# Inverse Heat Conduction Methods in the *CHAR* Code for Aerothermal Flight Data Reconstruction

A. Brandon Oliver,* and Adam J. Amar,*
*NASA Lyndon B. Johnson Space Center*
*2101 NASA Parkway, Houston, TX, 77058*

**Reconstruction of flight aerothermal environments often requires the solution of an inverse heat transfer problem, which is an ill-posed problem of determining boundary conditions from discrete measurements in the interior of the domain. This paper will present the algorithms implemented in the *CHAR* code for use in reconstruction of EFT-1 flight data and future testing activities. Implementation details will be discussed, and alternative hybrid-methods that are permitted by the implementation will be described. Results will be presented for a number of problems.**

## I. Introduction

### I.A. Motivation

Validation of aerothermal environments developed for atmospheric entry of a spacecraft can be incrementally performed in carefully scaled ground tests; however, only flight can capture all of the relevant physics and their interactions, making flight data extremely valuable to the design and development of a spacecraft. A number of terms in the energy balance equation are of interest to the aerothermodynamicist during reentry as far as validation goes, but perhaps the most fundamental quantity of interest is the net heat flux from the environment to the surface of the heatshield. One of the primary purposes for predicting aerothermal environments is to size the thermal protection system (TPS) heatshield, and errors in the heat flux predictions can lead to heatshield failure or unnecessarily massive heatshields, which rob the system of valuable payload mass. When flight data is obtained to validate predicted environments, measurement, processing, and interpretation errors can likewise contribute to the likelihood of heatshield failure or over-conservatism if they falsely 'validate' bad predictions or 'invalidate' good predictions. Care must be taken to reduce experiment errors to every extent possible.

Heat flux can be a difficult term to measure. It is not an intrinsic property, like temperature or pressure, and it can only be inferred from measurements of other properties. Commonly available sensors that 'directly' measure heat flux, such as Schmidt-Boelter and Gardon gages, actually use temperature measurements on either side of a material with known configuration and thermal properties, and the heat flux is inferred from the temperature difference. Other sensors, such as slug calorimeters, measure the time rate of change of temperature of a carefully-designed mass with known thermal properties to infer the influx of thermal energy. Still another family of sensors operate by measuring as near as possible the surface temperature, and as with the calorimeters, the heat flux is inferred from the time rate of change of the surface temperature and heatshield thermal properties. This reliance on inference places considerable restrictions on the operational conditions of heat flux sensors. There are few sensors that can operate at the conditions seen in atmospheric reentry.

A further complication in the effort to measure heat flux is the behavior of the heatshield on which the environments are being assessed. The windside heatshields for NASA's Mars Science Laboratory (MSL) and Orion capsules, SpaceX's Dragon, and Boeing's Starliner capsules are all charring ablators, so it is expected that the heatshield will ablate, leaving anything embedded in the heatshield to protrude into the oncoming flow. A protruding sensor can amplify heating in the vicinity of the measurement and therefore measure heating higher than intended (and then likely fail).

The Apollo Program instrumented a few of the flight test vehicle ablative heatshields with sacrificial calorimeters[1], but this path was not followed for the Orion capsule EFT-1 flight test, nor the MSL MEDLI (MSL Entry Descent and

*Applied Aeroscience and CFD Branch.

Landing Instrumentation) program[2–5]. These vehicles used embedded thermocouples (TCs) inside their respective heatshields with the TCs set deep enough that they would survive through the relevant part of reentry. As with the other types of heat flux sensor, the surface heating must be inferred from the actual sensor measurements by solving a problem known as the inverse heat conduction problem (IHCP).

### I.B. Objective

This paper will describe the IHCP algorithm implemented as the *INHEAT* module of the *CHAR* ablation response code with intent to be used in reconstruction of aerothermal environments for the Orion EFT-1 flight test. Following a discussion of the IHCP and several common approaches to solving the problem, the specific formulation implemented will be given, and examples demonstrating the various modes of operation will be presented.

## II. Inverse Heat Conduction Problem

### II.A. Problem Definition

In an *inverse* heat conduction problem, the objective is to estimate the boundary conditions given temporally and spatially discrete solutions interior to the domain of a standard heat conduction problem. Ozisik and Orlande[6] describe the problem as such:

> *In the direct problem the causes (heat flux) are given, the effect (temperature) is determined; whereas in the inverse problem the effect is given, the cause (or causes) is determined.*

The problem is an ill-posed problem, meaning that solutions are not guaranteed to be unique and that the solution does not always vary smoothly with small perturbations on the inputs. Frequently, optimization methods are used to obtain the desired boundary condition values. Given sufficiently accurate measurements and sufficient knowledge of the material thermal properties, very accurate estimates (sometimes termed reconstructions) of the boundary conditions can be obtained. However, the problem's ill-posedness means that the reconstruction will be very sensitive to measurement or modeling errors, and slight measurement noise could render an algorithm unstable. Any algorithm proposed to solve the IHCP must address the manner in which smoothing is applied to handle this tendency.

Depending on the problem of interest, the unknown boundary conditions can take several forms. They can be temporally varying, spatially varying, or both. In most cases in the literature, IHCP algorithms tend to focus on problems for which the boundary conditions are temporally varying, sometimes defined as *function estimation* problems. If the boundary conditions are fixed in time and only vary in space (or are just simply constant) the problem is generally classified as a *parameter estimation* problem. The gulf between function and parameter estimation techniques is not very large, but IHCP algorithms are often built to explicitly address issues introduced by temporally varying boundary conditions.

A large number of approaches have been proposed to solve the IHCP, however a fairly common and well characterized approach is to discretize the boundary condition values in the time domain and compute values that minimize the least-squares difference of the measured and computed temperatures.

$$\underset{\mathbf{q}}{\text{Minimize}}\ (\mathbf{Y}-\mathbf{T}(\mathbf{q}))^T(\mathbf{Y}-\mathbf{T}(\mathbf{q})) \tag{1}$$

where $\mathbf{Y}$ is a vector of temperature measurements and $\mathbf{T}(\mathbf{q})$ is a vector of computed temperatures given the vector of boundary conditions $\mathbf{q}$. Frequently, the derivative of Equation 1 is taken with respect to the boundary condition values and set equal to zero to find the minimum value. This introduces the derivatives

$$\frac{\partial \mathbf{T}}{\partial \mathbf{q}^T} = \begin{bmatrix} \frac{\partial T_1}{\partial q_1} & \frac{\partial T_1}{\partial q_2} & \cdots & \frac{\partial T_1}{\partial q_{max}} \\ \frac{\partial T_2}{\partial q_1} & \frac{\partial T_2}{\partial q_2} & \cdots & \frac{\partial T_2}{\partial q_{max}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial T_{max}}{\partial q_1} & \frac{\partial T_{max}}{\partial q_2} & \cdots & \frac{\partial T_{max}}{\partial q_{max}} \end{bmatrix} \tag{2}$$

which are known as sensitivity coefficients. As will be shown later, the sensitivity coefficients drive IHCP algorithms, and depending on how they behave for certain problems, the ideal IHCP approach may change. The sensitivity coefficients will be discussed at length in Section II.C.

### II.B. Common Algorithm Classifications

A review of the literature reveals many different algorithms and approaches that have been proposed for solving the IHCP. There are many different ways that these algorithms can be categorized to help identify which ones may work for a given problem. A brief overview of several methods of categorization will be described.

The first categorization metric describes the nature of the underlying direct problem. The spatial dimensions of the system (1-D, 2-D, or 3-D) can influence the choice of algorithm, particularly from a cost perspective. More selective for IHCP algorithms, however, is whether the problem is linear or non-linear. If the sensitivity coefficients depend on any values in the the boundary condition vector, or if any of the boundary condition functions are affected by multiple elements of the boundary condition vector, the IHCP is non-linear. Temperature-dependent thermal properties introduce non-linearity, as do convective boundary conditions (through the dependence on surface temperature). Solutions to linear IHCPs are considerably faster and generally do not require iteration as non-linear IHCPs do.

There can be one or multiple boundary conditions to a given system at a given time that influence the sensor response (e.g., different heating levels to different parts of the domain boundary). Given that many IHCP algorithms can handle spatially multiple dimensional systems with one or several unknown boundary conditions, the term *multi-dimensional inverse* can be ambiguous. In this work, the term *multi-component* will be used to describe problems with multiple time-varying boundary conditions.

Another key categorization metric that strongly influences IHCP algorithms is the time in which a solution is required. Many applications require the IHCP solution in a control system feedback loop and need to obtain a solution as quickly as possible, whereas other applications are not as time-critical and can instead focus on accuracy in a detailed reconstruction. Algorithms for real-time applications, such as the future time filter coefficient method[7,8], neural network algorithms[9], Kalman-filter methods[10], and the calibration integral method[11,12], must attempt to infer the boundary conditions without use of extensive *future time* data (temperature data at times after the time interval for which the boundary condition is being estimated). Algorithms for post-test applications, on the other hand, have all of the data available to work with. This work will focus on post-test analysis applications.

One of the most quoted categorization metrics in the literature concerns how the temporally discretized values of the unknown boundary condition are evaluated. In *sequential* algorithms, they are estimated one at a time, in order, from the earliest time to the latest. By contrast, *whole domain* algorithms simultaneously estimate all of the values in one step. Each method has strengths and weaknesses. Sequential algorithms can be computationally more efficient for non-linear problems as less time is spent computing temperature estimates and sensitivity coefficients at times where the initial guess is poor. On the other hand, whole domain algorithms perform much better when measured temperatures take a substantial amount of time to respond to changes in the boundary condition. Neither is universally better than the other; the optimal choice comes down to the sensitivity coefficients of the specific problem. Some examples of sequential algorithms are the future time family of algorithms from Beck et al.[8], and some examples of whole domain algorithms are the Gauss-Newton method[13] and conjugate-gradient methods (with and without the adjoint equation)[6,14]. Perhaps a bit tangent but still related to this metric, space marching techniques[15,16] swap a spatial dimension for time and compute temperatures sequentially in space from the sensor location to the boundary.

Because of the ill-posed nature of the IHCP and the inevitable presence of measurement noise, all practical IHCP algorithms require some form of smoothing, regularization, allowed bias, or other treatment of the solution's sensitivity to data noise. One method that Beck's future time algorithms rely on is to integrate a flux interval over several measurements to average out noise. Probably the most common method of smoothing is *Tikhonov regularization*[6,8,17], where a penalty function is added to the minimization objective function (Equation 1) that penalizes when elements of the solution vector $\mathbf{q}$ vary in certain undesirable ways. For instance, adding the term $\alpha\mathbf{Hq}$ to Equation 1 (with $\alpha$ being the regularization parameter, and $\mathbf{H}$ being the regularization matrix) can bias the predicted temperatures when elements of $\mathbf{q}$ become unreasonably large or if they vary too rapidly, depending on the choice of $\mathbf{H}$. These are both examples of a *regularization operator*. An alternative approach is to *regularize the data*, as is done in the mollification method[16]. Alternatively, the *iterative regularization principle* of the conjugate-gradient-based methods does not explicitly bias the solution, however it uses an intricate iteration stopping criteria[6] to establish when the solution is close enough to the data that the gross character of the data is matched, but the noise is not.

Since sensitivity coefficients are often needed for IHCP algorithms, the manner in which these are computed can be a discriminator between different methods. Sometimes analytic formulations are available for simple linear problems, but more frequently they must be computed by *finite differences* or specially formulated *sensitivity equations*. The *adjoint* equation used in some conjugate-gradient based methods is an example of a sensitivity equation, but sensitivity equations can be defined even for regular non-linear heat conduction problems[18].

### II.C. Sensitivity Coefficients

Physically, the sensitivity coefficients defined in Equation 2 represent the sensitivity of each temperature in the vector $\mathbf{T}$ to each model parameter or boundary condition value in the vector $\mathbf{q}$ that is sought in the IHCP solution. The IHCP optimization algorithm needs $\mathbf{T}(\mathbf{q})$ for objective function (Equation 1) evaluation, but updated estimates of $\mathbf{q}$ are largely driven by $\frac{\partial \mathbf{T}}{\partial \mathbf{q}^T}$. As such, a close study of the sensitivity coefficients can provide invaluable insight to how a particular IHCP algorithm will perform for a given problem.

Generalization of the sensitivity coefficient matrix can be difficult without putting constraints on it according to the problem of interest, but one general statement that can be made is that an IHCP will be linear if the sensitivity coefficient matrix is independent of $\mathbf{q}$. In this instance, a Taylor-series expansion of $\mathbf{T}(\mathbf{q})$ about $\mathbf{q} = \mathbf{0}$,

$$\mathbf{T}(\mathbf{q}) = \mathbf{T}(\mathbf{0}) + \frac{\partial \mathbf{T}}{\partial \mathbf{q}^T}\mathbf{q}, \tag{3}$$

with $\mathbf{T}(\mathbf{0}) = T_0$ being the uniform initial temperature, will yield the exact values of $\mathbf{T}(\mathbf{q})$ for a given $\mathbf{q}$. As in most cases, linear problems are much easier to solve, and many algorithms for non-linear IHCP problems linearize about a specific $\mathbf{q}$ inside an iteration loop to advance the solution.

To build intuition necessary for IHCP analysis, it is beneficial to simplify the problem. In the remainder of this section, we will limit ourselves to a transient, 1-D, semi-infinite solid heat conduction problem with temperatures evaluated at a single location at times consistent with the unknown values. The unknown values will be a piecewise-constant representation of a time-varying heat flux on the exposed boundary. In this case, the sensitivity coefficient matrix becomes a square lower-triangular matrix as the parabolic nature of the heat equation suggests that temperatures will be unaffected at time $t_i$ by a boundary condition applied at a later time $t_m$ (i.e. $\frac{\partial T_i}{\partial q_m} = 0$ if $t_i < t_m$).

If we assume constant thermal properties and a constant heat flux boundary condition, the exact solution can be determined by textbook[19] expressions. Because the heat equation with constant properties is a linear differential equation, the principle of superposition can be used to construct the solution corresponding to a transient piecewise-constant heat flux boundary condition. The building block solution used will be the solution to a unit step heat flux

$$q(t) = \begin{cases} 0, & t < 0 \\ 1, & t > 0 \end{cases}, \tag{4}$$

with the solution denoted by $\phi(x, t)$.

The transient heat flux boundary condition function is approximated by a piecewise-constant function with interval boundaries defined by times $0, \lambda_1, \lambda_2, ..., \lambda_M$. The boundary flux values will be denoted by $q_m = q(\lambda_{m-1/2})$, where $q(\lambda_{m-1/2})$ indicates the flux value between times of $\lambda_{m-1}$ and $\lambda_m$. Each $q_m$ will be referred to as a *component*. With this definition, the contribution to the temperature of a flux component $q_m$ at a specified time $t_i$ is given by

$$\frac{\partial T_i}{\partial q_m} = \Delta\phi_{i-m} = \begin{cases} \phi(x, t_i - \lambda_{m-1}) - \phi(x, t_i - \lambda_m), & i \geq m \\ 0, & i < m \end{cases} \tag{5}$$

Equation 5 is known as the *pulse* sensitivity coefficient, though the word 'pulse' is often omitted in discussion. The *step* sensitivity coefficient differs from this in that the subtracted $\phi$ term is omitted and represents the sensitivity to all future flux components changing uniformly. Given the linearity of the problem, the temperature at time $t_i$ can be evaluated as

$$T(x, t_i) = T_0 + \sum_{m=1}^{M} q_m \frac{\partial T_i}{\partial q_m}. \tag{6}$$

This is consistent with the approximate numerical solution using the Duhamel method presented by Beck et al.[8]

Consider the case of temperatures $1\,\mathrm{mm}$ and $5\,\mathrm{mm}$ below the surface of a material with constant properties ($\rho = 280\,\mathrm{kg/m^3}$, $k = 0.5\,\mathrm{W/mK}$, and $C_p = 1860\,\mathrm{J/kgK}$) shown in Figure 1. The heat flux profile for this case is shown in Figure 1(a) with the piecewise-constant flux components taken to be $1\,\mathrm{s}$ long. The sensitivity coefficients evaluated with Equation 5 are shown in Figures 1(b) and 1(c). The temperature evaluated by Equation 6 is shown in Figures 1(d) and 1(e) with the contributions from each flux component index $m$ indicated with different colored regions and $T_0 = 0\,\mathrm{K}$. Note that all plots in Figure 1 are color-consistent, with each color representing an index $m$ in Equation 6. For example, the first green curve represents $m = 3$: the heat flux component active from $t = 2\,\mathrm{s}$ to

3 s, the corresponding sensitivity coefficient which appears shortly after $t = 2\,\mathrm{s}$, and the corresponding temperature contribution, the lowest of the green bands.

A number of observations can be made regarding the results of this example case. First of all, note that because this is a linear problem, each of the sensitivity coefficient components shown for a given depth is identical except for a time shift (this is not generally the case for non-linear problems). Indeed, as suggested by Equation 5, each coefficient is a function of the time difference between the current time $t_i$ and the time that the respective heat flux component began $\lambda_{m-1}$. This time difference will be termed the *development time* of the sensitivity coefficient. As will be seen later, the development time required for the sensitivity coefficient to grow to substantial values is a strong function of the physical system and can have considerable impact on the performance of IHCP algorithms.

Secondly, the temperature response plots break out the individual components of the summation in Equation 6 in this manner to illustrate how long it can take for a particular flux component to affect the overall temperature response, relative to the contributions from all other flux components. The thickness of the colored bar at a particular time is the product of the developed sensitivity coefficient and the corresponding flux magnitude.

Comparison of the sensitivity coefficients and temperature response plots for the two depths shows some striking differences. At $x = 1\,\mathrm{mm}$, the contribution from early components is seen to grow and decay rapidly, and when the step increase in heating arrives, the blue component (the first with elevated heating) leads the change in the temperature response. In fact, the coefficients decay so rapidly that the total temperature response is shown to decrease at the end of the square wave without the flux going negative. By contrast, the $x = 5\,\mathrm{mm}$ case shows that it takes considerable time to even see the effect of the components with increased heating, and nearly all of the components are still contributing significantly to the total response at the end of the time domain.

The practical implication to an IHCP of the differences between to two responses in Figure 1 is this: for a measurement at a given time (say $t = 8\,\mathrm{s}$), how strongly is the measurement (the total temperature response) affected by each of the unknown heat flux components being sought? In the case of the shallow measurement, nearly half of the temperature response at $t = 8\,\mathrm{s}$ is provided by the two components that immediately precede the measurement. For the deep measurement at $t = 8\,\mathrm{s}$, the two preceding components have only just become visible. An inverse algorithm trying to piece together the magnitude of these two heat flux components will have plenty of 'information' on the components of interest by looking at measurements near $8\,\mathrm{s}$ in the shallow case. However, for the deep case, it will need to use lots of measurements after $8\,\mathrm{s}$ to gather sufficient information, and it will have to try and untangle that information from the information on neighboring flux components. Furthermore, the sharp reduction in magnitude of the sensitivity coefficients as the measurement moves deeper means that any measurement noise (presumably at some fixed level of units $K$) will wipe out considerably more usable information in the deeper case than in the shallow case.

### II.D. Motivation for New Algorithm

As was indicated in Section II.B, one common and significant discriminator between IHCP algorithms is the *sequential* vs. *whole-domain* approach to solving for the unknown boundary condition component values. The choice of approach primarily influences how the sensitivity coefficients are calculated and used. A whole domain algorithm computes the full sensitivity coefficient matrix and solves a system of equations similar to Equation 6 (one equation for each measurement) simultaneously for every unknown value. A sequential algorithm, however, will only compute a single step sensitivity coefficient for the component being evaluated (only for a short development time known as the *future time window*) and solve only for the temperatures (using Equation 6) that appear in this future time window.

