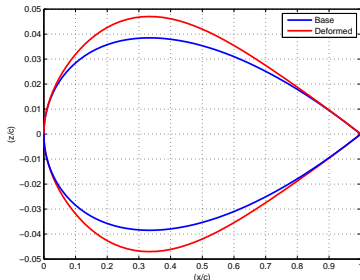
$$S(\phi) = \sum_{i=0}^n A_i S_i \quad \text{Shape Function}$$



CST Parameterisation

The surface is manipulated through the choice of function weights A_i :

$$S(\phi) = \sum_{i=0}^n A_i S_i$$

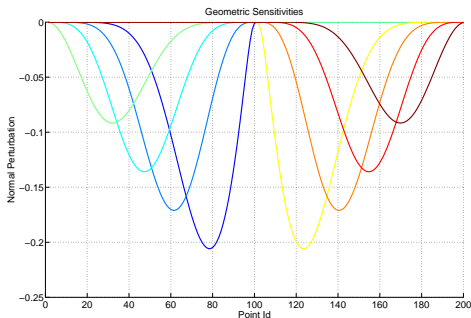


CST Parameterisation

Geometric Sensitivities ($\equiv V_n$)

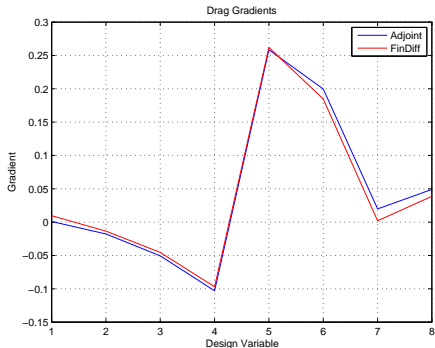
$\frac{\partial \mathbf{x}_j}{\partial A_i}$ – Geometric Sensitivities

$$\frac{\partial \mathbf{x}_j}{\partial A_i} = \left(\frac{\partial x_j}{\partial A_i} n_x + \frac{\partial y_j}{\partial A_i} n_y + \frac{\partial z_j}{\partial A_i} n_z \right)$$

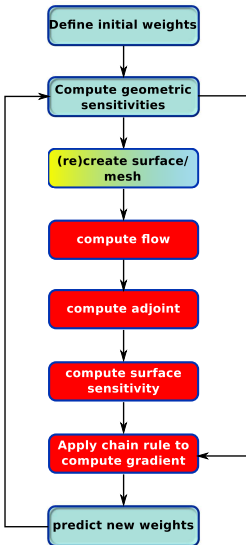


Validation - CST parameterisation

$$\frac{\partial f}{\partial A_i} - \text{Gradient}$$



SU^2 Optimisation Process



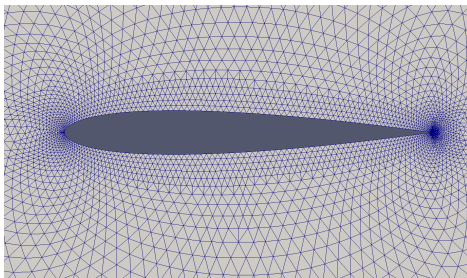
Outline

- 1 Overview
- 2 Motivation
- 3 Gradient Calculation
- 4 Test Cases**
- 5 Conclusions

NACA0012 Drag minimization

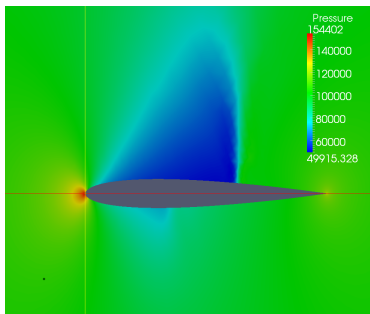
Inviscid test case: NACA0012 starting aerofoil.

- $M_\infty = 0.8$
- $\alpha = 1.25^\circ$
- $f = \min(C_d)$
- $C_l \geq 0.33$
- $C_m > 0.034$
- $nDV = 8$



NACA0012 Drag minimization

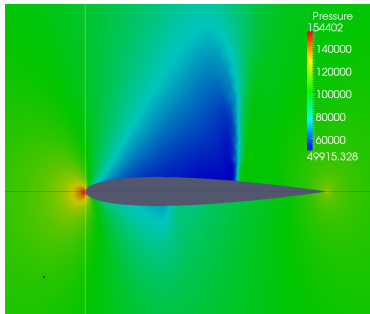
Inviscid test case: NACA0012 starting aerofoil.



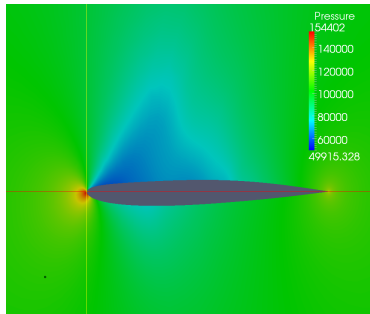
initial aerofoil

NACA0012 Drag minimization

Inviscid test case: NACA0012 starting aerofoil.



