Structural and Multidisciplinary Optimization

Discrete Adjoint for Coupled Conjugate Heat Transfer

--Manuscript Draft--

Manuscript Number:	SAMO-D-22-00194R1
Full Title:	Discrete Adjoint for Coupled Conjugate Heat Transfer
Article Type:	Research Paper
Funding Information:	Horizon 2020 Framework Programme (660759) — Prof Tom Verstraete
Abstract:	The typical method to solve multi-physics problems such as Conjugate Heat Transfer (CHT) is the partitioned approach which couples separate solvers through boundary conditions. Effective gradient-based optimisation of partitioned CHT problems requires the adjoint of the coupling to maintain the efficiency of the original multi-physics coupling, which is a significant challenge. The use of automatic differentiation (AD) has the potential to ease this burden and leads to generic gradient computation methods. In this paper, we present how to automate the generation of adjoint fluid and solid solvers for a CHT adjoint using Automatic Differentiation (AD). The derivation of the adjoint of the loose coupling algorithms is shown for three fixed-point coupling algorithms. Application is shown to two CHT optimisation benchmark cases for inverse design and shape optimisation. The results demonstrate that Robin-based coupling algorithms have faster runtimes and are an attractive option for CHT optimisation problems.
Corresponding Author:	Jens-Dominik Mueller Queen Mary University of London UNITED KINGDOM
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Queen Mary University of London
Corresponding Author's Secondary Institution:	
First Author:	Oluwadamilare Imam-Lawal, PhD
First Author Secondary Information:	
Order of Authors:	Oluwadamilare Imam-Lawal, PhD
	Tom Verstraete, Dr
	Jens-Dominik Mueller
Order of Authors Secondary Information:	
Author Comments:	We think we have addressed all the revewiers' concerns.

SAMO-D-22-00194: DISCRETE ADJOINT FOR COUPLED CONJUGATE HEAT TRANSFER

IMAM-LAWAL, VERSTRAETE, MÜLLER

Dear Reviewers, thank you for your comments which were very helpful. Below are our responses and relevant changes to the text:

1. Reviewer 1

1.1. **p.6, The authors claim that $\tilde{\lambda}$ is a non-physical quantity which affects the speed of convergence and the stability of the "Temperature Forward Solid Coefficient Back" (TFRB) coupling method. In the referenced paper [13], also by the group of authors, no stability analysis has been carried out for this particular coupling. Should this be addressed in this paper? Presumably, one has to perform the same analysis as for the "Heat-transfer Coefficient Forward Temperature Back" method presented therein.**

1.1.1. *Response:* Because the scope of the paper is on optimization rather than on conjugate heat transfer, the authors fear a stability analysis will reduce the focus of the paper, and hence it is left out, but properly referenced in the text. A full stability analysis has been done in [1]

1.1.2. *Action:* Reference to [1] has been added to the text.

1.2. **p.10 The adjoint to the TFFB CHT coupling method, seems to be a "Flux Forward Temperature Back" (FFTB) scheme. According to the stability analysis presented in a previous work of the authors' group [13], the stability of the TFFB method is guaranteed for Biot numbers $|Bi| > 1$. Otherwise, the use of under-relaxation is needed in order to ensure convergence. However, in the same paper, stability analysis for the FFTB method yields that the scheme is stable if $|Bi| < 1$. Does this contradictory behavior in stability of the two methods in regards to Bi affect the stability of the discrete adjoint method?**

1.2.1. *Response:* The Biot number indeed affects the stability of the primal solver. Both the forward and reverse (adjoint) differentiation linearize the primal solver and inherit the linear stability of the primal solver. In other words, even though the adjoint simulation may have the look of a FFTB scheme, it is a linearizion of the primal and is stable if the primal is stable.

1.2.2. *Action:* none required

1.3. **p.10-13, In tables 3,5,7 the authors validate the sensitivities of dTw/dx, where x=[x,y,z] the coordinates of a node at the bottom surface of the flat plate, using the adjoint method against the two other methods (tangent and central-differences). Why is there a sensitivity with respect to (wrt) the perturbation of the node in the z direction? Is the case 2D or 3D?.**

1.3.1. *Response:* The case is 2.5D with only one cell in the z direction.

1.3.2. *Action:* The following sentence has been added to the first paragraph of Sec 3.1
"A 3D CHT simulation is performed to obtain the interface temperature and heat flux."

1.4. **p.13, The authors claim that, in Robin based coupling methods, by reducing the number of reverse-coupling iterations only 0.5% gradient accuracy is lost. Can the authors further expand this comment? Does this 0.5% reduction only refer to the flat plate validation case?**

1.4.1. *Response:* This refers to the flat plate validation case. In the results in the table below, 19 coupling iterations were required to converge the primal solution. Performing just 3 reverse coupling iterations as opposed to 19 results in a very similar gradient. This behaviour was observed only for the Robin-based coupling algorithms. More details are provided in [2].

Primal Its	Adjoint Its	$\frac{dT_w}{dT_b}$ [E-04]	$\frac{dT_w}{dx}$
19	19	2.593**1768445246677**	12.803**442391969877**
19	3	2.593**2320190671476**	12.803**770636795289**

TABLE 1. hFRB gradients with reverse coupling iterations.

1.4.2. *Action:* Paragraph 2 of Sec. 3.2. now reads
Furthermore, rather than performing an equal number of primal and reverse coupling iterations, fewer reverse coupling iterations could be performed, at the cost of a slight reduction in gradient accuracy. On the flat plate validation case in Sec. 3.1, for Robin-based coupling algorithms, reducing the number of reverse coupling iterations from 19 to 3 only resulted in a difference of approximately 0.5% in the gradients. This reduction in the number of reverse coupling iterations can also be combined with partial convergence of the fluid adjoint to significantly reduce runtime without great loss of gradient accuracy. However the TFFB algorithm always required the same number of reverse iterations as the primal to obtain accurate gradients.

1.5. **p.18, In table 9, the authors show that by using the TFRB coupling method, the optimization converges to the target solution with a much lower computational cost compared to the other two methods, despite the loss in gradient accuracy due to performing less reverse coupling iterations for the**

adjoint. This proves the time savings of this method. Any reason why the authors performed the optimization of the MarkII turbine blade using the hFRB method instead of TFRB?.

1.5.1. *Response:* The hFRB method was used because the TFRB method failed to converge. This was due to extreme interface values at the shocks on the suction side which lead to divergence of the TFRB method. It is possible that a better choice of Robin parameters could solve this issue, however, it is not straightforward to conduct a stability analysis on the irregular geometry and flow field of the blade.

1.5.2. *Action:* The following has been added to paragraph 1 of 4.2:
The CHT problem was solved using the hFRB algorithm due to stability reasons.

1.6. **p.19-20, Is the optimization of the MarkII blade based on mesh adaptation or re-meshing between optimization cycles? In either case, is it somehow guaranteed that conforming/matching meshes will be obtained at the Fluid-Solid interface, as shown in Fig.7(b)?**

1.6.1. *Response:* The mesh deformation algorithm used is Inverse Distance Weighted interpolation (IDW). Only the internal mesh nodes are displaced while the boundary nodes are kept fixed. Hence it is guaranteed that the meshes match at the interface

1.6.2. *Action:* The following has been added to paragraph 1 of 4.2.2:
"The Inverse Distance Weighted (IDW) interpolation method is used to propagate the displacement of the cooling channels to the internal solid mesh nodes. The interface boundary nodes are kept fixed to maintain the match with the fluid domain. The IDW algorithm is also reverse differentiated to obtain fully accurate adjoint gradients of the entire design chain."

1.7. **p.19 The results section seems rather poor. In specific, there is only one 2D application (the optimization of the MarkII turbine blade) which terminated prematurely due to fact that geometrical constraints for the positions of the cooling channels not to cross the boundary of the blade were not imposed. Even though this is not the subject of this paper, the authors are kindly asked to include these constraints in the optimization to make their results more realistic.**

1.7.1. *Response:* Point taken however the focus of this particular paper is the methodology and not the application. The authors are currently working on more challenging test cases which will be published in the near future. Within the timescales of this special issue, we will not be able to redefine and run the cases with additional constraints.

1.7.2. *Action:* Addressed in the forthcoming paper.

2. Reviewer 2

2.1. **Even though adjoint method is a computationally efficient method, it is used to compute sensitives for a gradient-based algorithm which is more prone to result in a local minimum than a gradient-free one. Therefore, I would recommend to add a brief discussion on this matter in the introduction.**

2.1.1. *Response:* The merits and downsides of gradient-based approaches are well known and are not the main subject of this paper.

2.1.2. *Action:* The following has been added to paragraph 3 of Sec. 1
"Although gradient-based approaches are only guaranteed to converge to local minima, they are preferred because they typically require less function evaluations. This is advantageous in applications like CHT where the cost of each function evaluation can be high."