From a cost perspective, sequential methods are considerably more desirable than whole domain methods. Recall that the sensitivity coefficient matrix will have dimensions of number of measurements by number of unknowns. Since a finite-difference approximation to the sensitivity coefficients are evaluated for many realistic problem, evaluating this matrix can become costly if the numerical model for the system is expensive. As an example (described in more depth in Seciton III.D), a sequential reconstruction of the MSL dataset presented by Mahzari et al.[13] would require the ablator material response model integrating $1.95 \cdot 10^3\,\mathrm{s}$ of time to complete the reconstruction. By contrast, the whole domain reconstruction presented by Mahzari et al. required the integration of $6.79 \cdot 10^6\,\mathrm{s}$, three orders of magnitude more effort! With this sort of difference in cost, there must be a good reason to use a whole domain algorithm, and indeed there is a very good reason.

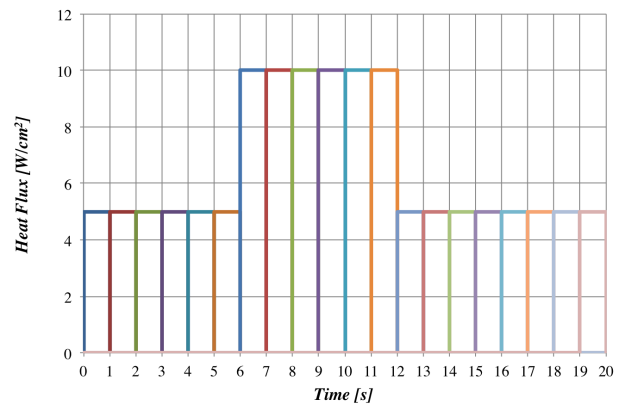
Conventional sequential algorithms make two key assumptions: 1) that the flux components through the future time window follow an assumed pattern that is reduced to one unknown (typically that flux is constant through the whole window), and 2) that there is a discernible amount of 'information' in the future time window measurements due to the unknown flux relative to measurement error. If the future time window is too short, assumption 2 breaks down and the algorithm becomes unstable and driven by measurement error. As such, the future time window must be chosen for each problem to ensure assumption 2 is valid. Recalling the example in Figure 1, if the sensitivity coefficients

develop rapidly, the flux components will contribute significantly to measurements near the flux component, but if they develop slowly, that may not be the case. If the coefficients develop too slowly, a long future time window is required and the first assumption begins to break down, yielding poor results that appear overly 'smoothed.' In this case, a whole domain algorithm performs much better since measurements can contribute their information to all of the flux components as they are all solved simultaneously.

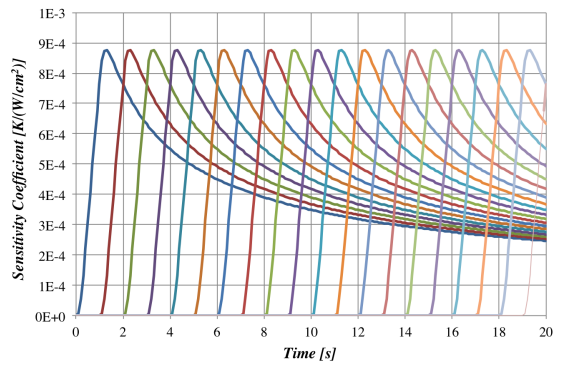
The conditions used in the example of Figure 1 are not completely arbitrary; they are approximations of the TACOT[20] material model, which is representative of a low density charring ablator similar to what is used on MSL and Orion. As such, the range of development times for sensitivity coefficients illustrated in these figures is representative of that which would be seen in flight. Surface recession generally forces thermocouples (TCs) to be embedded at a depths greater than $2\,\mathrm{mm}$ to ensure their survival. This depth drives sequential algorithms to use a long future time window as sensitivity coefficients can develop slowly. The computational cost of the additional equations modeled in ablation simulations is non-negligible and properties are highly non-linear, so the reduced simulation time of a sequential algorithm is strongly desirable. Additionally, ablation simulations are only modestly robust, so wild excursions of boundary conditions that could be seen in a non-linear iteration of a whole domain algorithm can lead a reconstruction to fail. An algorithm that is predominantly sequential but draws from the whole domain algorithm to permit longer sensitivity coefficient development time would be quite useful for reconstruction of heating environments on ablative materials.

Another situation that can arise in a practical flight experiment is a large localized change in relevant timescales of the heating profile. An example of this would be a rapid spike in heating caused by a steering jet firing in an otherwise slowly changing heating profile. In this case, the sensitivity coefficients of a near-surface TC could develop rapidly enough compared to the rate change of the overall heating that a sequential algorithm would be adequate, but so slowly compared to the change in heating during the jet firing that a whole domain method would be required to characterize the jet heating augmentation. An algorithm that could seamlessly switch from a predominantly sequential algorithm to a predominantly whole domain algorithm in a local region would have considerable practical application.

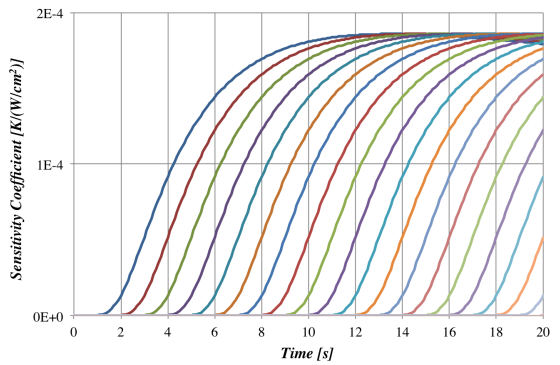
In the present work, a hybrid sequential-whole domain algorithm, termed the *Sequential Subdomain* algorithm, is proposed that permits the solution of multiple unknown boundary condition components on an otherwise sequential future time window. It is defined in such a way that allows the analyst to tailor the algorithm to the sensitivity coefficients of the physical system. Depending on the values of a few parameters, Beck's Future Time algorithm or a whole-domain Gauss-Newton method can be obtained. In the hybrid mode, it allows sequential specification of the unknown boundary condition values, allowing reduced computational effort than the pure whole domain approach without having to accept the excessive smoothing introduced for slowly responding sensors in the sequential function specification algorithm.



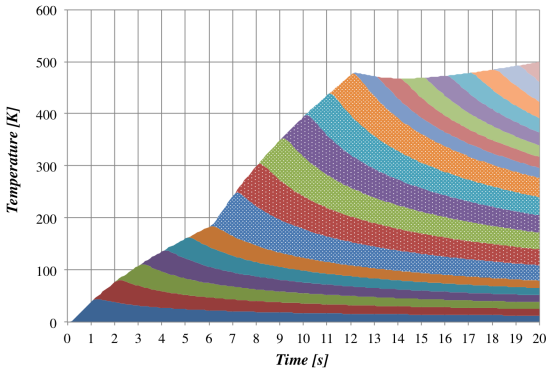

(a) Heat flux profile



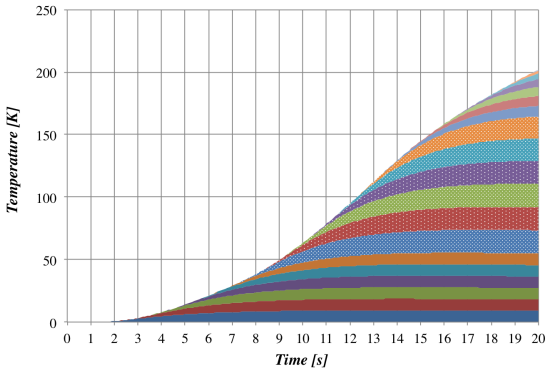

(b) Pulse sensitivity coefficients, $x = 1\,\mathrm{mm}$




(c) Pulse sensitivity coefficients, $x = 5\,\mathrm{mm}$




(d) Temperature Response, $x = 1\,\mathrm{mm}$




(e) Temperature Response, $x = 5\,\mathrm{mm}$

Figure 1: Duhamel solution for example case with two TC depths ($x = 1\,\mathrm{mm}$ and $x = 5\,\mathrm{mm}$) exposed to the same heating profile to illustrate how different sensitivity coefficients will affect the thermal response.

# III. *INHEAT* Reconstruction Algorithm

## III.A. Derivation

The algorithm implemented here is based on the trial function specification algorithm of Beck et al.[8] It is a least-squares optimization of the modeled temperature to the measured temperatures with a Tikhonov regularization term for stability. The function to be minimized is given by

$$S = (\mathbf{Y} - \mathbf{T})^T \boldsymbol{\Psi}^{-1} (\mathbf{Y} - \mathbf{T}) + \alpha \left[\mathbf{H} (\mathbf{q} - \tilde{\mathbf{q}})\right]^T \mathbf{W} \left[\mathbf{H} (\mathbf{q} - \tilde{\mathbf{q}})\right] \tag{7}$$

where $\mathbf{Y}$ is the vector of temperature measurements (the reconstruction target), $\mathbf{T}$ is the temperature resulting from the numerical model of the system, $\boldsymbol{\Psi}^{-1}$ is a covariance matrix describing $\mathbf{Y}$, $\alpha$ is the regularization parameter, $\mathbf{H}$ the regularization matrix, $\mathbf{W}$ is the regularization weighting matrix, and $\mathbf{q}$ is the vector of boundary condition values. The term $\tilde{\mathbf{q}}$ is a vector of expected boundary condition values that can be used to fine-tune regularization if desired. It is assumed to have the form $\tilde{\mathbf{q}} = \mathbf{B}\mathbf{q} + \tilde{\mathbf{q}}^f$ with $\mathbf{B}$ and $\tilde{\mathbf{q}}^f$ specified by the user. Note that an arbitrary number of regularization terms of this form can be applied as desired, but only one is shown to simplify the presentation. The specific structure of all of these matrices will be described in Section III.B. Depending on how they are defined, sequential function specification or whole domain solutions (or combinations in between those limits) can be obtained.

We seek to minimize the objective function $S$, so we take the derivative of Equation 7 with respect to each of the boundary condition values

$$\frac{\partial S}{\partial \mathbf{q}} = -2\mathbf{X}^T \boldsymbol{\Psi}^{-1} (\mathbf{Y} - \mathbf{T}) + 2\alpha \left[\mathbf{H} (\mathbf{I} - \mathbf{B})\right]^T \mathbf{W}\mathbf{H} (\mathbf{q} - \tilde{\mathbf{q}}) \tag{8}$$

where

$$\mathbf{X} = \left.\frac{\partial \mathbf{T}}{\partial \mathbf{q}^T}\right|_{\mathbf{q}^*} = \begin{bmatrix} \frac{\partial T_1}{\partial q_1} & \frac{\partial T_1}{\partial q_2} & \cdots & \frac{\partial T_1}{\partial q_L} \\ \frac{\partial T_2}{\partial q_1} & \frac{\partial T_2}{\partial q_2} & \cdots & \frac{\partial T_2}{\partial q_L} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial T_K}{\partial q_1} & \frac{\partial T_K}{\partial q_2} & \cdots & \frac{\partial T_K}{\partial q_L} \end{bmatrix}_{\mathbf{q}^*} \tag{9}$$

is the sensitivity coefficient matrix evaluated around the current solution vector $\mathbf{q}^*$ where $K$ indexes the total number of measurements and $L$ indexes the total number of unknowns[a]. Computation of the objective function requires temperatures corresponding to the new flux vector, so a Taylor series expansion about the current solution yields

$$\mathbf{T} = \mathbf{T}^* + \mathbf{X}\Delta\mathbf{q} \tag{10}$$

where $\mathbf{T}^*$ is the temperature evaluated with the solution vector $\mathbf{q}^*$, and $\Delta\mathbf{q} = (\mathbf{q} - \mathbf{q}^*)$ is the update to the estimated solution vector. Setting the objective function derivative to zero, substituting in the Taylor series expansion, and rearranging yields the system

$$\mathbf{L}\Delta\mathbf{q} = \mathbf{R} \tag{11}$$

where

$$\mathbf{L} = \mathbf{X}^T \Psi^{-1} \mathbf{X} + \alpha \left[\mathbf{H} (\mathbf{I} - \mathbf{B})\right]^T \mathbf{W} \left[\mathbf{H} (\mathbf{I} - \mathbf{B})\right] \tag{12}$$

$$\mathbf{R} = \mathbf{X}^T \Psi^{-1} (\mathbf{Y} - \mathbf{T}^*) - \alpha \left[\mathbf{H} (\mathbf{I} - \mathbf{B})\right]^T \mathbf{W}\mathbf{H} \left( (\mathbf{I} - \mathbf{B}) \mathbf{q}^* - \tilde{\mathbf{q}^f} \right) \tag{13}$$

The term containing $\mathbf{q}^*$ does not appear in the derivation of Beck et al.[8] as their derivation makes the assumption that $\mathbf{q}^* = \mathbf{0}$ when evaluating $\mathbf{T}^*$. This term is necessary for the regularization to appropriately smooth the solution $\mathbf{q}$ if $\mathbf{T}^*$ and $\mathbf{X}$ are evaluated using non-zero values of $\mathbf{q}^*$. If this term is omitted, the regularization smooths the update $\Delta\mathbf{q}$.

Introducing the Taylor series approximation (Equation 10) linearizes the problem about $\mathbf{q}^*$, so iteration will likely be required to converge the solution $\mathbf{q}$ if the problem is non-linear.

[a]Temperatures are measured at locations $x_j$ and times $t_i$ so $K = JI$. Unknowns are defined at $x_n$ distinct locations (different boundaries) and at times $t_m$, so $L = NM$. However, as will be described in the next section, $K$ and $L$ can be smaller if only a subset of the unknowns are evaluated.

### III.B. Matrix Definition

It is common in the literature to assume that boundary condition values are estimated on a uniform time scale that is consistent with measurement times. This is notationally convenient; however, it can cause difficulties in practical applications. The cost of a reconstruction scales with the number of unknowns estimated, so it may be desirable to reconstruct at a lower frequency than data is available. Additionally, reconstructing boundary conditions on a different time scale could be motivated by missing or otherwise compromised data, changes in data rate near an expected transient event, or a mismatch in measurement times of data supplied by multiple sensors. In an attempt to make this as general and flexible as possible, the algorithm has been defined to allow measurement and boundary condition estimation times to be completely flexible, and it is up to the user to specify a reasonable problem for which a solution exists. This choice will make the notation more complex.

The algorithm presented in this work permits sequential evaluation of the terms in $\mathbf{q}$. Each of the sequential evaluation steps will be referred to as a *local solution*. A local solution will solve the non-linear system of Section III.A for a subset of the $\mathbf{q}$ vector on an appropriate subdomain. Introduction of the local solution and keeping the global and local index systems distinct further complicates the notation. Furthermore, an arbitrary number of sensors can be providing data and multiple distinct unknown boundary conditions could be estimated at a given time, further complicating the notation.

Given the complexity of this system, the notation will be explained in detail. Figure 2 presents an illustration of how time is indexed in this definition using two timelines. The top timeline describes the notation for the overall problem, and the red line below shows the nomenclature for a specific local solution. The top row of each timeline denotes the measurement times for which target data is provided to the algorithm, with the letter $i$ identifying the global index and $r$ identifying the local index. Note that the spacing between these times are not uniform, but the times in the local solution are a subset of the global measurement times. The bottom row of each line denotes the time *intervals* for which BC values are to be estimated, with the letter $m$ indexing the global set and $f$ indexing the local set. There is no requirement that these intervals be regularly spaced, and the BCs defined in each interval are assumed to be constant through the interval (the estimated function is assumed to be piecewise-constant). The max measurement time in the global problem is denoted by $I$, the max measurement time index in a particular local solution is $R$, the max BC interval index in the global problem is $M$, and the max BC interval index in a particular local solution is $F$. Note that for different local solutions in the same global problem, $R$ and $F$ may change based on the length of the future time window (which defines the length of the local solution) and the defined global measurement and BC intervals.

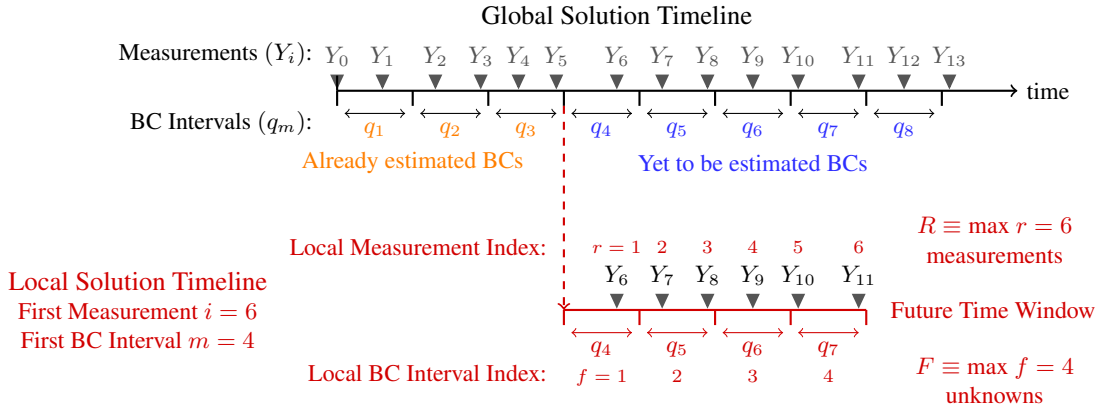



Figure 2: Definition of nomenclature describing the time-scales of the reconstruction.

Beyond the times described in Figure 2, there may be input from multiple sensors at each measurement time (for instance, if there are several TCs in the system), and there may be multiple distinct BC values estimated on each interval. Multiple measurement locations are indexed using the letter $j$ (max $J$) in the matrix definitions, and multiple BCs on each interval are indexed with the letter $n$ (max $N$). In the present derivation, it is possible to have only a subset of sensors providing data for a specific measurement time by using a special blanking matrix (whereby contributions from 'missing' data are ignored). However, at this time, it is not possible to define separate intervals for each of the distinct BCs; all distinct BCs are estimated on every BC interval. Given this, there will always be $(N \times F)$ BC values estimated in each local solution, and this dimension is given a special symbol $P$. A summary of the index system

described is presented in Table 1.

Table 1: Summary of indices used in matrix definitions

| Global Solution Indices: | |
|---|---|
| $j$ | TC number (max $J$) |
| $i$ | Global measurement times (max $I$) |
| $m$ | Global boundary condition time intervals (max $M$) |
| $n$ | Distinct boundary conditions (max $N$) |

| Local Solution Indices: | |
|---|---|
| $r$ | Measurement times in current local solution (max $R$) |
| $f$ | Boundary condition time intervals in current local solution (max $F$) |
| $k$ | Boundary condition values in current local solution (max $P = FN$) |

All of the matrices presented herein are built for a specific local solution with $i$ referring to the first global measurement time in a local solution and $m$ the first global BC interval in the local solution domain.

The vectors that represent temperature (either the target data $\mathbf{Y}$ or modeled temperatures $\mathbf{T}$) have the following block form:

$$\mathbf{T} = \overbrace{\begin{bmatrix} \mathbf{T}(i) \\ \mathbf{T}(i+1) \\ \vdots \\ \mathbf{T}(i+R-1) \end{bmatrix}}^{1} \Big\updownarrow R \text{ blocks } (JR \text{ elements})} \tag{14}$$

where each block is of the form

$$\mathbf{T}(r) = \overbrace{\begin{bmatrix} T_1(r) \\ T_2(r) \\ \vdots \\ T_J(r) \end{bmatrix}}^{1} \Big\updownarrow J} \quad . \tag{15}$$

If any data is unavailable from either $\mathbf{Y}$ (due to missing measurement) or $\mathbf{T}$ (possible due to surface recession in ablation problems), those elements in $\mathbf{Y}$ and the corresponding elements in the $\mathbf{T}$ matrix are both set to zero to prevent that time point from contributing to the objective function (Equation 7).