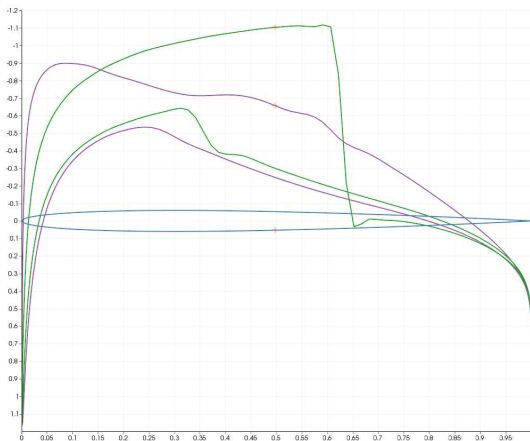
initial aerofoil



final aerofoil

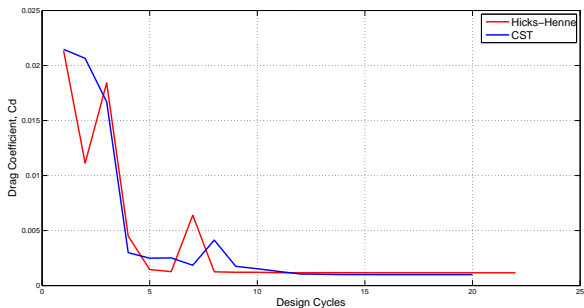
NACA0012 Drag minimization

Inviscid test case: NACA0012 starting aerofoil.



NACA0012 Drag minimization

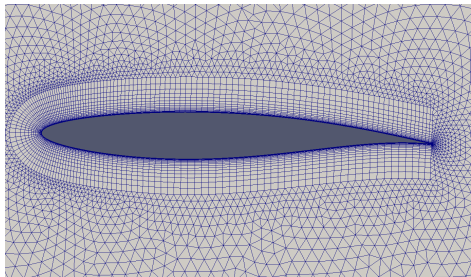
Inviscid test case: NACA0012 starting aerofoil.



RAE2822 Drag minimization

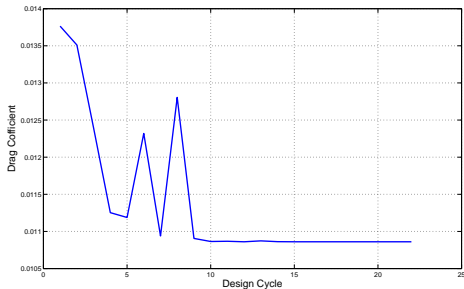
Viscous test case: RAE2822 starting aerofoil.

- $M_\infty = 0.729$
- $\alpha = 2.31^\circ$
- $f = \min(C_d)$
- $nDV = 8$
- SA turbulence model
- $y^+ \leq 5$



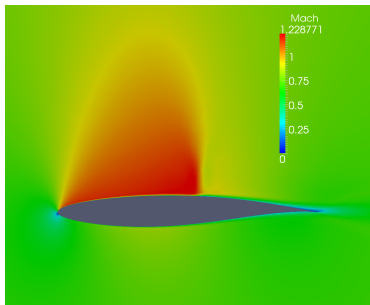
RAE2822 Drag minimization

Viscous test case: RAE2822 starting aerofoil.



RAE2822 Drag minimization

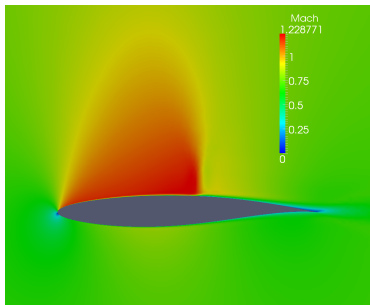
Viscous test case: RAE2822 starting aerofoil.



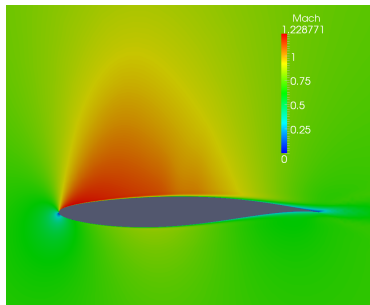
initial aerofoil

RAE2822 Drag minimization

Viscous test case: RAE2822 starting aerofoil.



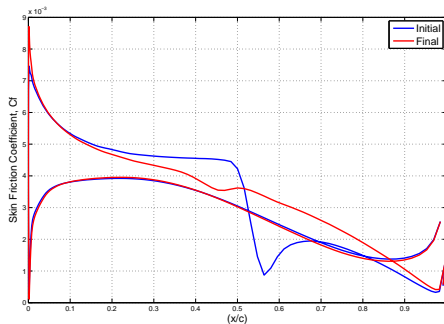
initial aerofoil



final aerofoil

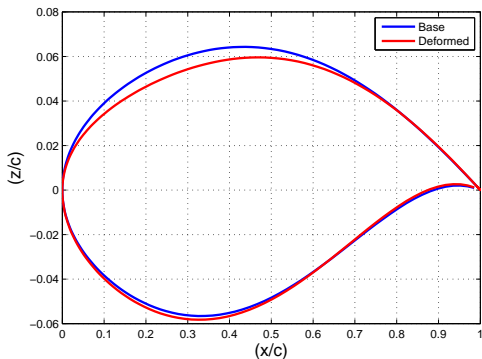
RAE2822 Drag minimization

Viscous test case: RAE2822 starting aerofoil.



RAE2822 Drag minimization

Viscous test case: RAE2822 starting aerofoil.



Outline

1 Overview

2 Motivation

3 Gradient Calculation

4 Test Cases

5 Conclusions

RAE2822 Drag minimization

- An alternative parameterisation was introduced into the SU^2 suite
- Model deformation can be performed outside SU^2
- alternative approach does not compromise optimisation efficiency with respect to native parameterisations

Q & A

Thank you for your attention

Questions Welcome

CST Parameterisation

$$\begin{aligned}\zeta(\phi) &= C_{N2}^{N1}(\phi)S(\phi) + \phi\Delta\zeta_{TE} \\ C_{N2}^{N1}(\phi) &= \phi^{N1}(1-\phi)^{N2} \\ S(\phi) &= \sum_{i=0}^n A_i S_i \\ S_i &= K_{i,n} \phi^i (1-\phi)^i \\ K_{i,n} &= \binom{n}{i} = \frac{n!}{i!(n-i)!}\end{aligned}$$