2.2. **From my point of view, the introduction could be extended with more references on the topics covered.**

2.2.1. *Response:* There are a number of papers that summarise the literature of the topic, we have cited the main ones. To add a summary stability analysis to the manuscript, we have added Scholl et al, [1] to the introduction. In our view it would not be best use of the journal pages to repeat the overview.

2.2.2. *Action:* Reference to [1] has been added to the introductin.

2.3. **I suggest to provide more details about the partitioned approach followed. What kind of loosely-coupled scheme did you use? Are the coupling algorithms the loosely-coupled scheme? From my point of view, it is not clear from the text.**

2.3.1. *Response:* Yes the coupling algorithms are the loosely-coupled scheme.

2.3.2. *Action:* The following sentence has been added to the last paragraph above Sec 2.1
These separate solvers are loosely coupled to solve CHT problems.

2.4. **A reference to the Spalart-Allmaras turbulence model could be provided to the interested reader.**

2.4.1. *Response:* Thank you for pointing out this omission.

2.4.2. *Action:* Reference has been included

2.5. **Why do you have 226 for the inverse optimization problem? I suggest to include more details about the computational model. Did you perform a mesh dependency study?**

2.5.1. *Response:* The 226 design variables arise from the number of mesh nodes used to discretise the computational domain. A mesh dependence study using double the number of mesh points showed no significant change in results.

2.5.2. *Action:* Paragraph 3 of Sec 4.1 has been revised as follows:

Each mesh node at the bottom of the plate has an independent value of T_b specified as a boundary condition, and is used in this work as a design variable (α) that needs to be changed to drive J to zero. The selected mesh following a mesh dependence study results in 226 design variables for the present work.

2.6. **Why did you use the BFGS as optimization algorithm?**

2.6.1. *Response:* Any gradient-based optimisation algorithm e.g. steepest-descent, conjugate gradient, etc, could be used. The BFGS algorithm is known to have good performance and is the default method for the scipy.optimize.minimize library. It is widely used for unconstrained optimisation.

2.6.2. *Action:* none required.

2.7. **From my point of view, you should present more details about the computational model of the turbine blade.**

2.7.1. *Response:* Thank you for pointing this out.

2.7.2. *Action:* The following has been added to paragraph 2 of Sec. 4.2

Matching meshes are used for both domains as shown in Figure 7 with 49,532 nodes in the fluid domain and 5,714 nodes in the solid. A near wall spacing y^+ of less than 1 is used for the fluid.

2.8. **There is a typo on page 19, line 48, where it reads "form", it should read "from".**

2.8.1. *Response:* Thanks for spotting that, this has been changed in the text

2.8.2. *Action:* Changed in text

2.9. **How was the scaling process done for the design optimization problem?**

2.9.1. *Response:* The chord length of the turbine blade is only 0.06855 m. Therefore a unit displacement in any direction would take a cooling channel outside the solid domain. As we have no control over the line search and step size used in scipy these large displacements occur. When this happens, a high value of the objective function is returned to the optimiser to indicate that the step is too large.

2.9.2. *Action:* Revised the following sentence:

"The shallow slopes of the curves in Fig. 10 are due to the scaling performed to prevent the optimiser from taking large steps" to "The irregular shape of the curves in Fig. are a result of the SciPy line search."

2.10. **Please include the DOI information of the references.**

2.10.1. *Response:* Including the doi is not practised in the recently published papers in the journal.

2.10.2. *Action:* none required at this stage, if requested by the editors, can be added during proofreading.

References

[1] Sebastian Scholl, Bart Janssens, and Tom Verstraete. Stability of static conjugate heat transfer coupling approaches using robin interface conditions. *Computers & Fluids*, 172:209–225, August 2018.

[2] O.R. Imam-Lawal. *Adjoint based optimisation for coupled conjugate heat transfer*. PhD thesis, Queen Mary University of London, 2020.

Discrete Adjoint for Coupled Conjugate Heat Transfer

Oluwadamilare Imam-Lawal[1], Tom Verstraete[2†] and Jens-Dominik Müller[1*†]

[1*]School of Engineering and Materials Science, Queen MaryUniversity of London, Mile End Road, London, E1 4NS United Kingdom.
[2]Dept. of Turbomachinery, Von Karman Institute for Fluid Dynamics, Waterloose Steenweg, Sint Genesius Rode, Belgium.

*Corresponding author(s). E-mail(s): j.mueller@qmul.ac.uk; Contributing authors: dareIL@protonmail.com; tom.verstraete@vki.ac.be;
†These authors contributed equally to this work.

Abstract

The typical method to solve multi-physics problems such as Conjugate Heat Transfer (CHT) is the partitioned approach which couples separate solvers through boundary conditions. Effective gradient-based optimisation of partitioned CHT problems requires the adjoint of the coupling to maintain the efficiency of the original multi-physics coupling, which is a significant challenge. The use of automatic differentiation (AD) has the potential to ease this burden and leads to generic gradient computation methods. In this paper, we present how to automate the generation of adjoint fluid and solid solvers for a CHT adjoint using Automatic Differentiation (AD). The derivation of the adjoint of the loose coupling algorithms is shown for three fixed-point coupling algorithms. Application is shown to two CHT optimisation benchmark cases for inverse design and shape optimisation. The results demonstrate that Robin-based coupling algorithms have faster runtimes and are an attractive option for CHT optimisation problems.

Keywords: Gradient-based optimisation, adjoint method, conjugate heat transfer, Robin boundary conditions, inverse problems, shape optimisation

1 Introduction

Conjugate Heat Transfer (CHT) describes the process of heat transfer between a fluid and solid and is ubiquitous in engineering applications such as turbine blade cooling, modelling of heat exchangers, and cooling of electronics.

CHT problems may be solved using a monolithic approach in which both fluid and solid equations are solved simultaneously by a single solver. However, the partitioned or segregated approach is often adopted where separate solvers for the fluid and structure are loosely coupled through boundary conditions. These conditions need to be updated iteratively until the temperature and heat flux are continuous between the two domains [1, 2].

Recent work in CHT has evolved beyond merely solving the CHT problem to an increased interest in shape optimisation [3–6].This has lead to the need for efficient optimisation methods such as gradient-based approaches. *Although gradient-based approaches are only guaranteed to converge to local minima, they are preferred because they typically require less function evaluations. This is advantageous in applications like CHT where the cost of each function evaluation can be high.* Adjoint methods have been shown to be highly efficient as the cost of obtaining gradients can be made almost independent of the number of design variables. However, for partitioned coupling approaches, the flexibility of using different solvers for both domains results in an increased level of complexity with regard to obtaining the required gradients.

Adjoint methods can be grouped into continuous or discrete methods. In the continuous method the adjoint equations are analytically derived before being discretised while the discrete method discretises the state equations before formulating the discrete adjoint equations[7]. Arguments in favour of either approach revolve around stability, accuracy and computational effort. The discrete adjoint is considered to have the advantage with regards to stability and accuracy but not with computational effort. However, the use of automatic differentiation (AD) can significantly reduce the implementation effort associated with producing the adjoint code.

The majority of the work related to CHT optimisation has favoured the continuous adjoint formulation [3, 4, 6, 8] while Burghardt and Gauger [9] use the discrete adjoint but with a monolithic solver. In the present study, we demonstrate a partitioned/segregated coupling methodology in which the discrete adjoint formulation is achieved through AD. The use of AD has the potential to lead to very generic and less labour intensive implementations of adjoint methods for CHT. We focus on coupling boundary conditions and highlight the advantages Robin boundary conditions in the fluid domain. Furthermore, the gradient exchange required by partitioned coupling approaches is demonstrated using three fixed-point coupling algorithms.

This paper is organised as follows: we first describe the CHT problem, governing equations, and the coupling procedure. We then discuss the adjoint procedure for the partitioned approach and gradient verification. Two CHT optimisation problems are then solved using the adjoint method. Finally, a summary is given in the conclusion.

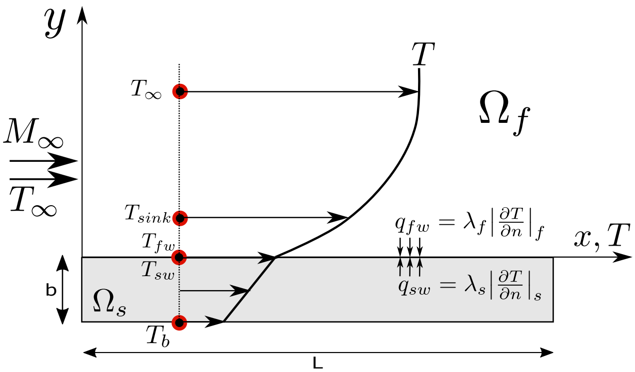


Fig. 1: Description of primal CHT problem

2 PRIMAL PROBLEM

Consider flow over a flat plate with finite thickness. The free stream flow temperature is T_∞, while the bottom of the plate is maintained at a lower temperature T_b. Consequently, heat is transferred at the interface between the solid plate and the fluid. The primal problem is to accurately compute the wall temperature at the interface between the fluid and solid, which is unknown a priori and can only be computed by considering the coupled problem (see Fig 1).