The vectors that represent the estimated boundary condition values have a similar block form:

$$\mathbf{q} = \overbrace{\begin{bmatrix} \mathbf{q}(m) \\ \mathbf{q}(m+1) \\ \vdots \\ \mathbf{q}(m+F-1) \end{bmatrix}}^{1} \Big\updownarrow F \text{ blocks } (P = FN \text{ elements})} \tag{16}$$

where each block is given by

$$\mathbf{q}(f) = \overbrace{\begin{bmatrix} q_1(f) \\ q_2(f) \\ \vdots \\ q_N(f) \end{bmatrix}}^{1} \Bigg\updownarrow N \quad . \tag{17}$$

The sensitivity coefficient matrix has the form:

$$\mathbf{X} = \overbrace{\begin{bmatrix} \tilde{\mathbf{a}}(i) \\ \tilde{\mathbf{a}}(i+1) \\ \vdots \\ \tilde{\mathbf{a}}(i+R-1) \end{bmatrix}}^{1 \text{ super block } (F \text{ blocks, } P = FN \text{ columns})} \Bigg\updownarrow R \text{ blocks } (JR \text{ rows})} \tag{18}$$

where

$$\tilde{\mathbf{a}}(r) = \overbrace{\begin{bmatrix} \mathbf{a}_{11}(r) & \mathbf{a}_{12}(r) & \dots & \mathbf{a}_{1F}(r) \\ \mathbf{a}_{21}(r) & \mathbf{a}_{22}(r) & \dots & \mathbf{a}_{2F}(r) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_{J1}(r) & \mathbf{a}_{J2}(r) & \dots & \mathbf{a}_{JF}(r) \end{bmatrix}}^{F \text{ blocks } (P = FN \text{ columns})} \Bigg\updownarrow J \tag{19}$$

and each term in $\tilde{a}(r)$ is of the form

$$\mathbf{a}_{jf}(r) = \overbrace{\begin{bmatrix} a(r,f,j,1) & a(r,f,j,2) & \dots & a(r,f,j,N) \end{bmatrix}}^{N} \Big\updownarrow 1 \tag{20}$$

and

$$a(r,f,j,n) = \frac{\partial T(\mathbf{x}_j, t_r)}{\partial q_n(t_f)} \tag{21}$$

Note that in Equation 21, $t_f$ refers to the BC time interval $f$ and not a specific time.

The regularization matrix $\mathbf{H}$ governs how the algorithm attempts to provide stability by smoothing the solution. Tikhonov regularization is a penalty function, where the penalized condition is defined by $\mathbf{H}$. One common regularization method is to penalize based on the absolute magnitude of the solution, $\alpha_0 \sum_{i=1}^{F} q_i^2$, which is often referred to as $0^{th}$-order regularization. Alternatively, $1^{st}$-order regularization penalizes based on the difference of one solution value compared to the next, $\alpha_1 \sum_{i=1}^{F-1} (q_{i+1} - q_i)^2$. For $0^{th}$-order regularization, the definition of $\mathbf{H}$ is straightforward; however, for higher-order regularization where multiple solution terms are included, the structure of $\mathbf{H}$ will be different if the smoothing is applied to multiple BCs in space, multiple BC values in time, or both.

In a departure from the method described in Beck et al.[8], temporal and spatial regularization are handled in separate regularization terms, each with their own $\mathbf{H}$ (with dimensions $P \times P$) and $\alpha$. Spatial regularization is highly problem dependent and requires the user to input a matrix $\mathbf{h}_x$ of dimensions $N \times N$ that provides the desired mapping between uncertain components. The overall spatial regularization matrix is then given by putting $F$ copies of $\mathbf{h}_x$ in block diagonal form

$$\mathbf{H}_x = \text{diag}\Big[\mathbf{h}_x \quad \mathbf{h}_x \quad \dots \quad \mathbf{h}_x\Big]_{(P \times P)} . \tag{22}$$

Temporal regularization is a bit more straightforward given the known temporal connectivity of the solutions. $0^{th}$-order temporal regularization is obtained by setting $\mathbf{H}_t = \mathbf{I}_P$. $1^{st}$-order regularization is obtained by defining the temporal regularization matrix as the block $(F \times F)$ matrix of the form:

$$\mathbf{H}_t = \begin{bmatrix} -\mathbf{I}_N & \mathbf{I}_N & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_N & \mathbf{I}_N & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & -\mathbf{I}_N & \mathbf{I}_N \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}_{(P \times P)} \tag{23}$$

Selection of the value for the regularization parameter $\alpha$ is a common topic in the literature. A too-small value of $\alpha$ will not provide the desired smoothing effect, but a too-large value will overly smooth the solution and introduce too much bias in the reconstructed temperature vector. As pointed out by Mahzari et al.[13], the value of $\alpha$ will depend on the magnitude of the unknowns (in their case, small values of the heat transfer coefficient in their reconstructions required regularization parameters that were orders of magnitude greater than those needed for heat flux reconstructions in the literature). Many authors resort to a trial-and-error method for defining appropriate values for $\alpha$. The "discrepancy principle" described by Woodbury[17] captures a commonly stated notion that $\alpha$ should be chosen as the minimum value that will yield an average difference between measured and modeled temperatures that is consistent with the expected accuracy of the measurements. The cross-validation method[16,21,22] is a commonly cited method to quantitatively define an optimum value. In the present implementation, it was observed that the $\alpha$ term seemed to work the best when the two terms in Equation 12 contained elements of approximately the same magnitude. To this end, the user may indicate that $\alpha$ should be chosen for every local solution iteration to be a constant input factor $c$ of the maximum element of the squared sensitivity coefficient matrix:

$$\alpha = c \cdot \left\| \mathbf{X}^T \mathbf{\Psi}^{-1} \mathbf{X} \right\|_\infty \tag{24}$$

A degree of experimentation is still required as this metric does not take into consideration how much noise is present in the data to be fit. However, a value of $c = 1$ generally provides a good starting point considering the thermal properties of the system and the magnitude of the solution vector.

The regularization weighting matrix $\mathbf{W}$ has taken on a novel form to permit 'focusing' the regularization for sequential subdomain solutions. For pure whole domain solutions, shifting the emphasis of the regularization likely will not have any benefit. However, as is shown later, there is an advantage for certain problems to suppress the regularization at different ends of the local solution time domain (different $f$ indices) with the following:

$$\mathbf{W} = \text{diag}\Big[ \mathbf{W}(1) \quad \mathbf{W}(2) \quad \dots \quad \mathbf{W}(F) \Big]_{(P \times P)} \tag{25}$$

$$\mathbf{W}(f) = \text{diag}\Big[ \tfrac{1}{10^{fW_f}} \quad \tfrac{1}{10^{fW_f}} \quad \dots \quad \tfrac{1}{10^{fW_f}} \Big]_{(N \times N)} \tag{26}$$

where $W_f$ is a user input factor. For pure whole domain solutions, $W_f$ can be set to 0 to obtain $\mathbf{W} = \mathbf{I}_P$ as desired. However, a positive value of $W_f$ will result in less regularization at later $f$, whereas a negative $W_f$ will yield less regularization on earlier $f$.

The covariance matrix $\mathbf{\Psi}$ describes the nonsystematic errors in the measurements provided to the algorithm. This matrix can be of considerable concern in parameter estimation and experiment design[23]; however, in inverse heat conduction problems, the transient nature of the problem complicates the computation of this matrix. When several "standard" assumptions are made of the measurements, specifically that the measurement errors are uncorrelated, additive, normally distributed with zero mean, the inverse of this matrix reduces to the reciprocal variance matrix

$$\mathbf{\Psi}^{-1} = \text{diag}\Big[ \Psi^{-1}(i) \quad \Psi^{-1}(i+1) \quad \dots \quad \Psi^{-1}(i+R-1) \Big]_{(JR \times JR)} \tag{27}$$

with

$$\Psi^{-1}(r) = \text{diag}\Big[ \sigma_{Y_1}^{-2}(t_r) \quad \sigma_{Y_2}^{-2}(t_r) \quad \dots \quad \sigma_{Y_J}^{-2}(t_r) \Big]_{(J \times J)} \tag{28}$$

where $\sigma_Y(t)$ is the standard deviation of measurement $Y$ at time $t$. In this form, this matrix effectively scales the difference between the measurements and numerical model in such a way that if the error is less than the measurement uncertainty, the least-squares objective function contribution is reduced; but, if it is larger than the measurement uncertainty, the least-squares contribution is increased. If off-diagonal elements of $\mathbf{\Psi}$ are included, then additional contributions of $2\Psi_{kl}^{-1}(Y_k - T_k)(Y_l - T_l)$ for $k \neq l$ are added to the least-squares objective function to similarly weight contributions according to expected correlated deviations of the measurements.

In the present implementation, $\mathbf{\Psi}^{-1}$ is defined to be the identity matrix unless an override is provided by the user. It is shown by Beck et al.[8] that $\mathbf{\Psi}^{-1}$ cancels out under the "standard" measurement assumptions unless regularization terms are present. If regularization is present, neglecting $\mathbf{\Psi}$ will simply change the effective value of the regularization parameter. However, if the variance of measurements are not equal in space or time, then including $\mathbf{\Psi}$ will result in modeled errors at more precise measurements being weighted heavier in the objective function and potentially reduce the reconstruction bias at those measurements.

At the present time, very little has been done to characterize the behavior of the $\mathbf{B}$ matrix enabling the trial function capability. For all presented results, $\mathbf{B} = \mathbf{0}_{(P\times P)}$ and $\tilde{\mathbf{q}}^f = \mathbf{0}_{(P\times 1)}$.

### III.C. Algorithm Variants

The algorithm presented here can be considered a unified method since it has a number of parameters, set by the user, which significantly alter the behavior of the algorithm and yield performance ranging from pure whole domain Gauss-Newton to Beck's Future Time algorithm.

- **Length of the future time window**. This determines the length of time covered in a specific local solution. It affects how many measurements will inform this reconstruction step, and what period of time over which sensitivity coefficients must be calculated.
- **Discretization of BC interval times**. Determines how many boundary condition values must be estimated and how many measurements will inform each boundary condition interval. Note that having multiple measurements inside a boundary condition interval implicitly adds stabilization to the algorithm in the same manner as Beck's future time algorithm.
- **The number of considered BC intervals (Parameter $F$)**. If the user specifies $F$ to be lower than the number of intervals covered by the future time window, then it will ignore the intervals $f > F$. The $f = F$ interval will be assumed to apply all the way to the end of the future time window (equivalently, all intervals $f > F$ will be assumed equal to $F$). The step sensitivity coefficient is evaluated for interval $F$ whereas pulse sensitivity coefficients are calculated for $f < F$.
- **How many estimated BC intervals are retained after a local solution**. This algorithm allows the user to retain a subset of the estimates calculated in a given local solution. In this way, solutions earlier in the future time window, which have sensitivity coefficients that have had time to develop, are retained as they are likely more accurate, and estimates at later times go towards improving the initial guess of those intervals.
- **Regularization parameters: $\alpha$ and $W_f$**. If $F > 1$, these options control solution stabilization beyond that implicitly included if multiple measurements are included in the defined solution intervals.

All of these options can be varied for different local solutions in a global problem if it is desirable for a particular reconstruction.

#### *III.C.1. Beck's Future Time Algorithm*

The Future Time (FT) algorithm of Beck et al.[8] can be obtained by setting $F = 1$, discretizing the boundary condition intervals to match the measurement times, and setting the length of the future time window to the appropriate number of 'future times' in Beck's algorithm. Since $F = 1$, the heat flux is assumed constant through the future time window (equivalently, the boundary condition intervals on the future time window are assumed equal to the first). A couple steps of this algorithm are illustrated in timeline form in Figure 3.

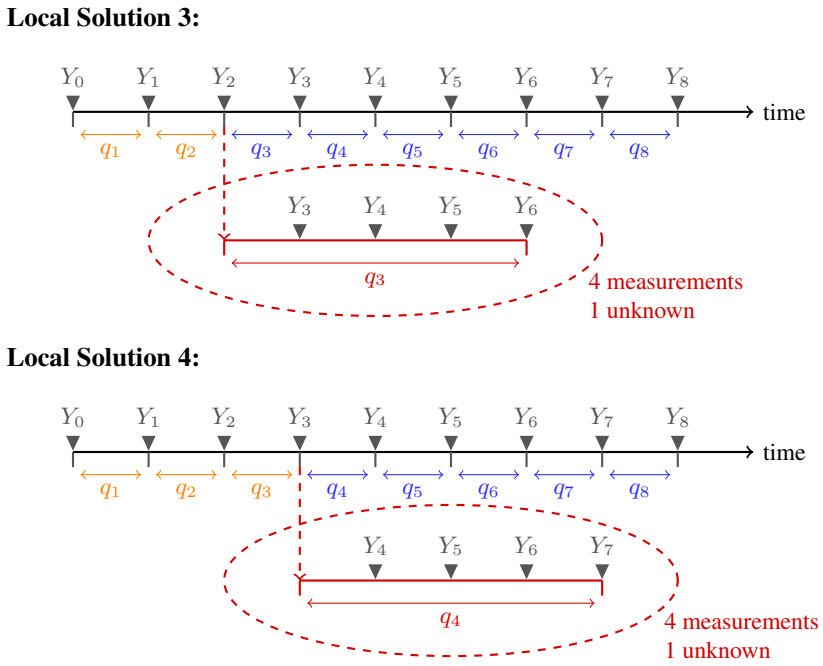



Figure 3: Timeline illustrating two local solutions of a Future Time reconstruction.

### *III.C.2. Whole Domain Algorithm*

On the other end of the spectrum, a pure Whole Domain (WD) algorithm solution can be obtained by setting $F = M$ to include all intervals, discretizing the boundary condition intervals to match the measurement times, and setting the length of the future time window to the whole problem domain. Regularization will need to be specified to stabilize the solution, and all solution values must be retained. This is illustrated in timeline form in Figure 4.

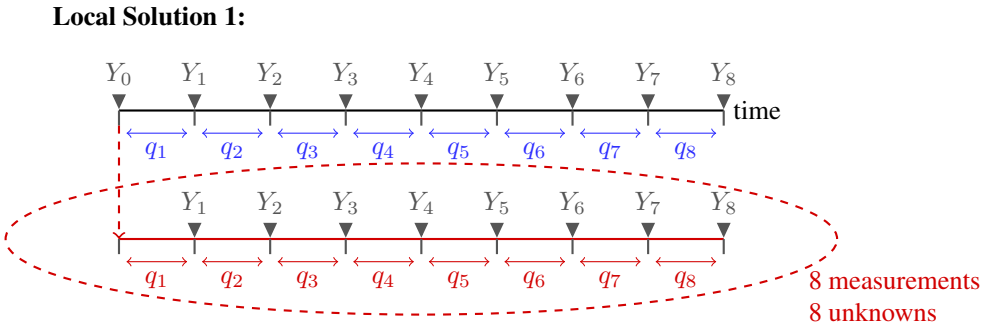


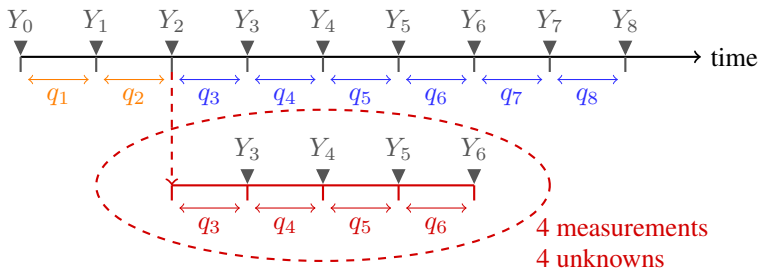

Figure 4: Timeline illustrating the local solution of a pure Whole Domain reconstruction.

### *III.C.3. Sequential Subdomain Algorithm*

An example illustrating the proposed hybrid algorithm, which we will refer to as the *Sequential Subdomain* (SSD) algorithm, is obtained by setting $F$ to a value in the range $1 < F < M$, setting the length of the future time window long enough to cover the $F$ boundary condition intervals, and setting the terms retained in each local solution to a desired value (typically 1). In Figure 5, the values of $q_3$ through $q_6$ are estimated in Local Solution 2, but only $q_3$ and $q_4$ are retained. The initial guess for $q_5$ and $q_6$ are updated based on the results of Local Solution 2, but they are re-estimated in Local Solution 3.

This algorithm is proposed to address the concerns raised in Section II.D. It was shown that the minimum length of the future time window is a constraint of the material properties, the measurement distance from the boundary, and the accuracy of the measurements. Given that constraint, Figure 3 shows why the future time algorithm can perform poorly

**Local Solution 2:**




**Local Solution 3:**

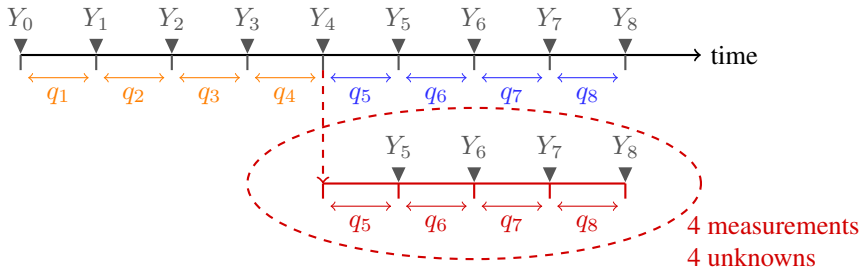



Figure 5: Timeline illustrating two local solutions of a Sequential Subdomain reconstruction with two retained terms.

if BC intervals necessary to capture the relevant behavior are short compared to the future time window: the estimated flux is essentially going to be the 'average' of the true solution over all solution intervals in the future time window. To get around this, the SSD algorithm draws on the whole domain approach of evaluating pulse sensitivity coefficients to the end of the future time window for all solution intervals in the future time window and performs a simultaneous estimation of all of the boundary condition values touched. Since the true 'whole domain' is not considered, the solution estimates at later times are not necessarily very accurate due to the shorter development time allowed for their sensitivity coefficients. As a result, these estimates are not retained for the final solution. They are, however, used as an improved initial guess for subsequent local solutions. Using this approach, many problems can benefit from greatly improved reconstructions as compared to the future time algorithm and greatly reduced computation times compared to the whole domain algorithm.

The SSD algorithm appears similar in several aspects to Lamm's modified local Tikhonov regularization algorithm[24,25]. Both methods utilize Tikhonov regularization on restricted 'local' domains and allow the extent of these local domains to vary in size throughout the overall problem domain. The SSD seems to differ from Lamm's algorithm in that the SSD considers only data at future times relative to the component being estimated (as opposed to applying the Tikhonov regularization considering data on an interval centered on the component being estimated), it estimates multiple components on the local domain, and it performs the overall estimation on a single sequential pass through the time domain as opposed to iterating with multiple passes over the full domain[b]. There is quite a bit of material in the applied math community on problems of this nature, and more investigation of the literature is necessary to establish the novelty of the proposed algorithm.

### III.D. Computational Cost

This implementation provides considerable flexibility for reconstruction problems, but each option has an impact on the overall computational effort required. Two metrics are defined to compare the computational cost of different modes: linear system cost, and sensitivity coefficient cost.

Linear system cost describes the computational effort required to solve Equation 11 once it has been defined. General dense matrix algorithms such as LU or QR decompositions can be used to solve such a system. Generic implementation of these methods usually require order $\mathcal{O}(n^3)$ operations, though more efficient algorithms can provide some advantage. Equation 11 is solved at the local solution level, so the linear system cost scales by the unknowns in a local solution.

[b]Note: *INHEAT* can take a user-provided initial guess, so this last point could easily be implemented to perform 'multiple-passes' on the data.

The FT algorithm requires the solution of $M$ local solutions; however, it generally does not require any non-linear iteration, so the total linear system cost of the FT algorithm will be $\mathcal{O}(MN^3)$. The WD algorithm only requires a single local solution, but it will likely require iteration. If $\nu_{WD}$ iterations are assumed, the total linear system cost for the WD algorithm will be $\mathcal{O}(\nu_{WD}(NM)^3)$. The SSD algorithm will likely require some non-linear iteration, though it is hoped that the sequential evaluation will reduce the number of non-linear iterations relative to the WD algorithm, so $\nu_{SSD} < \nu_{WD}$ iterations will be assumed. The number of local solutions needed in the SSD algorithm is $M/N_{ret}$, where $N_{ret}$ is how many solution intervals are retained from each local solution. With these assumptions, and neglecting that the linear system will be smaller for local solutions containing the end of the global problem, the linear system cost of the SSD algorithm is $\mathcal{O}(\nu_{SSD}\frac{M}{N_{ret}}(NF)^3)$.