In order to solve the CHT problem using a partitioned approach, the fluid and solid governing equations must be coupled leading to a coupled system of equations

$$F(U^i, W^i) = 0, \tag{1}$$

$$S(U^i, W^{i+1}) = 0, \tag{2}$$

where U denotes the fluid state variables, W the solid state variables, and i the coupling iteration. F represents the Reynolds Averaged Navier-Stokes equations:

$$\frac{\partial}{\partial t}\int_{\Omega_f} \vec{U} d\Omega + \oint_{\partial\Omega_f} (\vec{f_c} - \vec{f_v}) \cdot dS = \int_{\Omega_f} \vec{Q} d\Omega_f, \tag{3}$$

where t denotes pseudo time, Q the source term, and x_j, $j = 1, 2, 3$ are the Cartesian coordinates. The state vectors U, and the inviscid and viscous flux vectors $\vec{f_c}$ and $\vec{f_v}$ are defined as

$$\vec{U} = \begin{bmatrix} \rho \\ \rho\vec{v} \\ \rho e \\ \tilde{\nu} \end{bmatrix}, \vec{f_c} = \begin{bmatrix} \rho\vec{v}\cdot\vec{n} \\ (\rho\vec{v}\vec{v} + p)\cdot\vec{n} \\ \rho(e+p)\vec{v}\cdot\vec{n} \\ \tilde{\nu}\vec{v}\cdot\vec{n} \end{bmatrix}, \vec{f_v} = \begin{bmatrix} 0 \\ \bar{\bar{\tau}}\cdot\vec{n} \\ \vec{\Theta}\cdot\vec{n} \\ \frac{1}{\sigma}(\nu_L + \tilde{\nu})(\nabla\nu\cdot\vec{n})\tilde{\nu} \end{bmatrix}, \tag{4}$$

$$\vec{\Theta} = \overline{\overline{\tau}} \cdot \vec{v} + \lambda \nabla T. \tag{5}$$

Where, ρ, p, and $\vec{v}$ are the fluid density, pressure, and velocity vector respectively, $\vec{n}$ the surface normal vector, e the internal energy per unit mass, τ is the stress tensor for Newtonian fluids and λ is the fluid thermal conductivity, $\tilde{\nu}$ is the modified eddy viscosity which is obtained using the Spalart-Allmaras turbulence model [10] and S in Eqn. (2) refers to the governing equation of the solid domain Ω_s, that is the steady state heat conduction equation

$$\lambda_s \nabla^2 T = 0, \tag{6}$$

where λ_s is the conductivity of the solid.

Each domain depends on the other through the boundary conditions specified at the fluid-solid interface. The solid state W, which is the solid's temperature, affects the fluid state U through the viscous flux component of the energy equation and through the heat flux at the non-adiabatic fluid-solid interface which must be specified using a boundary condition. Similarly, the fluid state variables U affect the state of the solid W through the interface boundary conditions specified while solving the heat equation.

Consequently, separate stand alone solvers for the fluid and solid, which are loosely coupled through interface boundary conditions, can be used to solve the primal problem. These boundary conditions are updated iteratively until the temperature and heat flux are continuous between the two domains. That is until

$$\begin{gathered} T_{sw} = T_{fw}, \\ \underbrace{\lambda_s \left.\frac{\partial T}{\partial n}\right|_s}_{q_{sw}} = \underbrace{\lambda_f \left.\frac{\partial T}{\partial n}\right|_f}_{q_{fw}}, \end{gathered}$$

where T_{fw} is the interface temperature in the fluid domain Ω_f, T_{sw} the interface temperature in the solid domain Ω_s, λ_f and λ_s the thermal conductivity of the fluid and solid respectively, n the surface normal, and q_{fw} and q_{sw} the interface heat flux in the fluid and solid domains respectively (see Figure 1).

In this work, the fluid equations are solved using the in-house mgOpt flow solver, a vertex centered, finite volume solver, which solves the 3-D compressible RANS equation using unstructured grids [11, 12]. The solid equations are solved using the finite element method with open-source solver, CalculiX, developed by Dhondt and Wittig. [13]. *These separate solvers are loosely coupled to solve CHT problems.*

2.1 Coupling Algorithms

In the present work, an implicit coupling method is used to solve the primal problem with a partitioned approach. Different coupling algorithms exist depending on which type of boundary conditions are exchanged between both domains. In both domains, boundary conditions could be specified as Dirichlet, Neumann, or Robin, leading to a total of 7 different types of coupling algorithms [14]. In this work, we use the three different fixed-point coupling algorithms described in the following sections.

2.1.1 Temperature Forward Flux Back (TFFB)

In the TFFB method [15], the solid interface heat flux distribution, q^i_{sw}, where i is the current coupling iteration, is imposed as a boundary condition to the fluid domain. The fluid solver $\mathbf{F}$ solves the flow equations resulting in a fluid interface temperature distribution, T^i_{fw}. This temperature is then imposed as boundary condition for the solid domain and the solid conduction solver, $\mathbf{S}$, provides an updated heat flux distribution q^{i+1}_{sw}. This loop is continued until there is no change in the boundary conditions exchanged by both solvers (see Figure 2),

$$T^i_{fw} = \mathbf{F}(q^i_{sw}), \tag{7}$$

$$q^{i+1}_{sw} = \mathbf{S}(T^i_{fw}). \tag{8}$$

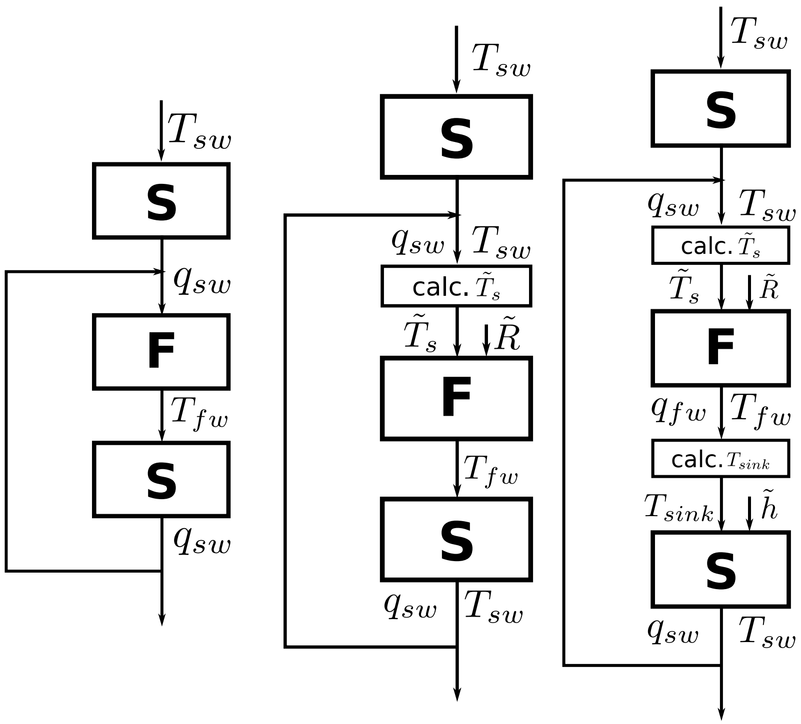


Fig. 2: CHT coupling algorithms. Left:TFFB, Centre: TFRB, Right:hFRB.

2.1.2 Temperature Forward Solid Coefficient Back (TFRB)

The TFRB coupling algorithm imposes a Dirichlet boundary condition in the solid domain and uses a Robin boundary condition in the fluid domain. The method also requires a virtual conductivity $\tilde{R}$ to be specified, with $\tilde{R} = \frac{\tilde{\lambda}}{L}$, $\tilde{\lambda}$ being the virtual solid conductivity, and L a solid length scale. The values of $\tilde{\lambda}$ and L are non-physical quantities chosen by the user which affect the speed of convergence and stability of the method [14, 16].

$$
\begin{aligned}
q^i_{sw} &= \mathbf{S}(T^i_{sw}), && (9)\\
\tilde{T}^i_s &= \frac{q^i_{sw}}{\tilde{R}} + T^i_{sw}, \implies (\text{calc. } \tilde{T}_s) && (10)\\
T^i_{fw} &= \mathbf{F}(\tilde{T}^i_s, \tilde{R}), && (11)\\
q^i_{fw} &= \tilde{R}(\tilde{T}^i_s - T^i_{fw}), \implies (\text{Robin BC in } \Omega_f) && (12)\\
T^{i+1}_{sw} &= T^i_{fw}. && (13)
\end{aligned}
$$

Assuming the algorithm begins in the solid domain, an initial guess for the wall temperature T^i_{sw} is imposed on the solid and used to obtain the heat flux q^i_{sw}. Next, the virtual solid sink temperature $\tilde{T}^i_s$ is calculated using Eqn. (10). The virtual conductivity and solid sink temperature are used for a Robin boundary condition in the fluid domain (Eqn. (12)). The flow solver then returns an update of the interface temperature which is given to the solid as a Dirichlet boundary condition (see Figure 2).

2.1.3 Heat Transfer Coefficient Forward Solid Coefficient Back (hFRB)

The hFRB method uses Robin boundary conditions in both domains and is the same as TFRB on the fluid side (see Figure 2). Similar to the TFRB method, the algorithm can start with an initial guess of the interface temperature on the solid side. Next, the virtual solid temperature $\tilde{T}_s$ is then calculated and passed to the flow solver along with the virtual conductivity $\tilde{R}$.

$$
\begin{aligned}
q^i_{sw} &= \mathbf{S}(T^i_{sw}), && (14)\\
\tilde{T}^i_s &= \frac{q^i_{sw}}{\tilde{R}} + T^i_{sw}, \implies (\text{calc. } \tilde{T}_s) && (15)\\
T^i_{fw}, q^i_{fw} &= \mathbf{F}(\tilde{T}^i_s, \tilde{R}), && (16)\\
T^i_{sink} &= T^i_{fw} - \frac{q^i_{fw}}{\tilde{h}}, \implies (\text{calc. } T_{sink}) && (17)\\
q^{i+1}_{sw}, T^{i+1}_{sw} &= \mathbf{S}(T^i_{sink}, \tilde{h}). && (18)
\end{aligned}
$$

The outputs from the flow solver are the interface temperature and heat flux (T_{fw}, q_{fw}). These are used in Eqn. (17) to calculate the ambient fluid temperature (T^i_{sink}) for a user specified value of the virtual heat transfer coefficient ($\tilde{h}$). The solid solver then returns an update on the interface temperature and heat flux and the exchange is continued until convergence [14, 16].

3 DISCRETE ADJOINT

For gradient-based CHT optimisation, we require the gradient of the objective function I w.r.t the design variables α

$$\frac{dI}{d\alpha} = \frac{\partial I}{\partial \alpha} + \begin{bmatrix} \frac{\partial I}{\partial U} & \frac{\partial I}{\partial W} \end{bmatrix} \begin{bmatrix} \frac{dU}{d\alpha} \\ \frac{dW}{d\alpha} \end{bmatrix}, \tag{19}$$

$$\frac{dI}{d\alpha} = \frac{\partial I}{\partial \alpha} + g^T u, \tag{20}$$

where U and W represent the fluid and solid state variables respectively. The term $g^T u$ is expensive to solve hence it is advantageous to use the adjoint formulation. The derivatives of the state variables for fluid and solid with respect to the design variables $\left(\frac{dU}{d\alpha}, \frac{dW}{d\alpha}\right)$ can be obtained from the state equations

$$F(\alpha, U, W) = 0, \tag{21}$$

$$S(\alpha, U, W) = 0. \tag{22}$$

Where F is used to represent the RANS equations, and S represents the heat equation. The derivative of the state equations with respect to the design variables which are required to solve Eqn. (19) are obtained from

$$\begin{bmatrix} \frac{\partial F}{\partial U} & \frac{\partial F}{\partial W} \\ \frac{\partial S}{\partial U} & \frac{\partial S}{\partial W} \end{bmatrix} \begin{bmatrix} \frac{dU}{\partial \alpha} \\ \frac{dW}{\partial \alpha} \end{bmatrix} = \begin{bmatrix} -\frac{\partial F}{d\alpha} \\ -\frac{\partial S}{d\alpha} \end{bmatrix}, \tag{23}$$

$$\mathbf{A}u = f.$$

The diagonal terms are the Jacobians of each discipline while the off-diagonal terms show how the states of one discipline affect the state of the other e.g. how the fluid temperature affects the solid flux and vice-versa. Therefore the sensitivity of the cost function, Eqn. (19), for the coupled problem can be written as

$$\frac{dI}{d\alpha} = \frac{\partial I}{\partial \alpha} + \begin{bmatrix} \frac{\partial I}{\partial U} & \frac{\partial I}{\partial W} \end{bmatrix} \begin{bmatrix} \frac{\partial F}{\partial U} & \frac{\partial F}{\partial W} \\ \frac{\partial S}{\partial U} & \frac{\partial S}{\partial W} \end{bmatrix}^{-1} \begin{bmatrix} -\frac{\partial F}{\partial \alpha} \\ -\frac{\partial S}{\partial \alpha} \end{bmatrix} \tag{24}$$

$$\frac{dI}{d\alpha} = \frac{\partial I}{\partial \alpha} + (g^T \mathbf{A}^{-1}) f. \tag{25}$$

The cost of calculating the gradient can be reduced through the adjoint method. The adjoint variables are the solution to the adjoint equation

$$\begin{bmatrix} \frac{\partial F}{\partial U} & \frac{\partial F}{\partial W} \\ \frac{\partial S}{\partial U} & \frac{\partial S}{\partial W} \end{bmatrix}^{T} \begin{bmatrix} \psi \\ \phi \end{bmatrix} = \begin{bmatrix} \frac{\partial I}{\partial U} \\ \frac{\partial I}{\partial W} \end{bmatrix}, \tag{26}$$

$$\mathbf{A}^T v = g.$$

Where $v^T = [\psi, \phi]$ represents the fluid and solid adjoint variables respectively and

$$v = (\mathbf{A}^T)^{-1} g, \tag{27}$$

$$v^T = g^T (\mathbf{A}^{-1}). \tag{28}$$

Therefore, after only one solve of Eqn. (26) for the adjoint variable v, the sensitivity can be calculated as

$$\frac{dI}{d\alpha} = \frac{\partial I}{\partial \alpha} + v^T f, \tag{29}$$

$$\frac{dI}{d\alpha} = \frac{\partial I}{\partial \alpha} + [\psi \ \phi] \begin{bmatrix} -\frac{\partial F}{\partial \alpha} \\ -\frac{\partial S}{\partial \alpha} \end{bmatrix}. \tag{30}$$

In the partitioned coupling approach, the Jacobian of the coupled system in Eqn. (26) is not calculated. Instead, the adjoint solution is obtained through an iterative approach, similar to the primal coupling, to compensate for the missing off-diagonals. The adjoint of system of each discipline is solved using the adjoint solution of the other discipline from the previous iteration [17]

$$\left(\frac{\partial F}{\partial U}^T\right) \psi^i = \left(\frac{\partial I}{\partial U}^T\right) - \left(\frac{\partial S}{\partial U}^T \phi^{i-1}\right) \tag{31}$$

$$\left(\frac{\partial S}{\partial W}^T\right)\phi^i = \left(\frac{\partial I}{\partial W}^T\right) - \left(\frac{\partial F}{\partial W}^T \psi^i\right), \tag{32}$$

where i is the current coupling iteration. The partial derivatives $(\frac{\partial S}{\partial U}, \frac{\partial F}{\partial W})$ in equations (31) and (32) depend on the type of coupling algorithm used (see Table 1) and are obtained by differentiating the solvers w.r.t. the coupling boundary conditions and coordinates. Reverse mode automatic differentiation is applied to the fluid and heat conduction solvers [18] using Tapenade [19], a source-transformation AD tool. The use of AD significantly reduces the effort required to obtain the adjoint solvers and in the case of the fluid solver, the procedure was fully automated. This allows for easy implementation of new coupling boundary conditions and optimisation objective functions.

	Coupling algorithm		
Partial derivative	TFFB	TFRB	hFRB
$\frac{\partial S}{\partial U}$	$\frac{\partial q_{sw}}{\partial T_{fw}}$	$\frac{\partial q_{sw}}{\partial T_{fw}}$	$\frac{\partial q_{sw}}{\partial T_{sink}}, \frac{\partial q_{sw}}{\partial h}, \frac{\partial T_{sw}}{\partial T_{sink}}, \frac{\partial T_{sw}}{\partial h}$
$\frac{\partial F}{\partial W}$	$\frac{\partial T_{fw}}{\partial q_{sw}}$	$\frac{\partial T_{fw}}{\partial \tilde{T}_s}$	$\frac{\partial T_{sink}}{\partial \tilde{T}_s}, \frac{\partial h}{\partial \tilde{T}_s}$

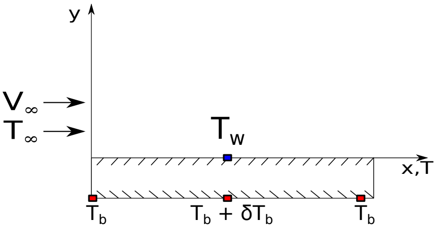
Table 1: Description of multidisciplinary partial derivative terms

The iterative approach shown in equations (31) and (32) results in a reversed/inverted version of each coupling algorithm. The total sensitivity in Eqn. (30) is then obtained by accumulating the gradients output after each solid adjoint iteration.