Despite the cubic scaling of the linear system cost, far more time will generally be spent evaluating sensitivity coefficients (unless a problem is linear and special accounting is made for that fact). Sensitivity coefficients are assumed to be computed by a finite-difference approximation, which means a number of direct material response simulations must be computed using appropriately perturbed boundary conditions. Depending on the specific problem, the material response evaluation could be cheap or very expensive. To generalize the cost of the methods, the sensitivity coefficient cost will be measured by *accumulated simulated time*. This measures the total amount of material response time integrated over the course of the reconstruction, both for sensitivity coefficient evaluation and temperature residual evaluation. For example, evaluating a finite-difference sensitivity coefficient on a $5\,\mathrm{s}$ future time window using central differences will require two evaluations of the $5\,\mathrm{s}$ domain (each referred to as a *future time track*) with perturbed boundary conditions, so $10\,\mathrm{s}$ is added to the accumulated system time for every non-linear iteration. This metric is not a direct measure of wall-time required to compute a reconstruction as wall-time per unit of simulated time will be problem specific. This metric also does not take into account differing parallel efficiencies of different reconstruction approaches. However, it provides a means of comparing to a first-order the differing costs of reconstruction algorithms.

The accumulated simulated time of any local solution is the number of required tracks times the length of each track. In the present implementation, one track must be evaluated in each local solution for residual evaluation ($\mathbf{T}^*$ in Eq: 13), and two tracks for every unknown for sensitivity coefficient evaluation (one each for the positive and negative perturbations). For a generic solution, this yields an accumulated simulation time of $(1+2NF)t_{FT}$, where $t_{FT}$ is the length of the future time window. As with the linear system time, the cost of a single local solution is scaled by the number of local solutions required, which gives a total accumulated simulated time of

$$t_{AST} = \nu\frac{M}{N_{ret}}(1+2NF)t_{FT}. \tag{29}$$

For the FT algorithm, this simplifies to $M(1+2N)t_{FT}$ as $\nu$, $N_{ret}$, and $F$ are all 1. For the WD algorithm, it simplifies to $\nu_{WD}(1+2NM)t_{full}$ as only one local solution is used, $F = M$, and $t_{FT} = t_{full}$. Note that in this implementation, the WD algorithm does not take advantage of the parabolic nature of the problem. A more optimal WD implementation could evaluate sensitivity coefficients using tracks that start at the first time for which the corresponding unknown can affect the solution. If a uniform discretization of the boundary conditions is assumed, this could theoretically reduce the cost of sensitivity coefficient evaluation by a factor of 2. The SSD algorithm accumulated simulated time takes the form of Equation. 29 with $\nu = \nu_{SSD}$. These costs are summarized in Table 2.

Table 2: Summary of cost metrics for different algorithm modes.

| Algorithm Mode | Linear System Cost | Accumulated Simulated Time |
|---|---|---|
| Future Time | $\mathcal{O}(MN^3)$ | $M(1+2N)t_{FT}$ |
| Whole Domain | $\mathcal{O}(\nu_{WD}(NM)^3)$ | $\nu_{WD}(1+2NM)t_{full}$ |
| Sequential Subdomain | $\mathcal{O}(\nu_{SSD}\frac{M}{N_{ret}}(NF)^3)$ | $\nu_{SSD}\frac{M}{N_{ret}}(1+2NF)t_{FT}$ |

The formulae in Table 2 are not easily comparable due to the variable algorithm parameters. For some combinations of parameters, the SSD algorithm can provide significant computational savings over a pure WD algorithm; however, with other combinations of parameters, it is possible for the SSD to cost as much or more than a WD solution. For the sake of example, assume a problem with $N = N_{ret} = 1$ with uniform distribution of boundary condition intervals such that $t_{ft} \approx \frac{t_{full}}{M/F}$. With this, the ratio of accumulated simulated time of the WD and SSD algorithms is seen to reduce to

$$\frac{t_{AST,SSD}}{t_{AST,WD}} = \frac{\nu_{SSD}F(1+2F)}{\nu_{WD}(1+2M)}. \tag{30}$$

If $\nu_{SSD} < \nu_{WD}$ and $F^2 < M$, the SSD can be expected to provide a computationally cheaper result. However, if $F$ is too large relative to $M$, the SSD reconstruction could rapidly become the most expensive.

If we consider a numerical example, Mahzari et al.[13] considered a time domain of $t_{full} = 130\,\mathrm{s}$ with a reconstruction frequency of $1\,\mathrm{Hz}$ ($N = 1$, and $M = 130$) and $\nu = 200$ non-linear iterations for the MSL reconstruction. This means that a whole domain reconstruction requires $6.786 \cdot 10^6\,\mathrm{s}$ of accumulated simulated time. If this were done with a sequential method with a future time of $t_{FT} = 5\,\mathrm{s}$, the accumulated simulated time would be $1.95 \cdot 10^3\,\mathrm{s}$–three orders of magnitude less. The SSD algorithm in this instance, with $F = 5$ and the same number of non-linear iterations, would have an accumulated simulated time of $1.43 \cdot 10^6\,\mathrm{s}$. However, if the future time window were $15\,\mathrm{s}$ with a corresponding $F = 15$, this would jump to $12.1 \cdot 10^6\,\mathrm{s}$. While the accuracy of an SSD reconstruction relative to a WD reconstruction must be taken into account, the potential for cost savings with SSD is clear.

## III.E. Implementation

### *III.E.1. CHAR Description*

The IHCP algorithm described herein has been implemented as the *INHEAT* module in the *CHAR* code. *CHAR* is a 1-D/2-D/3-D material thermal response code which solves general heat transfer problems on decomposing ablators as well as non-decomposing, non-ablating TPS materials. It uses a finite-element discretization with first- and second-order implicit time integrators. The pyrolysis gas and solid energy governing equations are solved fully-coupled and fully implicit in serial or parallel. A rich and flexible suite of boundary conditions are implemented, and rigorous verification using the method of manufactured solutions is used to assure accurate implementation of the governing equations. It has been developed over several years in the NASA Johnson Space Center Applied Aeroscience and CFD Branch to improve methods of integrating products between the aerothermal environment and thermal protection disciplines. It is implemented in the C++ programming language and is built on the libMesh library[26]. A more complete description can be found in references[27,28].

### *III.E.2. Inverse Capabilities*

*INHEAT* has been developed and integrated with *CHAR* with the goal of maximizing the user's flexibility in specifying an inverse problem. Any time-varying boundary condition value that can be specified in a *CHAR* boundary condition file can be the unknown parameter estimated in an inverse run. This includes values that serve as input to runtime-evaluated expressions which themselves can be functions of time, space, 'spatial' BC input files, and temperature/pressure/location values at 'monitor points' defined at fixed or node-following locations. Any number of temperature targets can be specified and moving targets are permitted (in order to make use of data provided by isotherm tracking sensors). This places a fair bit of responsibility on the user to define a problem that can actually be solved, however this is believed to be a small cost compared to the flexibility provided. It has also been implemented in such a way as to put as few restrictions on standard *CHAR* capabilities as possible (adaptive mesh refinement is the key feature not permitted, dynamically variable time-stepping is permitted).

As the cost of computing required sensitivity coefficients can be significant, *INHEAT* has been implemented with a limited threading capability over and above *CHAR*'s own ability to solve a direct problem on multiple processors using MPI. The majority of time in an inverse solution is spent in evaluating sensitivity coefficients. *INHEAT* has been implemented to compute sensitivity coefficients using finite-difference approximations. Each direct simulation of the future time window with a perturbed BC value is referred to as a *future time track*. Evaluation of the future time tracks needed for sensitivity coefficient evaluation is an embarrassingly-parallel problem, so parallel efficiency is maximized by assigning minimum-sized 'threads' to solve the tracks in parallel as opposed to all processors working on one track at a time (which maximizes parallel communication inside the direct solver). The desired number of threads are specified, the available processors are divided up between the threads, and future time tracks are divided up between threads for evaluation. If a direct problem is too large to fit on a single processor, the threads may themselves use multiple CPUs managed by *CHAR*. At this time, *INHEAT* is not capable of dynamically adjusting the number of threads to optimally fit the specified problem. *CHAR* was also given the ability to return the state of the object to a previously saved state to permit multiple passes through the future time window without having to recompute times prior to the current local solution. It should be noted that solving sensitivity coefficients using dedicated sensitivity equations could be more computationally efficient, however this greatly complicates the addition of physical models. *CHAR* has been developed with the intention that physical models can be quickly added, so finite-difference sensitivity coefficients maintains this development paradigm.

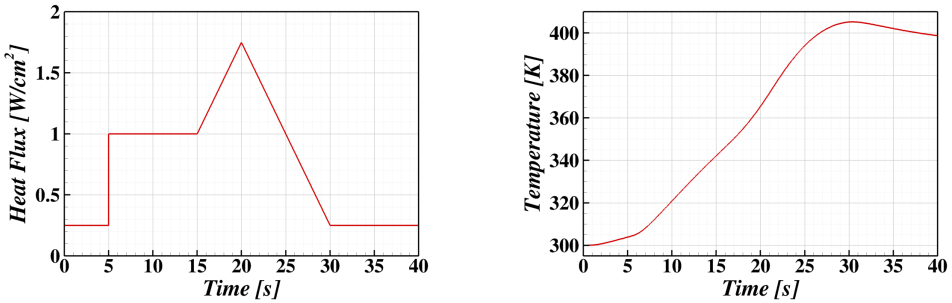
*INHEAT* has been used for several recent flight and ground test data reconstruction activities and has a small user base that has assisted with identifying bugs and providing advice on desirable input options and capabilities. It is now a standard component included with the *CHAR* distribution.

# IV. Examples

The behaviors of the different algorithms are perhaps best described by illustration. A standard linear benchmark case will be considered to demonstrate the basic function of the different algorithm modes. Next, a reconstruction of a decomposing ablator will be considered to assess the non-linear performance. Then a case representative of rapid steering jet heating augmentation will be considered to demonstrate performance when representative time-scales change during a reconstruction. Finally, a multi-component/multi-dimensional benchmark case will be shown to demonstrate the capability.

## IV.A. Benchmark Case

A 1-D semi-infinite slab of material with the constant properties ($\rho = 280\,\mathrm{kg/m^3}$, $k = 0.4\,\mathrm{W/m \cdot K}$, and $C_p = 1000\,\mathrm{J/kg \cdot K}$) is subjected to the heating profile defined in Figure 6(a) to generate artificial data for reconstruction. A single thermocouple is placed at a depth of $3.8\,\mathrm{mm}$ and is sampled at $10\,\mathrm{Hz}$. Two sets of simulated data will be considered: 'clean' data taken directly from this simulation, and 'noisy' data which is truncated to $0.2\,\mathrm{K}$ resolution, simulating a low resolution data acquisition system.




(a) Heat flux profile

(b) Temperature Data

Figure 6: Truth heating profile and target temperature data for benchmark example problem

The sensitivity coefficient at the sensor for this case peaks at approximately $5\,\mathrm{s}$, as seen in Figure 7. With this profile, the regularization parameter suggested by Equation 24 is approximately $2 \cdot 10^{-9}$. This is a relatively simple linear problem and does not stress the different algorithms, but it does demonstrate the general characteristics of typical solutions. All reconstructions assume a uniform distribution of solution intervals at $10\,\mathrm{Hz}$ to match the measurements.

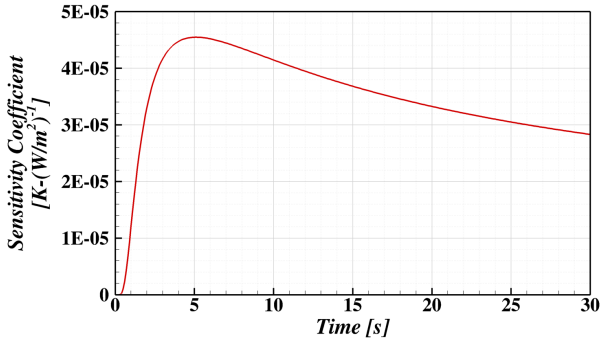
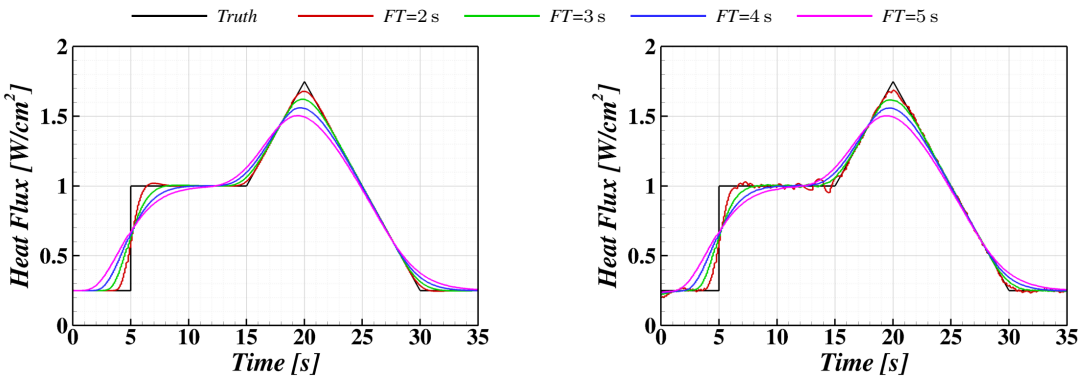



Figure 7: Pulse sensitivity coefficient for benchmark example problem

Future Time algorithm results are presented in Figure 8. Increasing the future time increases smoothing, as expected, with sharper features in the heating profile suffering as a result. The introduced noise is apparent only with the shortest future times. Since the future time window is not centered on the solution interval that it defines, this

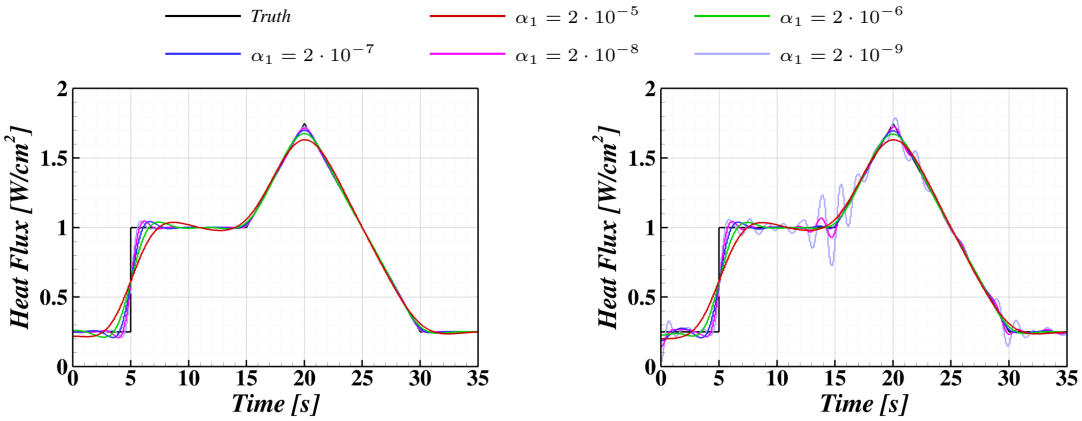
algorithm tends to 'lead' a transient heating profile, as evident by the forward shift in the peak near 20 s.




(a) Clean Data (b) Noisy Data

Figure 8: Reconstruction results of benchmark heating profile using the Future Time algorithm for several future time window lengths

The Whole Domain algorithm results in Figure 9 show some interesting contrasts with the Future Time results. First of all, even with fairly strong regularization, the peak heating is much more accurately resolved, and there is no apparent forward shift in time. The WD reconstructions appear to degrade more than the FT due to measurement noise; however, close examination of the higher values of the regularization parameter yield reasonable reconstructions.

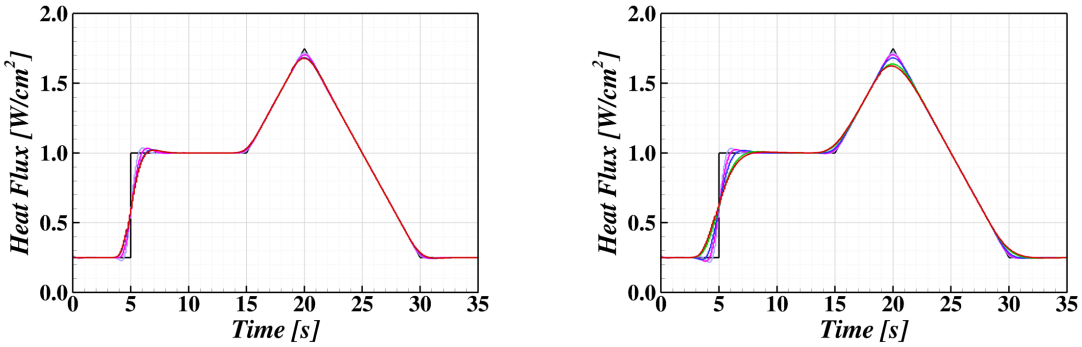



(a) Clean Data (b) Noisy Data

Figure 9: Reconstruction results of benchmark heating profile using the Whole Domain algorithm for several values of the first-order regularization factor $\alpha_1$

Sequential Subdomain results are presented in Figure 10. It is interesting to note that with a future time window of 2 s (before the peak in sensitivity coefficient), the SSD algorithm is largely unaffected by the regularization parameter. Conversely, for the longer future time window, the solution becomes defined by the regularization parameter and independent of the future time window. Given the formulation of the SSD algorithm, it is to be expected that as the first-order regularization parameter is increased, the solution components on the future time window will be driven to a constant value. This is consistent with the assumptions of the FT algorithm, so FT behavior should be recovered. Similarly, If the future time window is long enough that the sensitivity coefficients develop sufficiently and regularization is low enough that the influence of later flux components in the future time window (which have poorly developed sensitivity coefficients) is sufficiently reduced, WD behavior should be recovered. For the longer future time window, even with a larger $\alpha_1$, the time shift in the peak heating is largely removed. When noise is introduced, the SSD shows more effects of the noise than the FT results, however, for higher regularization parameters, it is a little bit smoother than the WD.

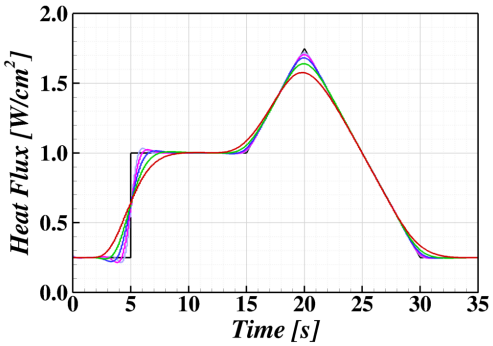
Figure 11 shows select reconstructions from each of the three algorithms on clean and noisy data. SSD reconstructions with short future time windows and larger regularization parameters very closely match FT reconstructions.