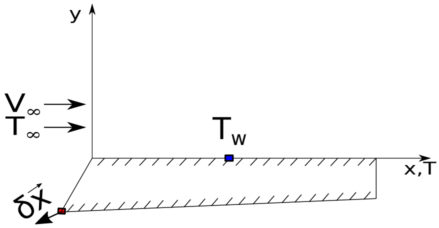
3.1 Gradient Calculation and Verification

We demonstrate accuracy of the partitioned adjoint methodology using a flat plate test case. The plate has a fixed temperature T_b at the bottom and comes in contact with fluid of a different temperature (see Fig 3). *A 3D CHT simulation is performed to obtain the interface temperature and heat flux* . A perturbation in the temperature of a red node at the bottom results in a change in the heat flux into the fluid domain. This perturbation travels through the coupling and results in a new interface temperature T_w at the blue node. Similarly, a coordinate perturbation (x, y, z) changes the volume of the plate and leads to a change in the heat flux at the fluid-solid interface and consequently alters the solution of the coupled problem. Therefore, the effect of these perturbations on the interface temperature is described by two gradients $\frac{dT_w}{dT_b}$ and $\frac{dT_w}{d\vec{x}}$ where $\vec{x} = [x, y, z]$.

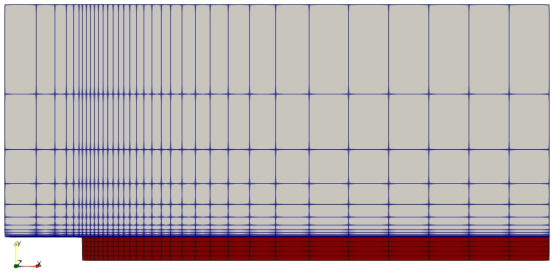
The partitioned adjoint approach to solving equations (31) and (32) is now described.



(a) Temperature perturbation



(b) $\vec{x}$ coordinate perturbation



(c) Cross-section of computational domain with matching Fluid and Solid meshes.

Fig. 3: Design variable perturbations for central difference

3.1.1 Temperature Forward Flux Back (TFFB)

The adjoint run of the TFFB algorithm, starts with seeding a vector $\overline{T}^i_{fw}$ which is set to 1 for the blue node in Figure 3 while all others are set to 0. $\overline{T}_{fw}$ is used by the adjoint flow solver ($\mathbf{F_b}$) to obtain the adjoint heat flux $\overline{q}_{sw}$. This is then passed directly to the adjoint solid ($\mathbf{S_b}$) solver to obtain an update of the adjoint temperature.

$$\overline{q}^i_{sw} = \mathbf{F_b}(\overline{T}^i_{fw}), \tag{33}$$

$$\overline{\vec{x}}^i, \overline{T}^i_b, \overline{T}^{i-1}_{fw} = \mathbf{S_b}(\overline{q}^i_{sw}). \tag{34}$$

The reverse loop is performed for the same number of iterations as for the primal solution and at the end of each coupling iteration, the gradient $\overline{T}_b$ is accumulated. This single adjoint solve obtains the gradient of the interface temperature at the blue node w.r.t to all design variables (red nodes). A block structure representation of the reverse differentiated coupling algorithm is shown in Figure 4. In fact, both $\frac{dT_w}{dT_b}$ and $\frac{dT_w}{d\vec{x}}$ are obtained in one reverse solve highlighting the advantage and cost savings of the adjoint method.

	Design variable		
Gradient method	1 [$\times 10^{-4}$]	2 [$\times 10^{-2}$]	3 [$\times 10^{-4}$]
Tangent	3.57424996556775**86**	1.09156391242752**44**	6.2298230582871**555**
Adjoint	3.57424996556775**91**	1.0915639124275**237**	6.2298230582871**479**
CD [$\Delta = 10^{-4}$ K]	3.5**665266295836773**	1.0**898488085331337**	6.2**618937590741552**

Table 2: TFFB gradient of interface temperature T_w w.r.t bottom temperature T_b ($\frac{dT_w}{dT_b}$)

Table 2 shows gradients for $\frac{dT_w}{dT_b}$ while Table 3 shows the gradients for $\frac{dT_w}{d\vec{x}}$. Both Tables show good agreement between the tangent, adjoint, and central difference (CD) methods.

	Design variable		
Gradient method	x	y	z
Tangent	16.3537546**63042668**	-13.270758**597756023**	0.410171858**55131066**
Adjoint	16.3537546**47042667**	-13.270758**610753868**	0.410171858**29135394**
CD [$\Delta = 10^{-7}$m]	16.3**21095586135925**	-13.2**44326737549272**	0.4**0858196825865889**

Table 3: TFFB gradient of interface temperature T_w w.r.t coordinates

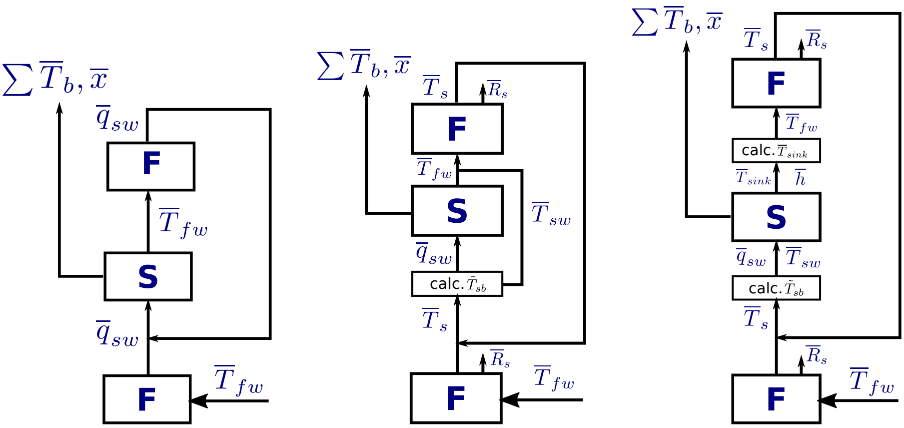


Fig. 4: Reverse differentiated CHT coupling algorithms; Left:TFFB, Centre: TFRB, Right:hFRB.

3.1.2 Temperature Forward Solid Coefficient Back (TFRB)

The adjoint TFRB run is started by seeding a vector of the adjoint fluid temperature $\overline{T}_{fw}$ to the flow solver to obtain $\tilde{T}_{sb}$. The reverse differentiated routine for Eqn. (10) is also required to obtain the adjoint heat flux and temperature as shown in equations (36). The solid solver then uses the adjoint heat flux to obtain $\overline{T}_{sw}$. The two adjoint interface temperatures are then summed to provide an update for the next iteration.

$$\tilde{T}^i_{sb} = \mathbf{F_b}(\overline{T}^i_{fw}), \tag{35}$$

$$\overline{q}^i_{sw}, \overline{T}_{sw} = f(\tilde{T}^i_{sb}, \tilde{R}) \implies (\text{calc. } \tilde{T}_{sb}), \tag{36}$$

$$\overline{\vec{x}}^i, \overline{T}^i_b, \overline{T}^i_{fw} = \mathbf{S_b}(\overline{q}^i_{sw}), \tag{37}$$

$$\overline{T}^{i-1}_{fw} = \overline{T}^i_{fw} + \overline{T}_{sw}. \tag{38}$$

The loop is done for the same number of coupling iterations and $\overline{T}_b$ and $\overline{\vec{x}}$ are accumulated. The final result returns $\frac{dT_w}{dT_b}$ and $\frac{dT_w}{d\vec{x}}$ and is shown in Tables 4 and 5. Good agreement is seen between the tangent, adjoint, and central difference (CD) methods.

	Design variable		
Gradient method	1 [$\times 10^{-4}$]	2 [$\times 10^{-2}$]	3 [$\times 10^{-4}$]
Tangent	3.5735550558001**070**	1.091430424038719**9**	6.2333366050062**219**
Adjoint	3.5735550558001**043**	1.091430424038719**6**	6.2333366050062**176**
CD [$\Delta = 10^{-4}$ K]	3.5**665607356349938**	1.0**898481832555262**	6.2**619392338092439**

Table 4: TFRB gradient of interface temperature T_w w.r.t bottom temperature T_b

	Design variable		
Gradient method	x	y	z
Tangent	16.370061**472907469**	-13.283993**573142387**	0.41096593**285022071**
Adjoint	16.370061**462907479**	-13.283993**581266039**	0.41096593**268774745**
CD [$\Delta = 10^{-7}$m]	16.3**21095586135925**	-13.2**44325032246707**	0.4**0858196825865889**

Table 5: TFRB gradient of interface temperature T_w w.r.t coordinates

3.1.3 Heat Transfer Coefficient Forward Solid Coefficient Back (hFRB)

For the hFRB adjoint, the adjoint wall temperature $\overline{T}_{fw}$ is used by the flow solver to obtain the adjoint solid sink temperature $\overline{T}_s$. This is then converted into an adjoint heat flux $\overline{q}_{sw}$ and adjoint temperature $\overline{T}_{sw}$ using the differentiated routine which calculates the Robin parameters T_{sink} and $\tilde{h}$ for the solid domain. The solid solver then returns an adjoint sink temperature $\overline{T}_{sink}$ and adjoint virtual heat transfer coefficient $\overline{h}$. The differentiated Robin preprocessing step uses the virtual heat transfer coefficient $\overline{h}$ to calculate the adjoint wall temperature $\overline{T}_{fw}$ while the adjoint sink temperature $\overline{T}_{sink}$ is assigned to the first off wall node, as shown in Eqn. (42).

$$\overline{T}^i_s, \overline{R}^i = \mathbf{F_b}(\overline{T}^i_{fw}) \tag{39}$$

$$\overline{q}^i_{sw}, \overline{T}^i_{sw} = f(R, \overline{T}^i_s) \implies (\text{calc. } \tilde{T}_{sb}) \tag{40}$$

$$\overline{\vec{x}}^i, \overline{T}^i_b, \overline{T}^i_{sink}, \bar{h}^i = \mathbf{S_b}(\overline{q}^i_{sw}, \overline{T}^i_{sw}), \tag{41}$$

$$\overline{T}^{i-1}_{fw}, \overline{T}^i_1 = f(\overline{T}^i_{sink}, \bar{h}^i) \implies (\text{calc. } \overline{T}_{sink}) \tag{42}$$

$$\overline{T}^{i-1}_s = \mathbf{F_b}(\overline{T}^{i-1}_{fw}, \overline{T}^{i-1}_1). \tag{43}$$

Tables 6 and 7 show good agreement between the gradients of $\frac{dT_w}{dT_b}$ and $\frac{dT_w}{d\vec{x}}$ obtained using the tangent, adjoint, and central difference (CD) methods.

	Design variable		
Gradient method	1 [$\times 10^{-4}$]	2 [$\times 10^{-2}$]	3 [$\times 10^{-4}$]
Tangent	2.57621104676445**55**	1.092538001954116**1**	6.26993861582025**40**
Adjoint	2.57621104676445**82**	1.092538001954116**4**	6.26993861582025**84**
CD [$\Delta = 10^{-4}$ K]	2.576**1096367205027**	1.092534**7169177257**	6.2**007757151150145**

Table 6: hFRB gradient of interface temperature T_w w.r.t bottom temperature T_b

	Design variable		
Gradient method	x	y	z
Tangent	12.77683411**8125210**	-10.5235973**06166273**	0.32383945**281109172**
Adjoint	12.77683410**8125197**	-10.5235973**14289930**	0.32383945**264861863**
CD [$\Delta = 10^{-7}$m]	12.7**60587537741230**	-10.5**09541539249767**	0.323**02864383382257**

Table 7: hFRB gradient of interface temperature T_w w.r.t coordinates

3.2 Advantage of fluid Robin-based coupling algorithms

All the gradients obtained in Tables 2 - 7 were obtained by fully converging the fluid and solid primal and adjoint equations as well as performing an equal number of primal and reverse coupling iterations. However, for all coupling algorithms, significant time savings can be obtained by only partially converging the fluid primal and adjoint equations each time without significantly impacting the accuracy of the gradients [18].

Furthermore, rather than performing an equal number of primal and reverse coupling iterations, fewer reverse coupling iterations could be performed, at the cost of a slight reduction in gradient accuracy. On the flat plate validation case in Sec. 3.1, for Robin-based coupling algorithms, reducing the number of reverse coupling iterations from 19 to 3 only resulted in a difference of approximately 0.5% in the gradients. This reduction in the number of reverse coupling *iterations can also be combined with partial convergence of the fluid adjoint to significantly reduce runtime without great loss of gradient accuracy. However the TFFB algorithm always required the same number of reverse iterations as the primal to obtain accurate gradients.*

Consequently, Robin-based coupling algorithms reduce the wall clock time required to obtain reasonably accurate gradients [18]. The Robin-based coupling algorithms have also been shown by Verstraete and Scholl [2, 14] to require less coupling iterations to converge the primal. Therefore, the time saving in both the primal and adjoint coupling runs will significantly speed up the optimisation process.

4 APPLICATION TO OPTIMISATION PROBLEMS

4.1 Inverse Design Optimisation

To demonstrate the efficacy of the partitioned adjoint methodology, we start with a simple inverse problem for which the final solution is known. Inverse problems are solved by providing a desired solution and adjusting design variables in order to achieve the desired target. In this problem, we seek the bottom temperature (T_b) which results in the best match for a given the interface wall temperature (T_{target}) as shown in Figure 5. Table 8 and Figure 1 show the remaining parameters.

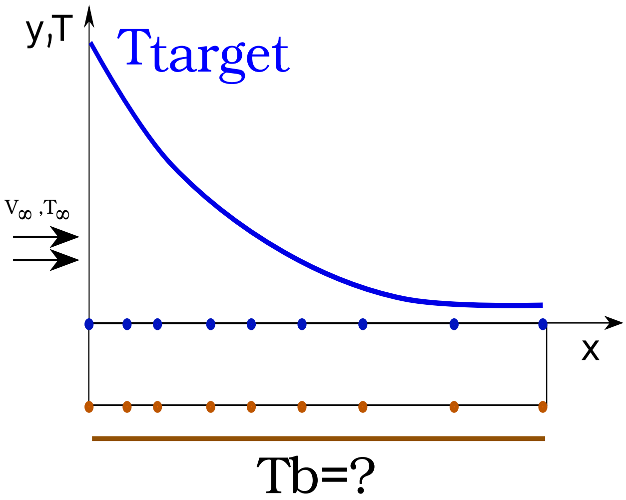


Fig. 5: Description of flat plate inverse problem

The inverse problem is solved by formulating an optimisation problem, which allows the use of classical direct methods to solve the physics involved. An objective function (J) is defined as the difference between the desired

Parameter	Value	Units
b	0.01	m
L	0.2	m
M_∞	0.1	
P_∞	$1.03 \cdot 10^5$	Pa
T_∞	1000	K
T_b	600	K
λ_s	0.2222	W/mK
λ_f	0.05568	W/mK
μ	$3.95 \cdot 10^{-5}$	Pa · s
Re_L	$1.132 \cdot 10^5$	
Pr	0.713	

Table 8: Table of parameters

interface temperature (T_{target}) and the obtained interface temperature (T_w) for an estimated bottom temperature ($\tilde{T}_b$).

$$J = \frac{1}{2}\int_0^L (T_{target} - T_w)^2 dx, \tag{44}$$

and minimising Eqn. (44) results in an interface temperature which matches the target temperature. The objective function depends implicitly on the bottom temperature T_b through the solution of the coupled problem. *Each mesh node at the bottom of the plate has an independent value of T_b specified as a boundary condition, and is used in this work as a design variable (α) that needs to be changed to drive J to zero. The selected mesh following a mesh dependence study results in 226 design variables for the present work.*

The objective is hence to minimise Eqn. (44) subject to the constraints of satisfying both the state equations of both domains and maintain continuity of state variables (T_w, q_w) across the interface.

A gradient based method is used to reduce the deviation of the current interface wall temperature with the desired one. The gradient of the objective function (sensitivity) w.r.t the control variables, α, is given as

$$\frac{dJ}{d\alpha} = \int_0^L (T_{target} - T_w)\frac{dT_w}{d\alpha} dx. \tag{45}$$

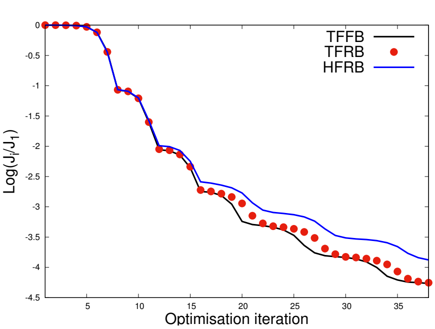
The gradients of the temperature w.r.t the design variables, i.e. the temperature specified on the bottom of the flat plate, are computed using the adjoint approach for calculating $\frac{dT_w}{dT_b}$ described in Sec. 3. The target temperature is obtained by solving the primal problem with a bottom temperature of 600K, hence it is guaranteed that a solution to the problem exists. $\tilde{T}_b$ is initially taken as 400K and refers to the estimated bottom temperature that should yield T_{target} .