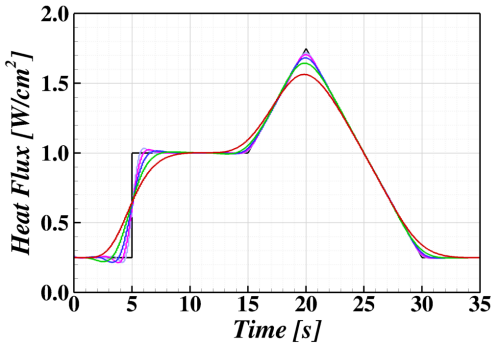

(a) Clean Data, Future Time = 2 s



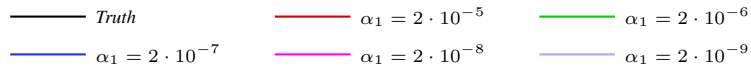

(b) Clean Data, Future Time = 3 s

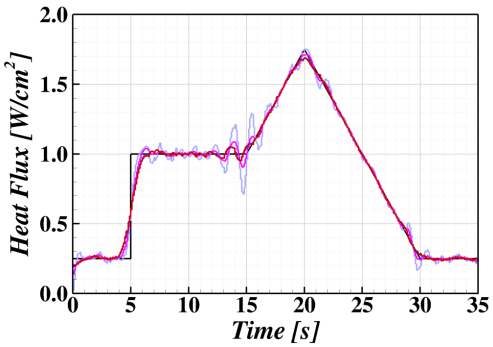



(c) Clean Data, Future Time = 4 s

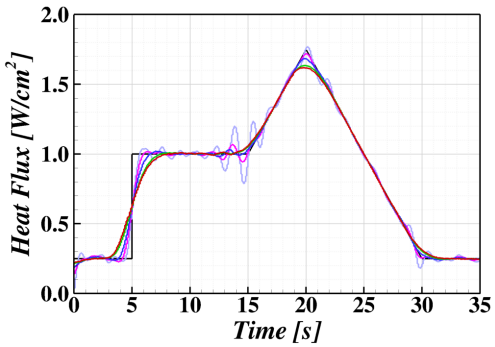



(d) Clean Data, Future Time = 5 s

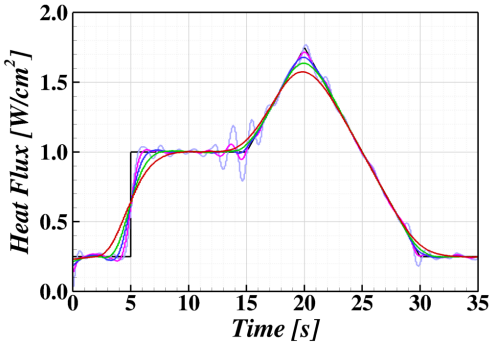


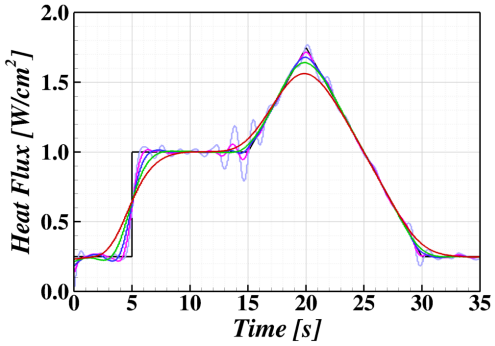

(e) Noisy Data, Future Time = 2 s




(f) Noisy Data, Future Time = 3 s




(g) Noisy Data, Future Time = 4 s



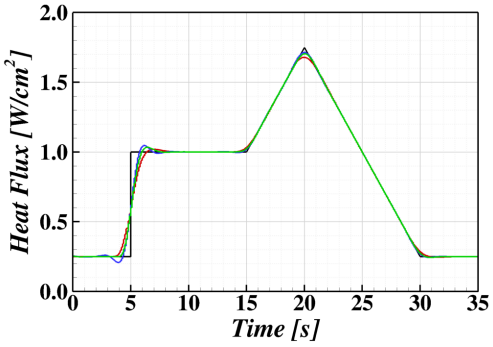

(h) Noisy Data, Future Time = 5 s

Figure 10: Reconstruction results of benchmark heating profile using the Sequential Subdomain algorithm for various values of $\alpha_1$

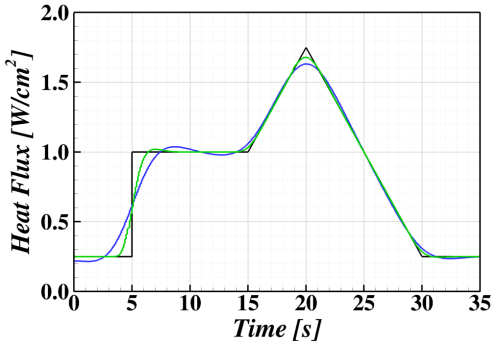
Similarly, for longer future time windows and smaller regularization parameters, the SSD algorithm closely matches WD reconstruction. For this case, the SSD algorithm seems somewhat less likely to overshoot the true solution than the WD solution. It is also apparent in this figure that the time shift introduced by the FT algorithm is mitigated in the SSD algorithm (provided the regularization is not too strong).

Quantitative measures of the accuracy of each reconstruction are presented in Figure 12. Since the true heating profile will generally not be known in practice, the metric used is the *residual temperature error* defined as $(\mathbf{Y} - \mathbf{T})^T(\mathbf{Y} - \mathbf{T})$, which is a direct measure of how well the reconstruction matches the target temperature data. Figures 12(a) and 12(b) show errors of the FT and SSD algorithms as a function of the length of the future time window for clean and noisy data, whereas Figures 12(c) and 12(d) show errors of the WD and SSD algorithms vs. the regularization parameter. The errors in the clean reconstructions behave as expected, with errors decreasing with decreasing future time window lengths and decreasing regularization parameter. SSD reconstructions with high $\alpha_1$ show a tendency to follow the pure FT reconstructions, but below about $\alpha_1 = 2 \cdot 10^{-7}$, the SSD reconstructions are independent of the length of the future time window (consistent with earlier observations). For a constant future time window, the SSD reconstructions show a tendency to converge to a common trend with sufficiently small $\alpha_1$ that decays at the same rate as the WD solutions, however there is an offset such that the SSD error at a given value of $\alpha_1$ is higher than the WD error.

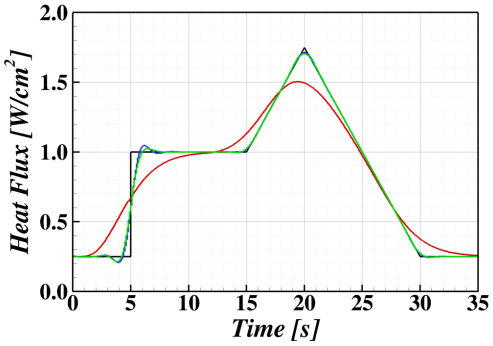
Introducing noise changes the behavior for the lowest regularization parameters, as the errors appear to asymptote to a relatively high level of error. If the residual temperature error of the noisy TC data relative to the clean TC data is computed, an error of approximately $3.93\,\mathrm{K}^2$ is obtained (and is indicated on the noisy plots). Note that for low values of $\alpha_1$, reconstructions achieve an error lower than the level of error in the data. This means that the reconstructions are are more faithfully representing the noisy data than the noisy data is representing the true signal. In other words, the reconstruction is following the noise and not the signal, and is a justification for added bias in a real reconstruction.



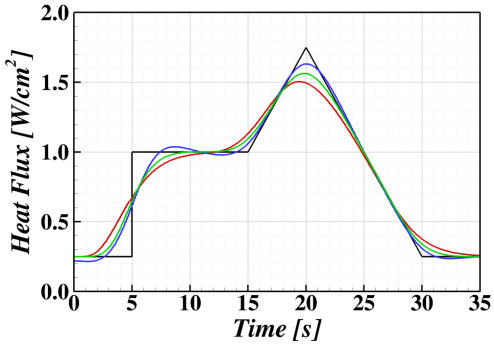

(a) Clean, Future Time = 2 s, $\alpha_1 = 2 \cdot 10^{-8}$



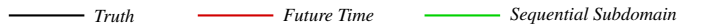

(b) Clean, Future Time = 2 s, $\alpha_1 = 2 \cdot 10^{-5}$

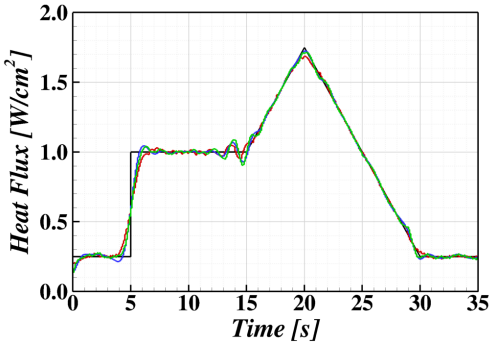



(c) Clean, Future Time = 5 s, $\alpha_1 = 2 \cdot 10^{-8}$

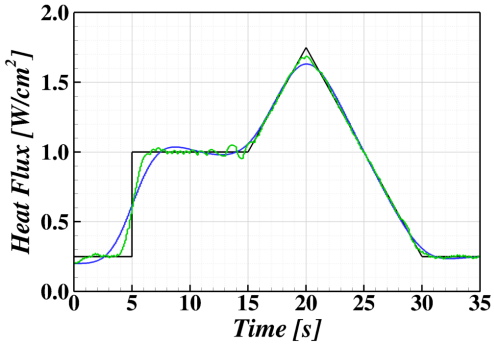



(d) Clean, Future Time = 5 s, $\alpha_1 = 2 \cdot 10^{-5}$

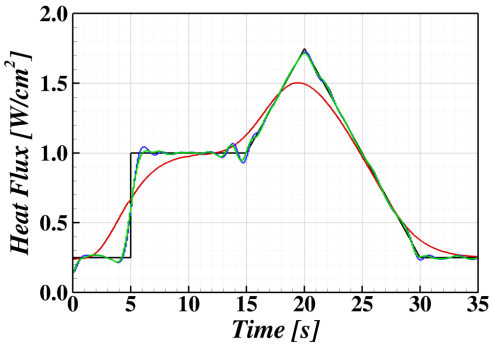


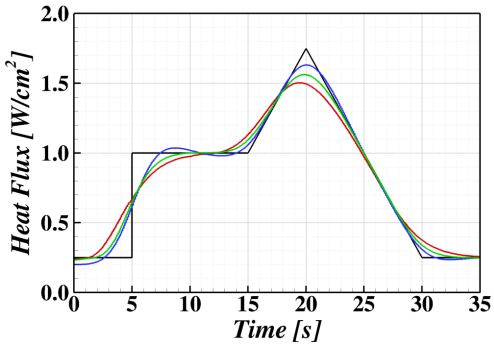

(e) Noisy, Future Time = 2 s, $\alpha_1 = 2 \cdot 10^{-8}$




(f) Noisy, Future Time = 2 s, $\alpha_1 = 2 \cdot 10^{-5}$

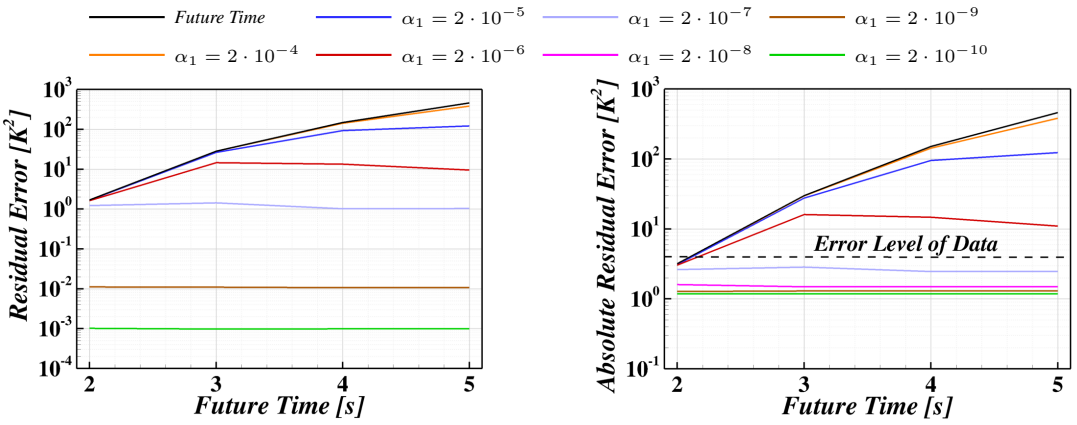



(g) Noisy, Future Time = 5 s, $\alpha_1 = 2 \cdot 10^{-8}$



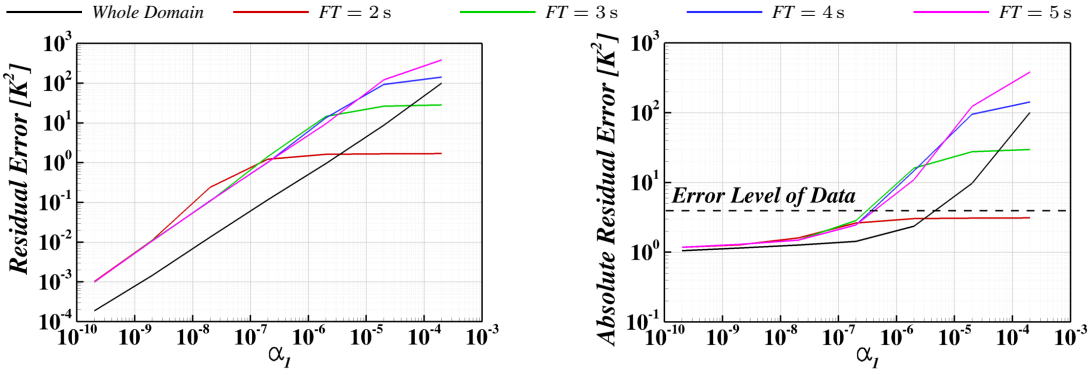

(h) Noisy, Future Time = 5 s, $\alpha_1 = 2 \cdot 10^{-5}$

Figure 11: Comparison of algorithms on benchmark heating profile test case




(a) Clean Data

(b) Noisy Data




(c) Clean Data

(d) Noisy Data

Figure 12: Residual error in benchmark reconstructions for FT and SSD algorithms.

### IV.B. Decomposing Ablator Case

The problems considered so far have been linear and iteration has not been required of the SSD or WD algorithms. Recall that an inverse problem will be non-linear if the sensitivity coefficients depend on the boundary condition solution. Sequential algorithms such as the FT algorithm generally do not require iteration as the sequential estimates provide an initial guess of the boundary condition solution sufficiently close to the correct value that the sensitivity coefficient evaluation is accurate. The SSD algorithm was designed to take advantage of this to reduce the number of required iterations in a non-linear problem.

To demonstrate the non-linear performance of the SSD algorithm, a 1-D decomposing ablator will be considered. Temperature-dependent thermal properties introduce a degree of non-linearity; however, decomposing ablators experience a change of state in response to heating that further complicates the problem. As illustrated in Figure 13, a decomposing ablator starts out as a virgin composite material containing volatile and non-volatile constituents. After sufficient heating, volatile constituents thermally decompose into pyrolysis gas in an endothermic reaction. The pyrolysis gas percolates through the remaining non-volatile material (referred to as char) and out of the domain. The thermal properties of the virgin and char states are typically quite different (which is further compounded if the contribution of percolating pyrolysis gas is considered), and a mass-weighted interpolation of properties is generally used in the decomposition zone. All of these effects result in a highly non-linear thermal response, and sensitivity coefficients will have to be evaluated from perturbations very near the correct solution values in order to be accurate.

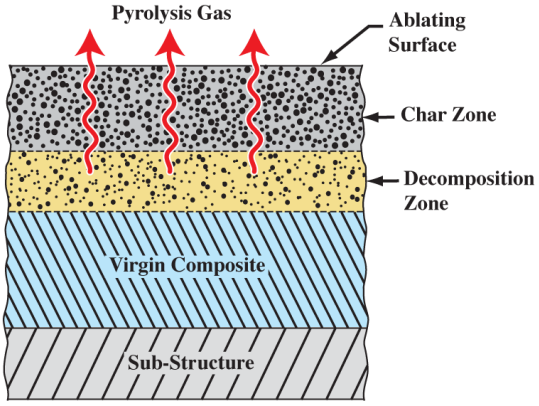



Figure 13: Regions of interest in a charring ablator. Obtained from reference[29].

The 1-D domain in this example consists of $40\,\mathrm{mm}$ of TACOT[20] on $3.175\,\mathrm{mm}$ of aluminum. TACOT is a fictitious low density charring ablator model that was developed to permit open publication of ablation modeling results; however, the model responds similarly to models for real materials such as PICA and AVCOAT and should be sufficiently realistic to assess the performance of *INHEAT* on those materials. An array of thermocouples are placed below the surface at depths of $3.175\,\mathrm{mm}$, $6.35\,\mathrm{mm}$, and $9.525\,\mathrm{mm}$. Surface recession will be neglected in this example to avoid issues of missing data due to thermocouple burn-through.

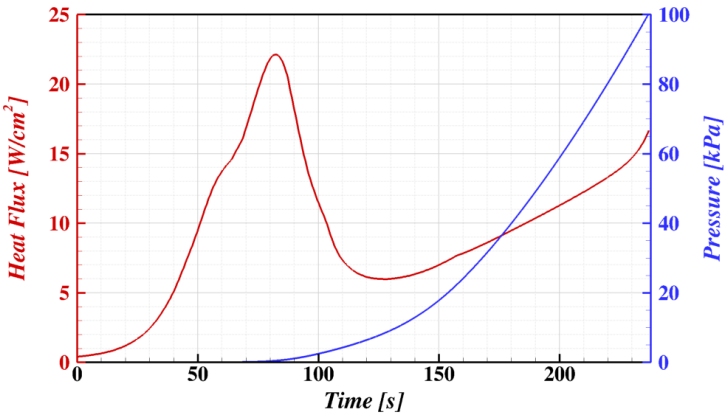



Figure 14: Heat flux and pressure boundary conditions for TACOT example problem.

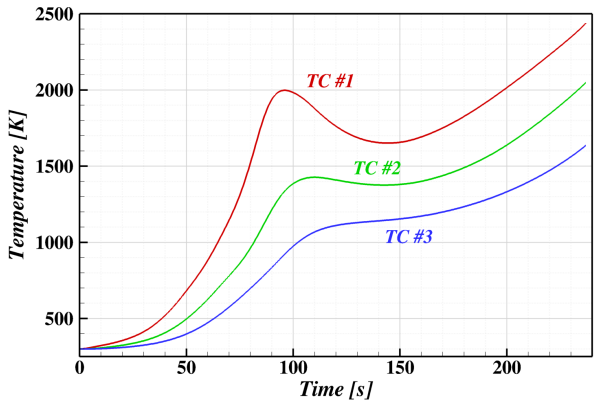


Figure 15: Simulated measured temperatures for TACOT example problem.

Figure 14 shows the heating and pressure profile used as the truth reference in this example. The simulated TC traces that result are shown in Figure 15. Representative sensitivity coefficients for this problem are shown in Figure 16. Sensitivity coefficients for pulses starting every $50\,\mathrm{s}$ are shown to illustrate the extent to which the coefficients change through the entry. The heating profile and the decomposition state of the material at depth during the simulated entry (with TC depths indicated) are shown on the same time scale to provide context. Several observations can be made: first, note that TC 1 sensitivity coefficients peak after approximately $4\,\mathrm{s}$ of development time when the TC is in virgin material, and this reduces to approximately $3\,\mathrm{s}$ when the TC is in charred material. TC 2 shows a similar trend with the peak moving from $13\,\mathrm{s}$ in virgin to $10\,\mathrm{s}$ in char. TC 3 peaks after about $25\,\mathrm{s}$ but does not get into enough char by the end of the trajectory to see much of a shift in time. Secondly, the amplitude of the coefficients vary through time, with the TCs being generally more sensitive in virgin material than in charred material. The sensitivity coefficients that start at $50\,\mathrm{s}$ have a somewhat depressed nature because of the ablator decomposition process. Decomposition is typically modeled as a temperature driven endothermic reaction[30]. An increase in heat flux to a decomposing material element will increase the rate of decomposition, which leads to more heat being carried away in the pyrolysis gas and proportionally less heat available to heat up the material element and those below it. The TACOT material model assumes a two-reaction decomposition model, which leads to the two local minima in the TC 1 response (at shallow depths, the reactions are advancing fast enough for their effects to appear separate). Finally, though it may be intuitively obvious, it is worth repeating that a different heating profile would lead to decomposition occurring at different times and rates, which would yield significantly different sensitivity coefficients, making this inverse problem highly non-linear.

*INHEAT* is used for the reconstructions presented. Identical *CHAR* settings are used in all sensitivity coefficient evaluations. The TACOT domain is represented by 300 uniformly spaced elements and $0.5\,\mathrm{s}$ time steps are used in time integration. The domain was initialized to $300\,\mathrm{K}$ and $20\,\mathrm{Pa}$. *INHEAT* was set to use un-scaled sensitivity coefficients with regularization applied to the solution magnitude. All unknown boundary conditions values were initialized to $10\,\mathrm{W/m^2}$ with a minimum finite-difference perturbation of $5\,\mathrm{W/m^2}$ (nominal perturbations are $0.1\%$ of the boundary condition value).