4.1.1 Results

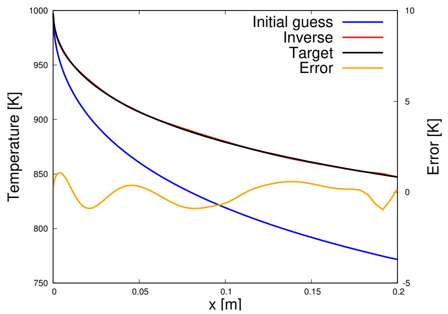
The objective is minimised using the BFGS optimisation algorithm from the SciPy library [20]. The difference between the temperatures obtained from the direct and inverse solution is defined as

$$\text{Error} = T_{\text{Target}} - T_{\text{Inverse}}. \tag{46}$$

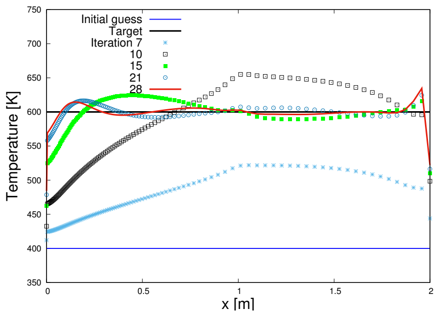
Figure 6a shows that the reduction in the objective function for all three coupling algorithms. The results for the TFRB algorithm in Fig. 6b show that the inverse solution is significantly closer to the target than the initial guess. Similar results were obtained for the TFFB and hFRB coupling algorithms (omitted for clarity). The successful solution of the inverse problem shows that the partitioned discrete adjoint methodology described is effective for solving CHT optimisation problems.



(a) Evolution of the objective function



(b) T_w distribution after 28 optimisation iterations (TFRB)



(c) Evolution of the bottom temperature (TFRB)

Fig. 6: Inverse solution after 28 optimisation iterations

Table 9 shows the time taken for the gradient norm to fall below 0.1 for all coupling algorithms. The Robin-based algorithms are seen to be faster and this is attributed to needing fewer primal and reverse coupling iterations. Despite the loss of gradient accuracy due to partially converging the fluid adjoint and performing $\approx$ 80% less reverse coupling iterations than primal coupling iterations, the Robin-based algorithms solve the inverse problem. This shows that the time savings are worth the sacrifice in gradient accuracy.

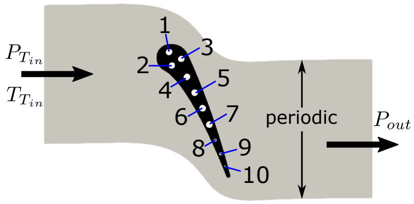
	Coupling method		
	TFFB	TFRB	HFRB
Time[hrs]	14.6	6.0	9.5
Iterations	30	30	38

Table 9: Comparison of optimisation runtimes for all coupling algorithms

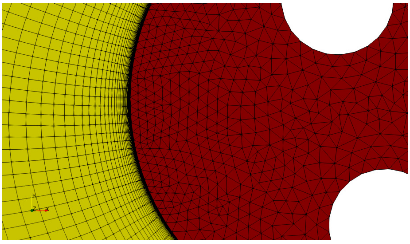
4.2 MarkII Turbine Blade

Modern gas turbines are equipped with internal cooling channels which cool the internal structure and prevent damage from high turbine inlet temperatures. Hence, an optimisation problem is formulated to minimise the maximum temperature in a turbine blade by changing only the location of each cooling channel.

The vane geometry is modelled after the Mark II turbine vane which has been investigated experimentally by Hylton et al. [21]. The blade is convectively cooled by ten cooling channels and a 2D simulation of the problem is carried out. *Matching meshes are used for both domains as shown in Figure 7 with 49,532 nodes in the fluid domain and 5,714 nodes in the solid. A near wall spacing* y^+ *of less than 1 is used for the fluid.*



(a) Vane geometry and boundary conditions



(b) Matching Fluid and Solid meshes

Fig. 7: MarkII turbine blade

In the fluid domain, boundary conditions for the freestream pressure and temperature, as well as the Mach number at inlet are specified, while a gauge pressure of zero is specified at the outlet. In the solid domain, a constant temperature is imposed in each of the cooling channels.

$P_{T_{in}}$[Pa]	$T_{T_{in}}$ [K]	P_{out}[Pa]	M_{in}	M_{out}
337100	788	167000	0.19	1.04

Table 10: MarkII run 5411 fluid boundary conditions

The blade is made of ASTM 310 stainless steel and the thermal conductivity is a function of temperature taken as

$$\lambda_s = 6.811 + 0.020176 \cdot T, \tag{47}$$

where T is the temperature at a point in the solid. The density and heat capacity are taken as 7900 kgm^{-3} and 586.5 $\mathrm{J\,kg^{-1}\,K^{-1}}$ respectively. We define an objective function (J) as the maximum temperature in the solid domain

$$J = \sqrt[p]{\frac{1}{\Omega_s}\int_{\Omega_s}\left(\frac{T}{T_{ref}}\right)^p d\Omega_s}, \tag{48}$$

where p is a user-defined integer taken as 10 currently and T_{ref} is a user-defined constant. The objective function depends on the solid temperature field which is obtained by solving the CHT problem. *The CHT problem was solved using the hFRB algorithm due to stability reasons.*

The coordinates (x, y) of each of the cooling channels are taken as control variables (α) that need to be changed to drive J to zero. As a result, the control variable α in the discrete problem is an array of size N. Where N is the number of channels (10), times the $x\&y$ coordinates leading to a total of 20. The gradients of the objective function w.r.t the design variables are computed using the partitioned adjoint approach described in Sec. 3.

4.2.1 Baseline Analysis

Figures 8 and 9 show the temperature, pressure, and heat transfer coefficient distributions on the baseline geometry. The computational surface temperature distribution deviates from the experiment near the leading edge but this discrepancy reduces closer to the trailing edge. The primary cause of the discrepancy on the suction surface of the leading edge is the due to the limitation of the Spalart-Allmaras turbulence model. This model is not well suited for modelling transitional flow and transitional models can be used to better match the experiment [22–24].

At the trailing edge where we see good agreement between the experiment and CHT results, the presence of the cooling channels leads to oscillations in the interface temperature and heat transfer coefficient distributions. We also see that the solid temperature is highest at the trailing edge.

4.2.2 Optimisation Results

An unconstrained optimisation is performed using the BFGS optimisation algorithm from the SciPy library [20]. *The Inverse Distance Weighted (IDW)*

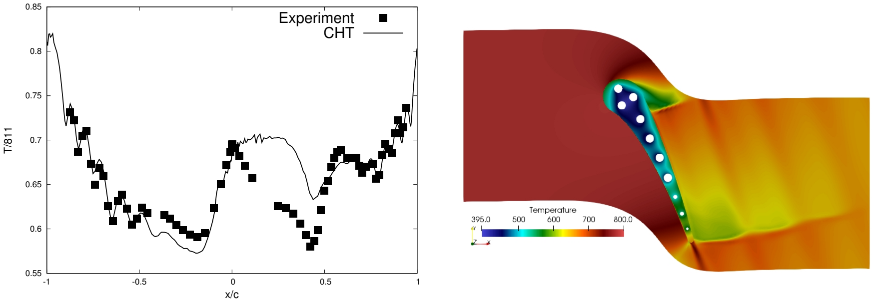


Fig. 8: Temperature distribution at fluid-solid interface and in the computational domain

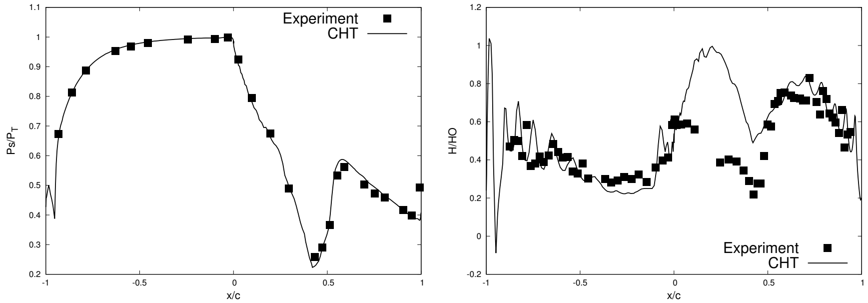
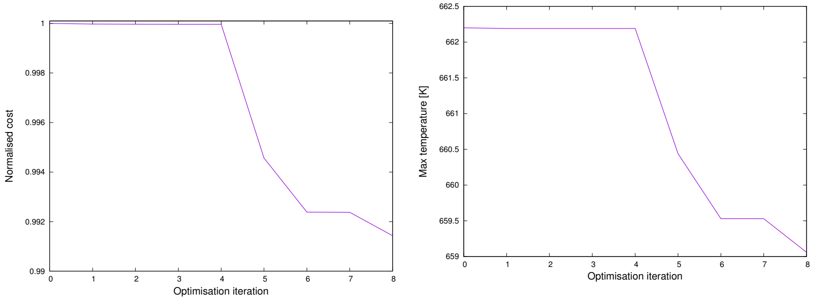


Fig. 9: Pressure and heat transfer coefficient distribution at the fluid-solid interface

interpolation method is used to propagate the displacement of the cooling channels to the internal solid mesh nodes. The interface boundary nodes are kept fixed to maintain the match with the fluid domain. The IDW algorithm is also reverse differentiated to obtain fully accurate adjoint gradients of the entire design chain.