Reconstructions using the Future Time algorithm are shown in Figure 17. Figure 17(a), reconstructions based on TC 1 are shown for several values of the future time window. As expected from previous examples, longer future time windows lead to overly-smoothed peaks. Figure 17(b) shows reconstructions based on the three different TCs holding the future time window constant at $8\,\mathrm{s}$. In this instance, the results may seem counter-intuitive, as the deepest TC produces the most accurate reconstruction. Consider, however, the differences in development time of the sensitivity coefficients at each TC, and it becomes apparent that $8\,\mathrm{s}$ is too long for the shallow TC but more appropriate for the deeper ones.

Reconstructions using the Whole Domain algorithm are shown in Figure 18. Given the dependance of the sensitivity coefficients on the unknown solution, the regularization parameter was specified using Equation 24 with $c$ given by `-a_1` (the *INHEAT* input nomenclature) as $1^{st}$-order regularization is used. Figure 18(a) shows the reconstructions obtained from each of the three TCs with `a_1` $= -1.0$. Visible differences are apparent, with the regularization having a much stronger effect on reconstructions based on the deeper TCs. Figure 18(b), in contrast, shows all three reconstructions collapsing to the resolution of the plot with `a_1` $= -0.001$ (three orders of magnitude less regularization).

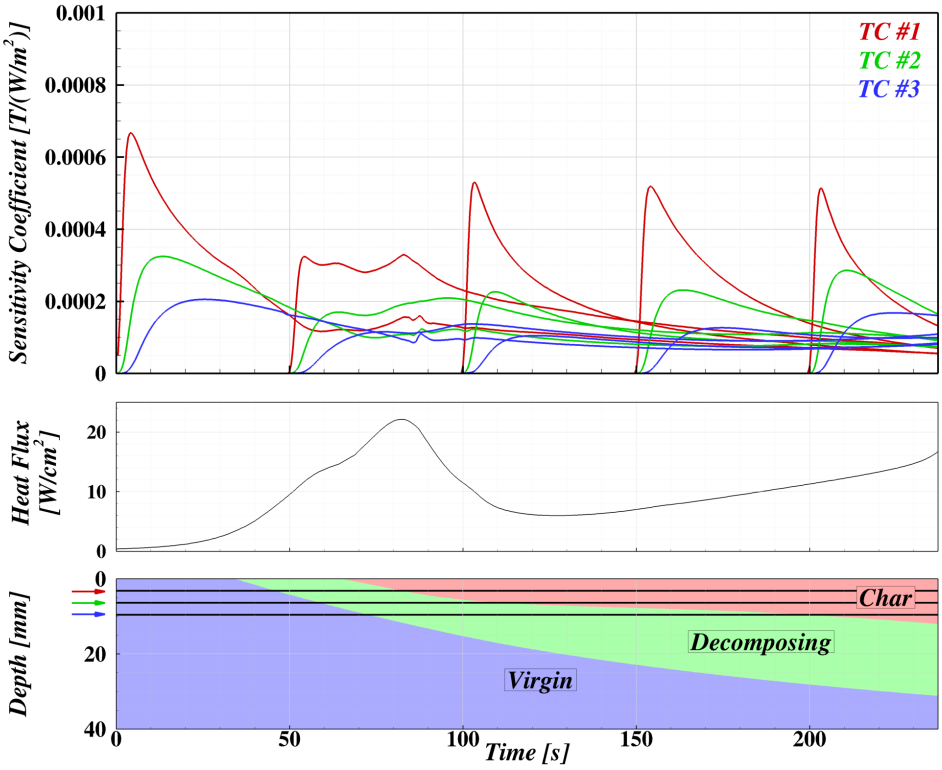



Figure 16: TACOT sensitivity coefficients for 1 s pulse perturbations from the true environment relative to applied flux and material decomposition state.

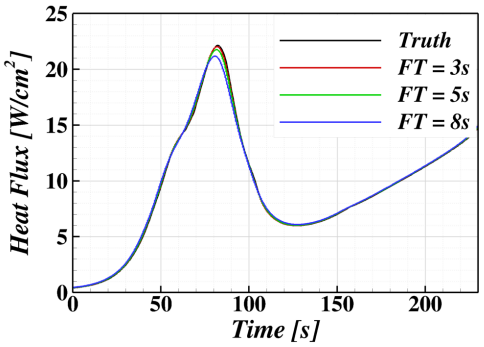
Reconstructions using the Sequential Subdomain algorithm are shown in Figure 19. All successful reconstructions using future time windows of 3 s, 5 s, and 8 s, each with $1^{st}$-order regularization scaling parameters of $-1.0$, $-0.1$, $-0.01$, and $-0.001$, are shown on the plot with all virtually collapsing onto a single line at this resolution. Some combinations of future time window and regularization scaling parameter (noted later) yielded insufficient smoothing and were either clearly unstable or resulted in *CHAR/INHEAT* aborting when material property table bounds were exceeded. While the individual FT and WD algorithms show some sensitivity to the input parameters, it is perhaps surprising, and desirable, that the SSD is much less sensitive to the input parameters.

The maximum instantaneous relative solution error, $\left\| \frac{q_{m,reconstructed}}{q_{m,truth}} - 1 \right\|_\infty$ over the first 200 s[c] of the trajectory, is shown in Table 3. This table shows variation in the error metric as a function of the future time window length across the horizontal dimension (with the WD results in the red-highlighted column) and the regularization scaling parameter in the vertical dimension (with the FT results in the gray-highlighted row), with a separate table for reconstructions based on each TC. In general, trends are as would be expected: increasing smoothing (either by increasing the future time window or regularization parameter) increases the error for this case with no measurement noise. Even without explicitly added noise[d], reconstructions based on the deeper TCs became unstable and failed to complete for FT and SSD reconstructions with shorter future time windows. A stark observation from these tables is that the SSD–if it converges–produces a result that is more accurate than WD and FT solutions with the same reconstruction inputs. Furthermore, there is less variation in the accuracy of the results across the same spread of reconstruction inputs for the SSD reconstructions than for the FT and WD algorithms.

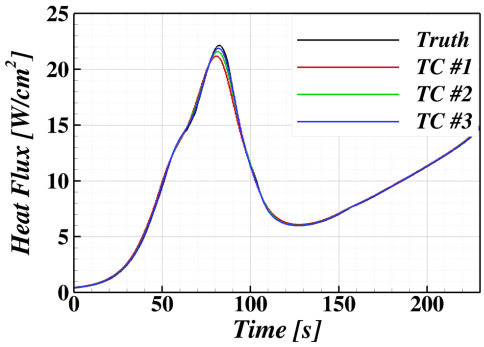
Part of the reason that the SSD performs perhaps better than the other algorithms is that the actual regularization parameter derived from Equation 24 is different for the SSD and the WD algorithms for the same `a_1`. Because $\alpha_1$ is made a function of the sensitivity coefficients, differences in the development time for the different algorithms (and different future time windows within the SSD) can lead to different values of $\alpha_1$. Furthermore, the SSD will derive $\alpha_1$ based only on sensitivity coefficients within the future time window, whereas the WD will derive $\alpha_1$ based on the whole

[c]The final 37 s of the trajectory were discarded for this assessment as the end of the reconstructions can become erratic as later coefficients do not have time to fully develop.

[d]Note that even though explicit noise is not added, there is still some noise introduced by rounding the measurement data to 10 significant digits as was done in this example

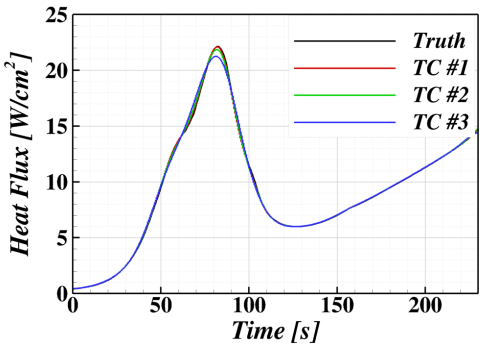



(a) TC 1

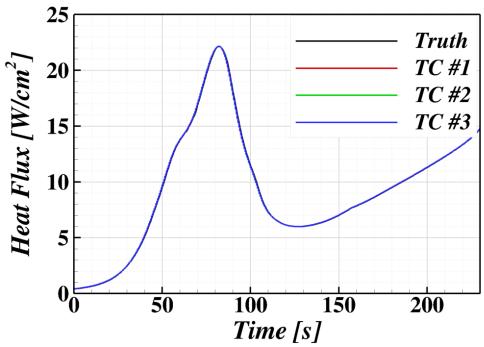



(b) Future Time = 8 s

Figure 17: Reconstructions of TACOT test case using Future Time algorithm




(a) a_1 $= -1.0$




(b) a_1 $= -0.001$

Figure 18: Reconstructions of TACOT test case using Whole Domain algorithm

time domain. As is seen in Figure 16, the peak values occur early and drive the values for the WD reconstructions, but the SSD can pick a value based on the solution intervals it is dealing with for a particular local solution. The actual values of $\alpha_1$ used in the reconstructions using a_1 $= -1.0$ are shown in Figure 20. For all three TCs, the SSD regularization parameter is one to three orders of magnitude lower than the WD regularization parameter, with greater discrepancies at deeper TCs where the shorter future time windows do not allow the sensitivity coefficients to fully develop. The SSD regularization parameter is also seen to vary by an order of magnitude or more during the course of a reconstruction. Note that all presented results were taken from the final non-linear iteration of a given local solution. As the sensitivity coefficients change through the non-linear iterations, different $\alpha_1$ values are used in each iteration. Understanding this fact, it perhaps makes more sense to compare the accuracy of the SSD reconstruction of TC 1 with an 8 s future time window and a_1 $= -1.0$ to the WD reconstruction of TC 1 with a_1 $= -0.1$ which brings the accuracy of the SSD more in line with observations from the previous section.

A goal of the SSD was to reduce the computational expense of non-linear inverse problems, so we will now consider the non-linear convergence behavior of the SSD and WD implementations in *INHEAT*. Figure 21 shows the non-linear convergence of the residual temperature error and the normalized update vector $\|\Delta\mathbf{q}\|_2$ (normalized by the value at the first iteration) for the WD reconstructions. All were run for 30 iterations regardless of convergence level to show the full behavior. Both plots show the error metric at the next iteration plotted against the error metric at the current iteration. As the Taylor series approximation for the modeled temperature (Equation 10) which introduces the sensitivity coefficients that form the Jacobian matrix is $2^{nd}$-order accurate, it is expected that points on this plot for a solution in the asymptotic range with exact Jacobians should move down and to the left parallel to the $2^{nd}$-order reference line until the precision limit is reached.

For the residual temperature convergence in Figure 21(a), the early iterations loosely follow the $2^{nd}$-order reference line until the limit allowed by the regularization is reached, at which point the iterations stall (denoted by points on the diagonal of the plot). It is reasonable that the final converged level of residual error is different according the magnitude of the regularization scaling parameter as the residual temperature error is only a part of the non-linear

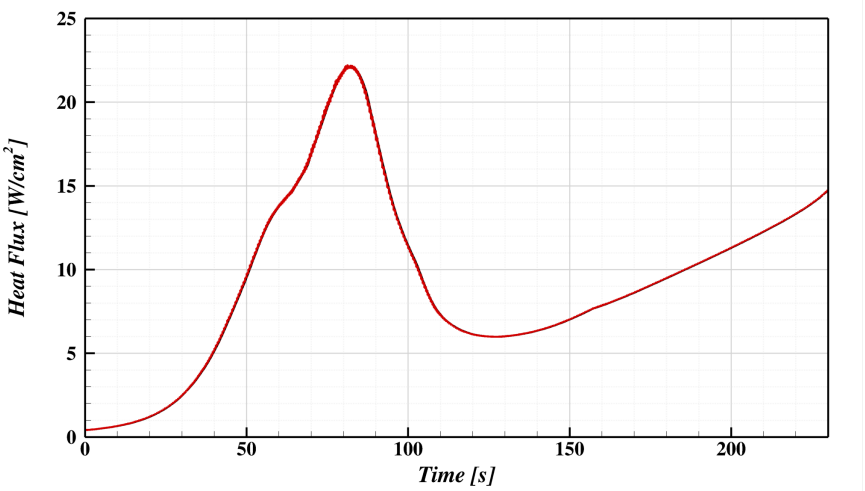
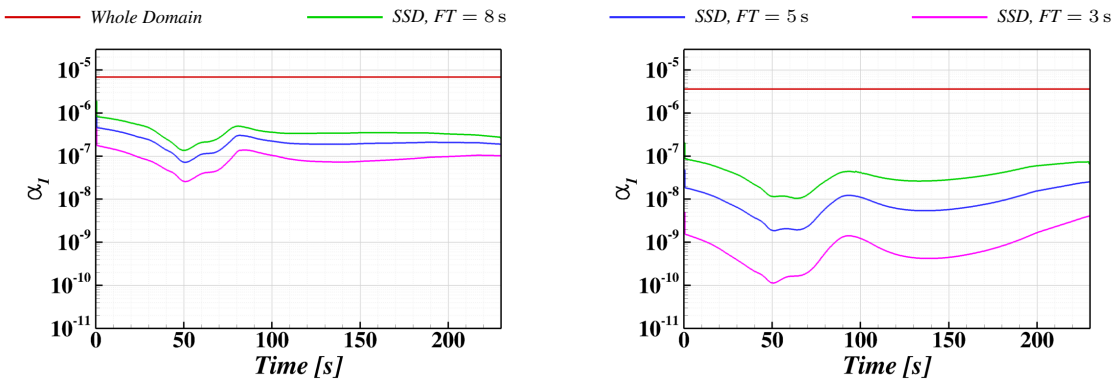



Figure 19: Reconstructions of TACOT test case using Sequential Subdomain algorithm. All successful solutions are included, showing insensitivity to reconstruction parameters.




(a) TC 1 (b) TC 2

Figure 20: Regularization parameter value for WD and SSD reconstructions for $\texttt{a\_1} = -1.0$.

system residual (Equation 13) which is actually being driven to zero in the iteration loop. The normalized update vector trend in Figure 21(b), however, does not show any significant difference in convergence behavior based on the regularization scaling parameter. Similar near $2^{nd}$-order behavior is seen in the normalized update vector, however all solutions appear to have a larger error at the second iteration than the first (due to the poor initial guess), before beginning to converge at nearly $2^{nd}$-order rates. Varying the size of the finite-difference perturbations has a very slight impact on the convergence rate. Neither error metric shows inconsistent or distinct behavior for reconstructions based on different TCs.

For local solutions in an SSD reconstruction, it is seen that very few local solutions require more than three iterations to converge. Figure 22 shows the number of iterations required of each local solution to reach different levels of convergence (defined by $\left\| \frac{\Delta \mathbf{q}}{\mathbf{q}} \right\|_2$ dropping below the specified level). It indicates that for all local solutions in this problem, three iterations are all that is required to converge the solution to better than 6 digits. The impact of this on the computational effort required to complete the reconstruction is shown in Figure 23. This figure shows the accumulated amount of time that must be computed by the direct solver to evaluate the sensitivity coefficients for each of the reconstructions presented in this section. Figure 23(a) compares the FT, SSD, and WD algorithms as a function of the length of the future time window. For this case, the SSD required fewer simulated seconds than the WD algorithms when both algorithms take the same number of non-linear iterations (30). Figure 23(b) shows the simulation time in sensitivity coefficients if convergence tolerances are used to limit iteration. The WD solutions plateau to a $\left\| \frac{\Delta \mathbf{q}}{\mathbf{q}} \right\|_\infty$ of between $10^{-5}$ and $10^{-6}$ in 6 iterations, so for tolerances of $10^{-5}$ and greater see a factor of 5

Table 3: Maximum instantaneous error in reconstructed heat flux for TACOT example.

(a) Reconstructions using TC 1

| | | WD | $FT = 3\,\mathrm{s}$ | $FT = 5\,\mathrm{s}$ | $FT = 8\,\mathrm{s}$ |
|---|---|---|---|---|---|
| | FT algorithm | | 2.3% | 4.7% | 10.0% |
| SSD | a_1 $= -0.001$ | 0.3% | 0.1% | 0.2% | 0.2% |
| SSD | a_1 $= -0.01$ | 0.4% | 0.2% | 0.3% | 0.3% |
| SSD | a_1 $= -0.1$ | 0.9% | 0.7% | 0.4% | 0.6% |
| SSD | a_1 $= -1.0$ | 2.4% | 1.7% | 2.2% | 1.4% |

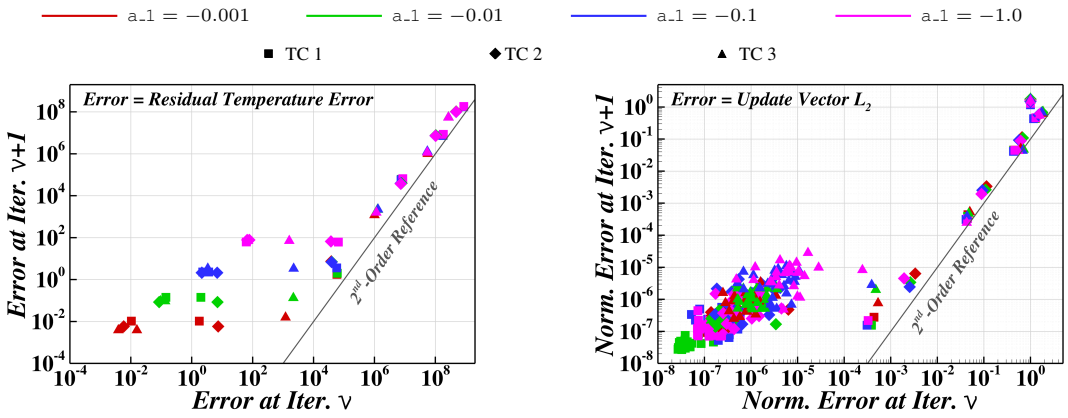
(b) Reconstructions using TC 2

| | | WD | $FT = 3\,\mathrm{s}$ | $FT = 5\,\mathrm{s}$ | $FT = 8\,\mathrm{s}$ |
|---|---|---|---|---|---|
| | FT algorithm | | 1.9% | 3.1% | 6.7% |
| SSD | a_1 $= -0.001$ | 0.5% | Failed | 0.3% | 0.3% |
| SSD | a_1 $= -0.01$ | 0.8% | 0.9% | 0.3% | 0.4% |
| SSD | a_1 $= -0.1$ | 1.8% | 0.9% | 1.0% | 0.5% |
| SSD | a_1 $= -1.0$ | 3.7% | 1.0% | 2.5% | 2.8% |

(c) Reconstructions using TC 3

| | | WD | $FT = 3\,\mathrm{s}$ | $FT = 5\,\mathrm{s}$ | $FT = 8\,\mathrm{s}$ |
|---|---|---|---|---|---|
| | FT algorithm | | Failed | 2.3% | 4.5% |
| SSD | a_1 $= -0.001$ | 0.6% | Failed | Failed | 1.1% |
| SSD | a_1 $= -0.01$ | 1.4% | Failed | Failed | 0.5% |
| SSD | a_1 $= -0.1$ | 2.7% | Failed | Failed | 0.9% |
| SSD | a_1 $= -1.0$ | 5.2% | Failed | 1.7% | 3.6% |

reduction in effort (from 30 to 6 non-linear iterations). The SSD algorithm shows a wider spread in the plateau levels, but iterations can be limited with convergence tolerances above $10^{-8}$. For this example, the SSD clearly requires less overall computational effort. However, it should be noted that using 48 processors, it required less wall-time to process the embarrassingly-parallel WD solutions than the $FT = 8\,\mathrm{s}$ SSD solutions because the sequential approach limits the parallel scalability of the SSD.




(a) Temperature Error

(b) Update Vector Magnitude

Figure 21: Non-linear convergence for WD reconstructions.

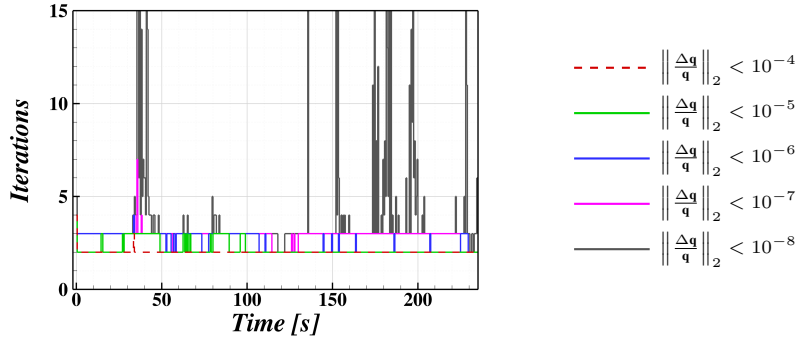
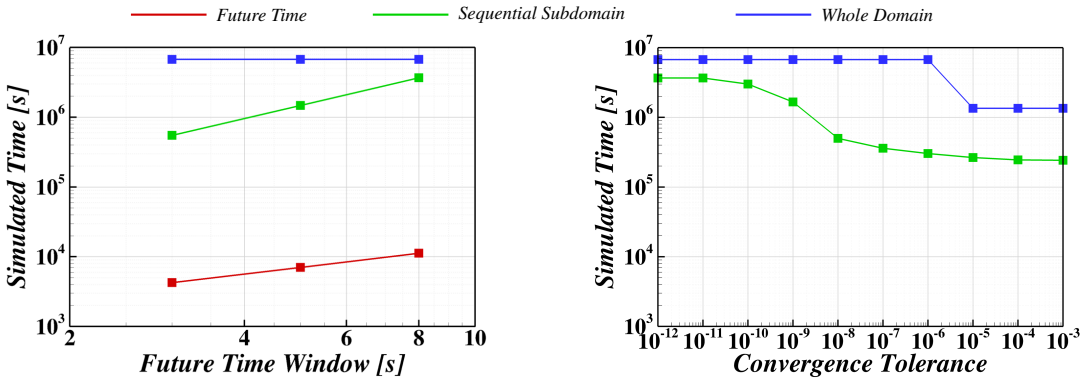



Figure 22: Non-linear iterations used in SSD reconstruction of TC 1 with $\texttt{a\_1} = -0.001$ as a function of the scaled solution convergence tolerance.




(a) Algorithm Comparison

(b) SSD, $FT = 8\,\mathrm{s}$

Figure 23: Amount of time simulated to evaluate sensitivity coefficients.

### IV.C. Jet Augmentation Case

It is not uncommon for transient events which significantly decrease the representative time-scales of the heating profile to occur in flight, and in some instances it would be desirable to be able to accurately characterize the heating behavior in these transient events. These rapid changes in relevant time scale can pose a challenge to IHCP algorithms, and particularly the regularization approaches employed. Transient events of this nature can increase the required reconstruction frequency and greatly increase the overall number of unknowns. While a WD reconstruction would likely be most able to cope with the changing time-scales, the number of total unknowns could make solution of the problem infeasible without special treatment of the numerical solution. In this situation, the SSD provides a means of obtaining a nearly WD-quality result without the computational expense or complexity of solving the whole problem in one step. In this example, the transient event in question will be the aerodynamic heating augmentation due to the firing of a nearby steering jet.

The material properties assumed in this example are representative of a reusable silica tile like those used on the heatshield of the Space Shuttle Orbiter ($25.4\,\mathrm{mm}$ of LI-900 coated with $0.5\,\mathrm{mm}$ of black reaction-cured glass (RCG), with material properties obtained from reference[31]). An aeroheating boundary condition is assumed on the RCG surface with the film coefficient and recovery enthalpy shown in Figure 24. Each of the spikes are caused by steering jet firings which augment the nominal film coefficient by a factor of 2 for $0.5\,\mathrm{s}$, with the middle $0.1\,\mathrm{s}$ being augmented by a total factor of 3.2 to simulate an overlapping firing of a second jet. The surface is assumed to reradiate to a $300\,\mathrm{K}$ farfield. The simulated thermocouple is placed at a depth of $0.5\,\mathrm{mm}$ at the interface between the RCG and LI-900. The TC is sampled at $50\,\mathrm{Hz}$, and a 'noisy' trace is constructed by truncating the resolution of the sampled TC data to a level representative of a 12-bit data acquisition system (Figure 25).

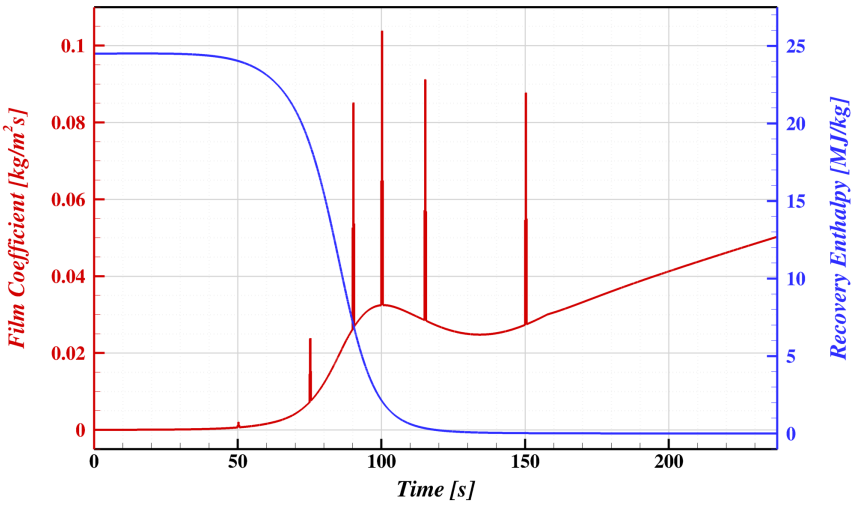


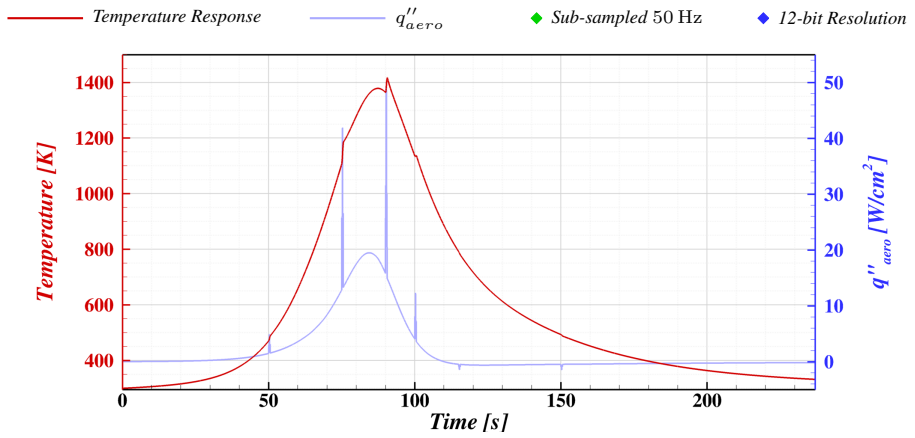

Figure 24: Film coefficient condition and resulting convective heating for RCS augmentation example problem.

The sensitivity coefficients that result for this case are shown in Figure 26. These coefficients are markedly different than those from earlier problems, and the effects of this will be seen in the results. The coefficients at early times show a very rapid rise (peaking approximately $0.3\,\mathrm{s}$ from the start of the pulse) followed by a slow decay. The magnitude of the peaks is observed to drop of rapidly in time, with the coefficients after the heat pulse very small by comparison. This reduction is due to the fact that the aerodynamic heat flux is given by

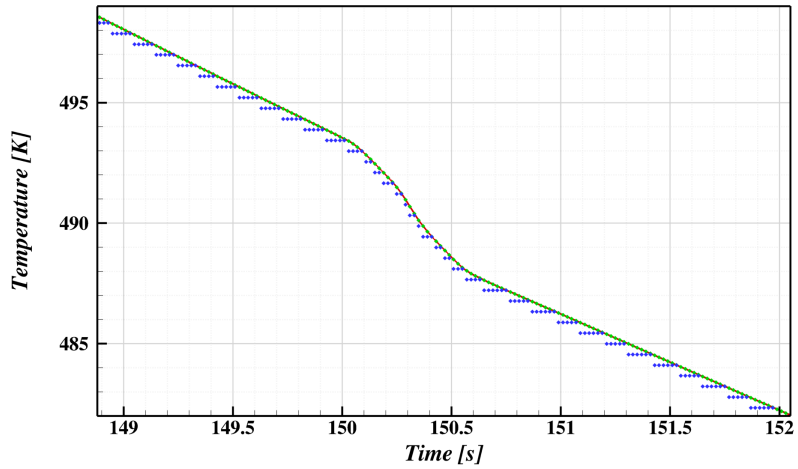
$$q''_{aero} = \rho_e u_e C_h (H_{rec} - H_w) \tag{31}$$

with $\rho_e u_e C_h$ being the film coefficient, $H_{rec}$ the recovery enthalpy, and $H_w$ the wall enthalpy. The TC actually responds to the heat flux, which is scaled by the recovery enthalpy. Since the recovery enthalpy drops during the entry profile, the TC becomes less sensitive to changes in the film coefficient. Note that the slight variation of the peaks relative to the co-plotted recovery enthalpy profile are caused by non-linearities in the thermal properties and reradiation which is a strong function of surface temperature.

Reconstructions are performed with the same integration time as the direct problem ($0.1\,\mathrm{s}$) and the same grid. The reconstructed film coefficient profile is assumed to be composed of uniformly distributed $0.1\,\mathrm{s}$ intervals, so there is an implicit factor of 5 future-time smoothing since the data rate is higher than the reconstruction rate. The FT and




(a) Overall Response




(b) 150 s Firing Detail

Figure 25: Target temperatures for jet augmentation case.

SSD algorithms will be employed, but the WD algorithm was not used since this setup requires the solution of 2,300 unknowns.

Figure 27 shows the reconstructions produced by the FT algorithm for several future time window lengths. Two features of the overall solution stand out: the 'glitch' seen at approximately 110 s and the growing instability of the noisy reconstructions at later times. The glitch at 110 s occurs at the time that the aerodynamic heat flux passes through zero (when the recovery enthalpy drops below the wall enthalpy at the onset of cooling, see the flux in Figure 25). At this point, the term $(H_{rec} - H_w)$ is very near zero, so the TCs become very insensitive to the film coefficient, making the IHCP particularly unstable. The growing noise later in times are caused by a combination of the small sensitivity coefficients late and the relatively large amount of noise introduced by quantizing the temperature trace to a 12-bit level. As the nominal signal is slowly decaying, the truncation operation yields correlated errors in the noisy trace, which the reconstruction algorithm tries to reproduce. The reconstruction with a future time window of 1 s is seen to significantly smooth out the transient profile of the jet firing, and even the shortest future time window of 0.3 s under-predicts the magnitude of the peak augmentation.

Reconstructions using the SSD with a future time window of 1 s are shown in Figure 28. The overall solutions show similar behavior as the FT reconstructions, however the detail views show much improved reconstructions of the transient heating pulse (only the 150 s firing is shown here as it is the most difficult, however all firings show very similar trends). This figure also shows the effect of varying the parameter $W_f$. Recall from the discussion of Equation 26 that a positive $W_f$ effectively decreases the regularization applied at the end of a local solution. Looking at the overall solution, it is apparent that having a positive $W_f$ slightly decreases the overall regularization applied (note the more rapidly growing instabilities late), however the transient pulse is more accurately represented with less

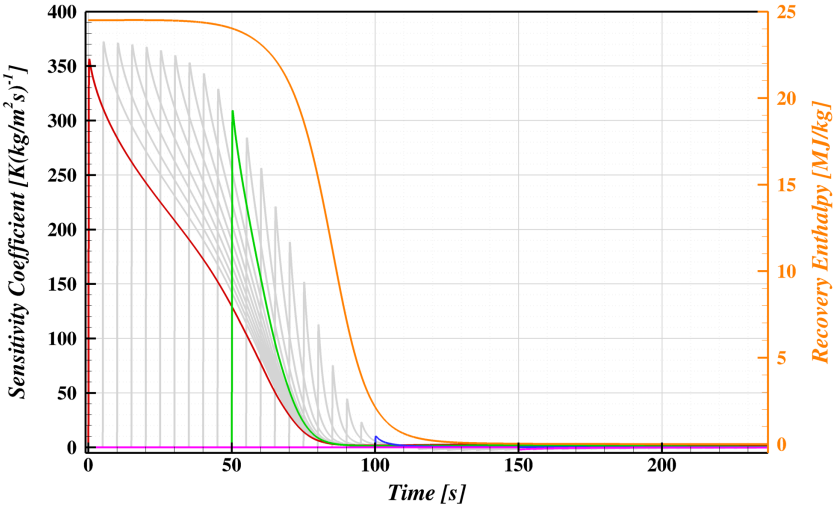
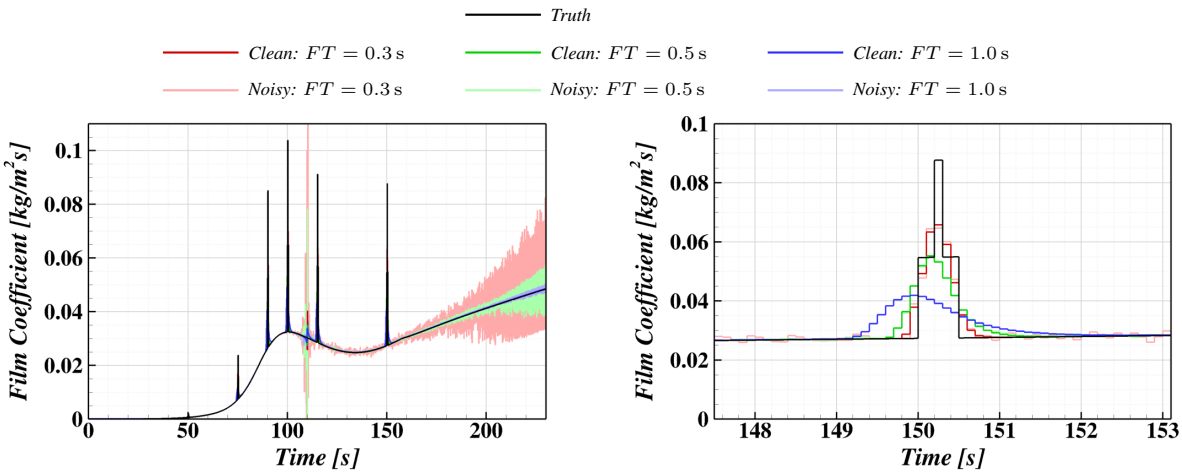



Figure 26: Sensitivity coefficients for jet augmentation example problem. Several curves are highlighted for clarity.




(a) Overall Solution

(b) 150 s Firing Detail

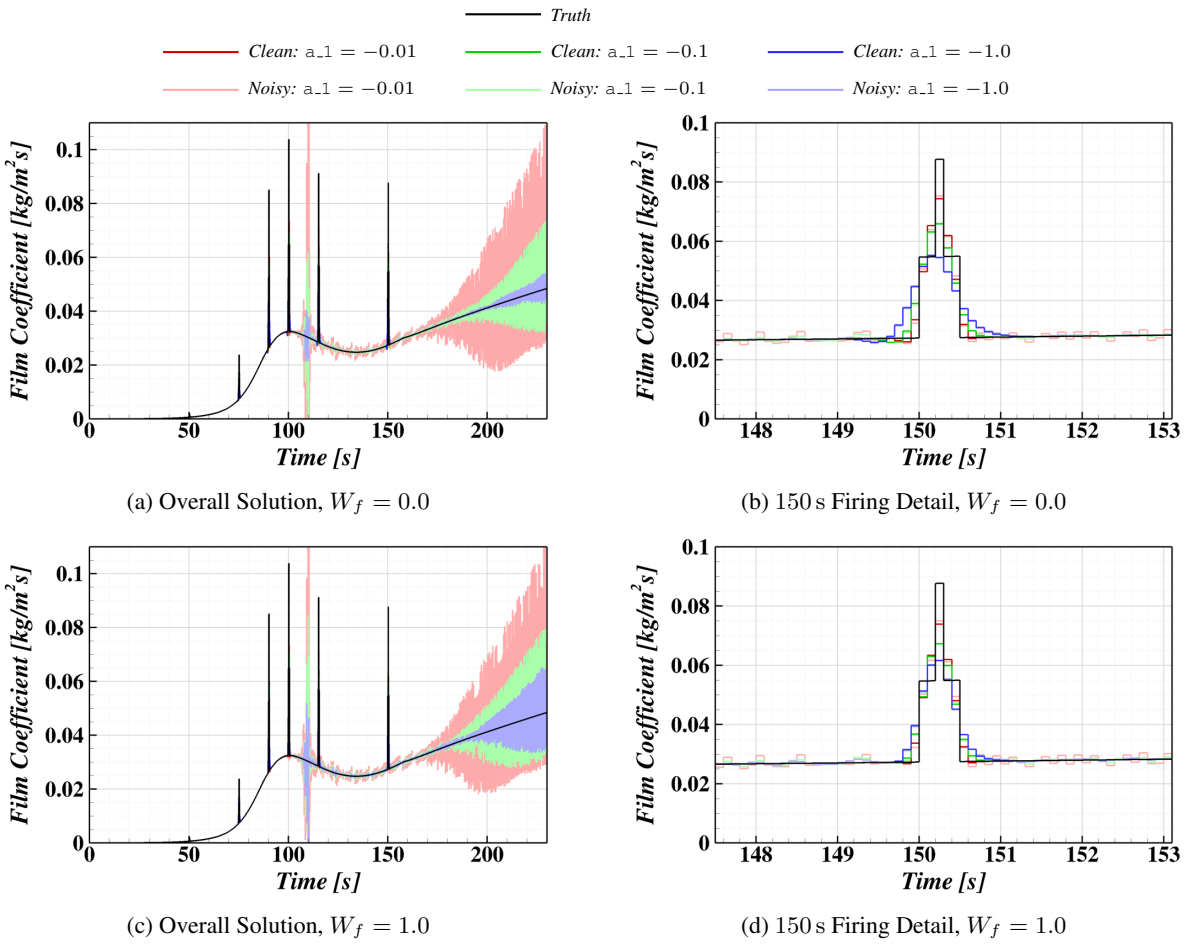
Figure 27: Reconstructions of jet augmentation test case using Future Time algorithm

tendency to undershoot before and after the firing. It would seem that this problem benefits from permitting more variation between later solution components in the local solution as the step change is approached.

The different reconstruction algorithms are compared in Figure 29. Figure 29(b) shows the transient response of each algorithm to with the minimum stable regularization applied. The SSD reconstruction with $W_f = 1.0$ is seen to capture the transient behavior the best, though the solutions outside the RCS firing are quite noisy. As the regularization is increased in Figures 29(c) and 29(d), the solutions behave as expected, with noisy responses being suppressed, but the under-prediction of the peak heating increases. Though not shown here, firings earlier in the entry profile, where the temperatures vary more rapidly (minimizing the quantization error) and the sensitivity coefficients are larger, the reconstructions with low regularization yield results comparable to 29(b) without the noise surrounding it.