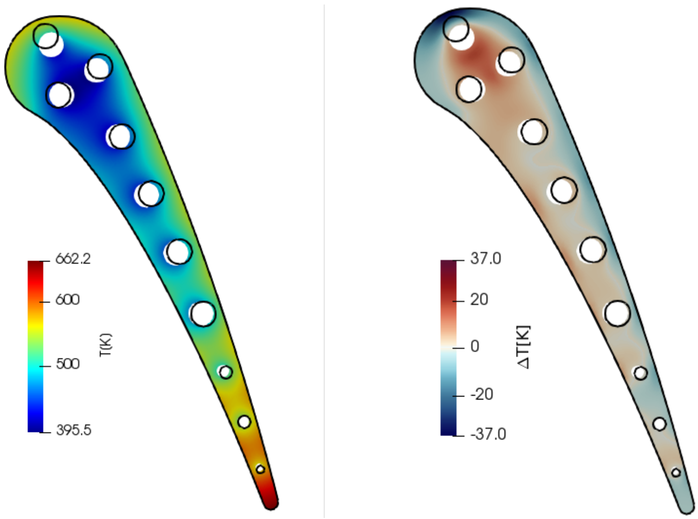
We achieve approximately 0.9% reduction in the objective function which corresponds to a 3.14 K drop in the maximum temperature as shown in Fig. 10. *The irregular shape of the curves in Fig. 10 are a result of the SciPy line search.*

Figure 11a shows the displacement of the cooling channels. The black lines represent the channels of the optimised blades while the solid surface is the initial geometry and temperature distribution. The optimisation is terminated prematurely as the first cooling channel (channel 1) moves out of the solid domain in the next optimisation step. The channels at the leading edge (channels 1-3) are displaced upwards and outwards towards the higher temperature regions, the trailing edge channels (channels 7-10) are displaced downwards, while the middle channels (channels 4-6) are displaced towards the suction surface.



(a) Evolution of objective function (b) Evolution of maximum temperature

Fig. 10: MarkII optimisation results



(a) Channel displacement (b) Temperature change

Fig. 11: MarkII optimisation results

Figure 11b shows the node based change in the blade temperature distribution plotted on the initial geometry, where ΔT is the difference between the optimised and baseline temperature. The greatest reduction in temperature is seen at the leading edge near the new location of channel 1.

5 CONCLUSION

A partitioned discrete adjoint method for the partitioned approach to Conjugate Heat Transfer modelling has been presented. Loose coupling of the fluid and solid domains with three fixed-point coupling algorithms was presented and their adjoints were derived. The discrete adjoint solvers were developed using reverse mode automatic differentiation (AD) which reduces the effort of development and in particular maintenance. The process of developing the fluid adjoint solver was fully automated which provides a strong case for adopting the discrete adjoint over the continuous adjoint for multidisciplinary optimisation problems. The exchange of sensitivity information between solvers was also described and the gradients verified. The presented methodology was then applied to two CHT optimisation problems involving a flat plate and a turbine blade. It was shown that Robin-based coupling algorithms have lower runtimes as they need fewer coupling iterations to converge the primal and adjoint. Consequently, partitioned methods with Robin-based coupling algorithms can be an attractive and competitive option for CHT optimisation in comparison to monolithic methods.

References

[1] Ganine, V., Hills, N.J., Lapworth, B.L.: Nonlinear acceleration of coupled fluid-structure transient thermal problems by anderson mixing. International Journal for Numerical Methods in Fluids **71**, 939–959 (2013)

[2] Verstraete, T., Scholl, S.: Stability analysis of partitioned methods for predicting conjugate heat transfer. International Journal of Heat and Mass Transfer **101**, 852–869 (2016)

[3] Mousavi, A., Nadarajah, S.K.: Adjoint-based multidisciplinary design optimization of cooled gas turbine blades. AIAA 2011-1131 (2011)

[4] Zeinalpour, M., Mazaheri, K., Kiani, K.C.: A coupled adjoint formulation for non-cooled and internally cooled turbine blade optimization. Applied Thermal Engineering **105**(327-335) (2016)

[5] Ferlauto, M.: An Inverse Method of Designing the Cooling Passages of Turbine Blades Based on the Heat Adjoint Equation. In: Proceedings of the Institution of Mechanical Engineers. Part A, Journal of Power and Energy, pp. 328–339 (2014)

[6] Gkaragkounis, K.T., Papoutsis-Kiachagias, E.M., Giannakoglou, K.C.: The continuous adjoint method for shape optimization in conjugate heat transfer problems with turbulent incompressible flows. Applied Thermal Engineering **140**, 351–362 (2018)

[7] Giles, M.B., Duta, M.C., Müller, J.-D.: Adjoint code developments using the exact discrete approach. AIAA-CP-2001-2596 (2001)

[8] Mousavi, S.A.: Constrained aerodynamic and heat transfer optimization

of gas turbine blades using an adjoint approach. PhD thesis, McGill University (August 2012)

[9] Burghardt, O., Gaugaer, N.R.: Accurate gradient computations for shape optimization via discrete adjoints in cfd-related multiphysics problems. Notes on Numerical Fluid Mechanics and Multidisciplinary Design (2018)

[10] Allmaras, S.R., Johnson, F.T.: Modifications and clarifications for the implementation of the spalart-allmaras turbulence model. In: Seventh International Conference on Computational Fluid Dynamics (ICCFD7), pp. 1–11 (2012)

[11] Xu, S., Radford, D., Meyer, M., Müller, J.-D.: Stabilisation of discrete steady adjoint solvers. Journal of Computational Physics **299**, 175–195 (2015)

[12] Christakopoulos, F.: Sensitivity computation and shape optimisation in aerodynamics using the adjoint methodology and automatic differentiation. PhD thesis, Queen Mary University of London (2013)

[13] Dhont, G., Wittig, K.: Calculix. http://www.calculix.de/

[14] Scholl, S.: Large-eddy simulation, stability and optimization of the conjugate heat transfer for cooling channels. PhD thesis, RWTH AACHEN (February 2017)

[15] Divo, E., Steinthorsson, E., Rodriguez, F., Kassab, A.J., Kappat, J.S.: Glenn-ht/bem conjugate heat transfer solver for large-scale turbomachinery models. Technical Report NASA/CR-2003-212195, NASA (2003)

[16] Scholl, S., Janssens, B., Verstraete, T.: Stability of static conjugate heat transfer coupling approaches using robin interface conditions. Computers & Fluids **172**, 209–225 (2018)

[17] Martins, J.R.R.A., Alonso, J.J., Reuther, J.J.: Complete configuration aero-structural optimization usinf a coupled sensitivity analysis method. AIAA 2002-5402 (2002)

[18] Imam-Lawal, O.R.: Adjoint based optimisation for coupled conjugate heat transfer. PhD thesis, Queen Mary University of London (2020)

[19] Hascoët, L., Pascual, V.: The Tapenade Automatic Differentiation tool: Principles, Model, and Specification. ACM Transactions On Mathematical Software **39**(3) (2013)

[20] Jones, E., Oliphant, T., Peterson, P.: SciPy: open source scientific tools for Python. Technical report, https://www.scipy.org/ (2014)

[21] L.D., H., M.S., M., E.R., T., D.A., N., R.E., Y.: Analytical and experimental evaluation of the heat transfer distribution over the surfaces of turbine vanes. Technical Report NASA CR 168015, NASA (1983)

[22] Hongjun, Z., Zhengping, Z., Yu, L., Jian, Y., Songhe, S.: Conjugate heat transfer investigations of turbine vane based on transition models. Chinese Journal of Aeronautics **26**, 890–897 (2013)

[23] Hao, Z.-R., Gu, C.-W., Ren, X.-D.: The application of discontinuous galerkin methods in conjugate heat transfer simulations of gas turbines. Energies (2014)

[24] Lin, G., Kusterer, K., Ayed, A.H., Bohn, D., Sugimoto, T.: Conjugate heat transfer analysis of convection-cooled turbine vanes using $\gamma - \mathrm{Re}_\theta$ transition model. International Journal of Gas Turbine, Propulsion and Power Systems **6**(3) (2014)