### IV.D. Multi-Component Case

This example will demonstrate the multi-component reconstruction capability of *INHEAT*. The case is representative of a wind tunnel test with measurable lateral conduction. Generally, materials and time scales are controlled to minimize lateral condition effects in a wind tunnel test to permit the use of 1-D reconstruction techniques; however, it is not always feasible to design a test for which 1-D reconstructions are sufficient. Multi-component reconstructions




(a) Overall Solution, $W_f = 0.0$

(b) 150 s Firing Detail, $W_f = 0.0$

(c) Overall Solution, $W_f = 1.0$

(d) 150 s Firing Detail, $W_f = 1.0$

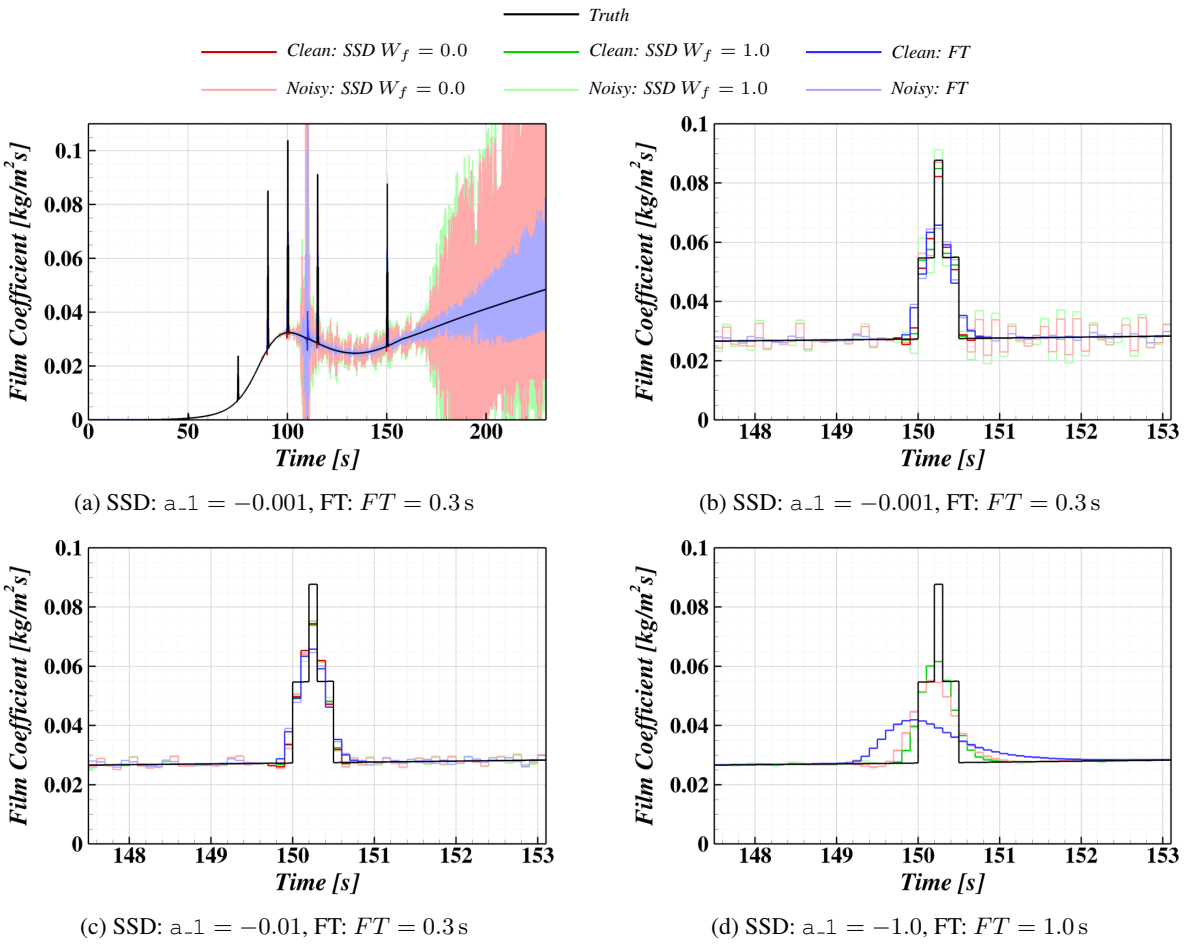
Figure 28: Reconstructions of jet augmentation test case using SSD algorithm with different values of $W_f$.

considering all of the simultaneous measurements will be compared to individual 1-D reconstructions at each sensor.

The 2-D domain considered is shown in Figure 30(a). The surface is broken up into 7 zones each with spatially uniform heating of different magnitudes. The surface left of the protuberance (simulating heating upstream of a protuberance) is represented by a low heating zone (red) and a very small peak region (green), which is crudely consistent with observed trends. The front face of the protuberance is broken into three sections, with heating increasing from the lower level (blue) through the middle region (gray) to the peak heating region high on the front face (orange). The top of the protuberance (magenta) has lower heating, but still higher than the upstream level. The aft portion of the protuberance, domain sides, and backface are all adiabatic. For all of the heated faces, the transient profiles of specified heat flux in Figure 31(a) (which are loosely representative of the flux in a convective environment) are applied for the direct problem using the grid in Figure 30(b) with an integration time step of $0.05\,\mathrm{s}$. Thermocouples are located $0.1\,\mathrm{in}$ below the surface of each heating zone. Material properties are $\rho = 7900\,\mathrm{kg/m^3}$, $k = 17\,\mathrm{W/m{\cdot}K}$, and $C = 400\,\mathrm{J/kg{\cdot}K}$. The temperature in the domain at $t = 10\,\mathrm{s}$ is shown in Figure 31(c), and the TC profiles are shown in Figure 31(b). Measurement noise will not be considered in this example.

Reconstruction of the temperatures in Figure 31(b) is performed using the SSD at $10\,\mathrm{Hz}$, with a future time window of $0.3\,\mathrm{s}$, and first order regularization scale factor of $\mathtt{a\_1} = -1.0$ (both 1-D and multi-component). There are 6 uncertain components in the multi-component reconstruction, one for the heat flux on each heated region in the domain. As the problem is linear, one-sided sensitivity coefficients and only one non-linear iteration are used. Since noise is not added, spatial regularization was not needed and was not applied.

Reconstruction results are shown in Figure 32. Notice first that the multi-component reconstruction results closely match the symbols (the true heating) very closely for each of the 6 components. The 1-D reconstructions of each individual TC are also shown in this plot. For some of the areas farther from geometric discontinuities (far upstream in red and on the protuberance top in magenta), the 1-D and multi-component reconstructions are reasonably close. For




(a) SSD: a_1 $= -0.001$, FT: $FT = 0.3\,\mathrm{s}$

(b) SSD: a_1 $= -0.001$, FT: $FT = 0.3\,\mathrm{s}$

(c) SSD: a_1 $= -0.01$, FT: $FT = 0.3\,\mathrm{s}$

(d) SSD: a_1 $= -1.0$, FT: $FT = 1.0\,\mathrm{s}$

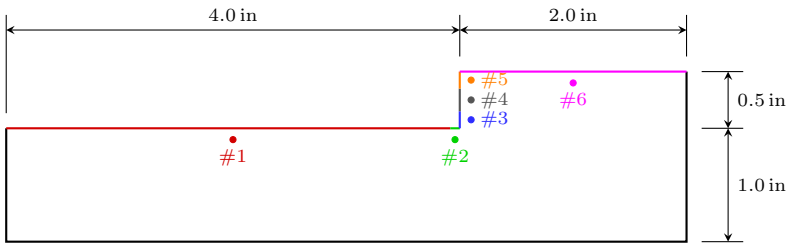
Figure 29: Comparison jet augmentation test case reconstructions with FT and SSD algorithms with varying regularization.

the other components, however, significant errors are observed in the 1-D reconstructions due to lateral conduction (typically high heating sensors are under-predicted and lower heating sensors are over-predicted). Note particularly the green trace, which has the highest heating, is significantly under-predicted when lateral conduction is not taken into account. The multi-component reconstruction algorithm, however, is able to account for the lateral conduction effects in the sensitivity coefficient evaluation and is therefore able to obtain an accurate reconstruction.

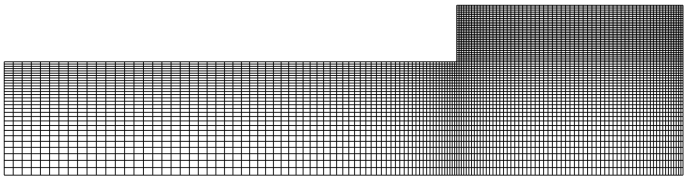
## V. Conclusions and Future Work

This paper has presented the formulation of the IHCP algorithm implemented as the *INHEAT* module of the *CHAR* thermal response code. A description of the key parameters in the algorithm have been provided. Furthermore, the algorithm has been demonstrated for several cases including a linear benchmark, a non-linear decomposing ablator, and a case representative of steering jet augmentation in a flight vehicle. While no one algorithm always performs best, the FT algorithm is seen to perform well when sensitivity coefficients develop rapidly compared to the rate change of the heating boundary condition, but suffers from over-smoothing otherwise. The WD algorithm can provide accurate reconstructions for any reasonable instrumentation depth and material combination, however constraints on the linear system limit the number of unknowns that may be estimated. The SSD algorithm presented here is a blend of the two algorithms, and provides many of the accuracy benefits of the WD algorithm while maintaining some of the computational efficiency of the sequential FT algorithm.

A number of improvements are planned for *INHEAT*. The first is adding the capability to assume piecewise linear functions in time as opposed to the piecewise constant assumed presently. Secondly, while multi-component reconstructions are possible and have been performed, the case-specific nature of these problems requires the user to input

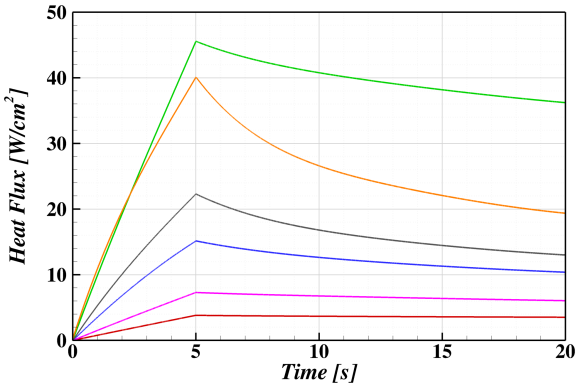



(a) Domain overview. Note that line colors are consistent with the remaining figures in this section.

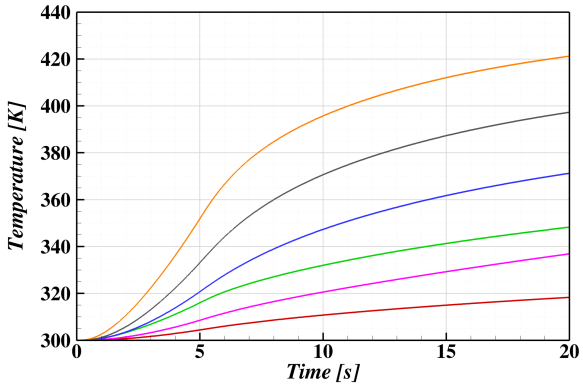


(b) Grid

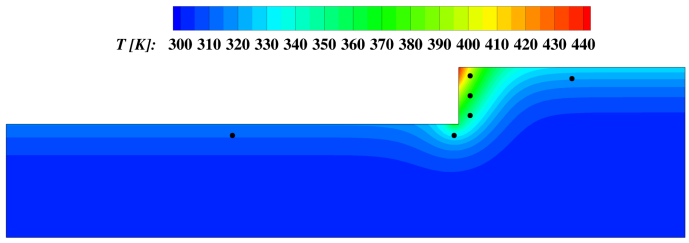
Figure 30: Domain and grid for multi-component example case.




(a) Applied heat flux




(b) Temperature response




(c) Volume solution at $t = 10\,\mathrm{s}$

Figure 31: Applied flux for direct problem and resulting temperature response. Note that line colors are defined by the colors in Figure 30(a).

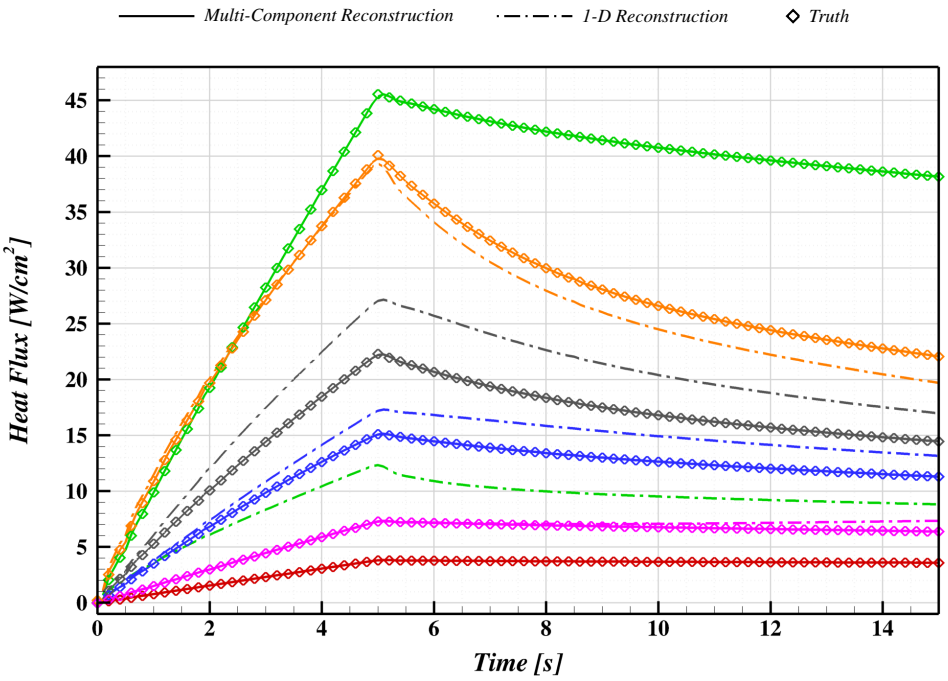



Figure 32: Reconstruction results of multi-component example case. Note that line colors are defined by the colors in Figure 30(a).

spatial regularization matrices in non-intuitive formats; it is desired to make this interface more user-friendly. Similarly, while the measurement covariance matrix can be provided (again, it an inconvenient form), more work is needed to fully understand the utility of this term and develop methods of inputting relevant information. Finally, the need for uncertainty estimates is well understood, but the present implementation has no built-in capability to assist in evaluating uncertainties. A number of methods are being investigated with the goal of easing the estimation of accuracy of inverse solutions for practical flight data analysis.

## VI. Acknowledgements

We would like to gratefully acknowledge that this work has relied heavily on input and material provided by Ben Blackwell. We would also like to thank Gregory Blaisdell and Anastasios Lyrintzis for providing reviews of the material that went into this paper.

## References

[1] Dorothy B. Lee and Windston D. Goodrich. The aerothermodynamic environment of the apollo command module during superorbital entry. Technical Report NASA TN D-6792, Manned Spacecraft Center, Houston, TX 77058, 1972.

[2] F. McNeill Cheatwood, Deepak Bose, Christopher D. Karlgaard, Christopher A. Kuhl, Jose A. Santos, and Michael J. Wright. Mars science laboratory (msl) entry, descent, and landing instrumentation (medli): Complete flight data set. Technical Report NASA/TM-2014-218533, NASA Langley Research Center, Langley Station, Hampton, VA, October 2014 2014.

[3] Deepak Bose, Todd White, Jose a Santos, Jay Feldman, Milad Mahzari, Michael Olson, and Bernie Laub. Initial Assessment of Mars Science Laboratory Heatshield Instrumentation and Flight Data. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, number January, 2013.

[4] Todd R. White, Ioana Cozmuta, Jose A. Santos, Bernie Laub, and Milad Mahzari. Proposed Analysis Process for Mars Science Laboratory Heat Shield Sensor Plug Flight Data. In *42nd AIAA Thermophysics Conference*, number June 2011, 2012.

[5] Todd R White, Milad Mahzari, Deepak Bose, and Jose A Santos. Post-flight Analysis of the Mars Science Lab-

oratory Entry Aerothermal Environment and Thermal Protection System Response. In *44th AIAA Thermophysics Conference*, pages 1–19, 2013.

[6] M. Necati Ozisik and Helcio R. B. Orlande. *Inverse Heat Transfer: Fundamentals and Applications*. Taylor & Francis, Washington, D.C., 2000.

[7] Keith A. Woodbury, James V. Beck, and Hamidreza Najafi. Filter solution of inverse heat conduction problem using measured temperature history as remote boundary condition. *International Journal of Heat and Mass Transfer*, 72:139–147, 2014.

[8] J. V. Beck, B. F. Blackwell, and C. R. St. Clair. *Inverse Heat Conduction Problems*. Wiley-Interscience, New York, NY, 1985.

[9] S. Deng and Y. Hwang. Applying neural networks to the solution of forward and inverse heat conduction problems. *International Journal of Heat and Mass Transfer*, 49(25-26):4732–4750, dec 2006.

[10] S. Deng and Y. Hwang. Solution of inverse heat conduction problems using Kalman filter-enhanced Bayesian back propagation neural network data fusion. *International Journal of Heat and Mass Transfer*, 50(11-12):2089–2100, jun 2007.

[11] Bryan S. Elkins, Majid Keyhani, and Jay I. Frankel. Surface Heat Flux Prediction Through Physics-Based Calibration, Part a: Theory. *Journal of Thermophysics and Heat Transfer*, 27(2):189–205, 2013.

[12] J I Frankel and M Keyhani. Inverse Ablation Analysis and the Calibration Integral Equation Method. *Journal of Thermophysics and Heat Transfer*.

[13] Milad Mahzari and Robert D Braun. Time-Dependent Mars Entry Aeroheating Estimation from Simulated In-Depth Heat Shield Temperature Measurements. *Journal of Thermophysics and Heat Transfer*, 27(3):435–446, 2013.

[14] Yvon Jarny. The adjoint method to compute the numerical solutions of inverse problems. In Keith A. Woodbury, editor, *Inverse Engineering Handbook*, chapter 3. CRC Press, 2003.

[15] Alfred Carassos. Determining surface temperatures from interior observations. *SIAM Journal of Applied Mathematics*, 42(3):558–574, 1982.

[16] Diego A. Murio. Mollification and space marching. In Keith A. Woodbury, editor, *Inverse Engineering Handbook*, chapter 4. CRC Press, 2003.

[17] Keith A. Woodbury. Sequential function specification method using future times for function estimation. In Keith A. Woodbury, editor, *Inverse Engineering Handbook*, chapter 2. CRC Press, 2003.

[18] Ben Blackwell. Inverse Methods for Estimating Heat Flux from Temperature Measurements: Seminar at Purdue Univeristy School of Aeronautics and Astronautics, 2008.

[19] Frank P. Incropera and David P. DeWitt. *Fundamentals of Heat and Mass Transfer*. John Wiley & Sons, fifth edition, 2002.

[20] J. Lachaud, A.J. Eckelen, D. Bianchi, and A. Martin. TACOT v3.0.

[21] Peter Craven and Grace Wahba. Smoothing noisy data with spline functions - Estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerische Mathematik*, 31:377–403, 1979.

[22] C.H. Huang and M.N. Ozisik. Optimal regularization method to determine the strength of a plane surface heat source. *International Journal of Heat and Fluid Flow*, 12(2):173–178, 1991.

[23] Adriaan van den Bos. *Parameter Estimation for Scientists and Engineers*. Wiley, 2007.

[24] Patricia K Lamm. Variable-smoothing local regularization methods for first-kind integral equations. *Inverse Problems*, 19(1):195–216, 2003.

[25] Patricia K Lamm and Zhewei Dai. On local regularization methods for linear Volterra equations and nonlinear equations of Hammerstein type. *Inverse Problems*, 21(5):1773–1790, 2005.

[26] Benjamin S. Kirk, John W. Peterson, Roy H. Stogner, and Graham F. Carey. `libMesh`: A C++ Library for Parallel Adaptive Mesh Refinement/Coarsening Simulations. *Engineering with Computers*, 22(3):237–254, 2006.

[27] Adam J Amar, Nathan D Calvert, and Benjamin S Kirk. Development and Verification of the Charring Ablating Thermal Protection Implicit System Solver. In *AIAA-2011-144, 49th AIAA Aerospace Sciences Meeting*, number January, 2011.

[28] Adam J. Amar, A. Brandon Oliver, Benjamin S. Kirk, Giovanni Salazar, and Justin Droba. Overview of the CHarring ablator response (CHAR) code. 46th AIAA Thermophysics Conference, Submitted for publication, 2016.

[29] Adam J. Amar. Modeling of One-Dimensional Ablation with Porous Flow Using F inite Control Volume Procedure. Master's thesis, North Carolina State University, 2006.

[30] Carl B. Moyer and Roald A. Rindal. An analysis of the coupled chemically reacting boundary layer and charring ablator, part ii: Finite difference solution for the in-depth response of charring materials considering surface

chemical and energy balances. Technical Report 66-7 Part II, Aerotherm, March 1967.
[31] S. D. Williams and Donald M. Curry. Thermal protection materials: Thermophysical property data. Technical Report NASA-RP-1289, 1992.