

Optimization of an Inverse Convection Solution Strategy

Joseph R VanderVeer, Yogesh Jaluria

*Department of Mechanical and Aerospace Engineering: Rutgers University, 98 Brett Rd,
Piscataway NJ, 08854*

Abstract

An optimization procedure was applied to the search process of a previously developed predictor - corrector method for solving inverse convection heat transfer problems. A genetic algorithm was utilized to minimize the area of acceptable error for the inverse methodology over the entire relevant domain. The optimization resulted in a new search shape for the inverse methodology. This substantially reduced the required sample size. An important additional advantage of the optimization was the elimination of alternative local minimums. The new and old sampling pattern were used to solve an inverse problem of a thermal plume in a crosswind. The results were that the maximum error from the original search shape was 10% while the new search shape reduced the error to under 1% for the entire usable domain. Also of importance is the significantly increased spatial range of the methodology down stream. This approach could be extended to many applied areas such as environmental flows, room fires, and thermal management systems.

Keywords: Inverse Problems, Computational Heat Transfer, Convection

1. Introduction

Thermal-fluid systems often have limited physical access to sensors and/or limited information. These situations are often inverse heat transfer problems

*Corresponding Author

and thus inverse problems are of great importance. These problems differ from normal/forward problems, by having limited or no a priori knowledge of the boundary conditions. These problems must rely on limited knowledge of the domain to solve for the unknown boundary conditions.

For example, the optical fiber drawing furnace wall temperature cannot be directly measured due to the shape of the furnace, inaccessibility, and high temperatures. For this case, an instrumented rod is used to determine the temperature at the centerline from which the wall temperature of the furnace is to be obtained. This results in an inverse heat transfer problem, solved through a regularization technique developed by [?].

An example of knowing limited information of the boundary condition and determining the rest of the boundary conditions would be determining the location of a fire in a room from outside the room. This would be accomplished using thermal sensors located on the door along with knowledge of the shape of the room to solve the inverse convection problem.

Newer methods for solving inverse heat transfer problems are constantly being developed and improved upon. Inverse conduction and radiation problems have been the focus of much of the previous research, as displayed by the several books agglomerating the subject (e.g. [? ?]). Relatively recently, the inverse convection problem has been gaining in popularity because of its considerable practical importance. For example [?] solved inverse free convection problems with time varying heat fluxes with a single sensor. They determined that as the Rayleigh number increases, the sensor needs to be placed progressively closer to the active boundary layer. Another example is the paper by [?], in which they use inverse convection methods to determine thermal profiles in a slot vented enclosure. They use an iterative approach requiring 20 to 30 iterations to achieve less than 1% error.

One last example of the inverse convection problem occurs with unknown jet conditions for a jet in a cross-wind. [?], in conjunction with [?], used a combination experimental-numerical model with a response surface to reasonably predict the jet temperature and velocity within the domain. They were able to predict the source temperature within 9%, but the second stage of the mechanism significantly over predicted the source temperature. The jet velocity was predicted within the experimental error.

The present study is an attempt to increase the overall accuracy and effectiveness of a methodology used to solve inverse convection problems as, originally proposed by [?] and fully implemented in a subsequent paper [?]. This methodology uses an effective spatial pattern to search the domain for

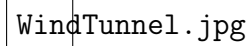


Figure 1: Schematic of the wind tunnel and the computational domain [?]]

a statistical match. The spatial pattern is identified by the relative location of the sampling points and is labeled the search shape. The minimization of error between the selected data points results in the solution to the inverse heat transfer problem. This method was demonstrated to be effective for numerous cases. However, the search shape was based upon intuition only and had no bearing upon the underlying problem. The goal of this work is to extend the method described earlier by optimizing the shape of the search pattern while extending its capabilities and accuracy.

2. Inverse Solution Methodology

For completeness, the methodology described by [?] will be briefly described here. The method was developed to solve an inverse convection problem consisting of a plume in a crosswind. The goal was to determine the plume strength and location within the domain. The crosswind was created via a small wind tunnel with test section dimensions of $54.5 \times 305 \times 254$ mm. A diagram of the experimental and computational domain is shown in [?]. All of the dimensions are in millimeters.

The variations in density, thermal buoyancy, and thermal radiation may be neglected in the energy equation over the range considered. Then, the relation between the local temperature and the source (plume) temperature is a linear function. [?] is the linear function with T_S referring to the source temperature and $T(\mathbf{r})$ referring to the local temperature. Also, \mathbf{r} is a simple x, y vector as shown in [?] and Δ is related to \mathbf{r} by the relation in [?]. The quantities m and b are defined in the subsequent [????], where the subscripts

A and B denote different simulation conditions [?].

$$T_S = m(\mathbf{r}) T(\mathbf{r}) + b(\mathbf{r}) \quad (1a)$$

$$\mathbf{r} = r(x, y) \quad (1b)$$

$$\mathbf{r}_i = \mathbf{r}_0 + \mathbf{\Delta}_i \quad (1c)$$

$$m(\mathbf{r}) = \frac{T_{SA} - T_{SB}}{T_A(\mathbf{r}) - T_B(\mathbf{r})} \quad (1d)$$

$$b(\mathbf{r}) = T_{SA} - m(\mathbf{r}) T_A(\mathbf{r}) \quad (1e)$$

The simulations were performed using Ansys Fluent version 13 [?]. The Navier-Stokes equations were solved using a three-dimensional, steady state, realizable, $k - \epsilon$ model with enhanced wall effects [?]. The fluid was modeled as an ideal gas at constant atmospheric pressure and specific heat at constant pressure was modeled as a constant at $C_P = 1006.43 \text{ J/(kg} - K)$. Dynamic viscosity and thermal conductivity were both modeled using the Chapman-Encog equations [?].

A turbulence model is required, while the Reynolds number is only of order 2000 the Richardson number is near 5 and thus the buoyancy introduces turbulence into the flow. The governing equations are ??????????????, they are in order: velocity decomposition, mass, momentum, energy, turbulent kinetic energy, turbulence dissipation rate, and Reynolds stress.

$$u_i = \overline{u_i} + u_i' \quad (2)$$

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i} (\rho u_i) = 0 \quad (3)$$

$$\begin{aligned} \frac{\partial}{\partial t} (\rho u_i) + \frac{\partial}{\partial x_j} (\rho u_i u_j) = \\ \frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\mu \left(2S_{ij} - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \right) - \overline{\rho u_i' u_j'} \right] \end{aligned} \quad (4)$$

$$\begin{aligned} \frac{\partial}{\partial t} (\rho E) + \frac{\partial}{\partial x_i} [u_i (\rho E + P)] = \\ \frac{\partial}{\partial x_i} \left[\left(\lambda + \frac{C_p \mu_t}{Pr_t} \right) \frac{\partial T}{\partial x_i} \right] \end{aligned} \quad (5)$$

$$\begin{aligned}
\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_j}(\rho k u_j) = \\
\frac{\partial}{\partial x_j} \left[\left(\mu + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] + \frac{\partial u_j}{\partial x_i} \left(-\rho \overline{u'_i u'_j} \right) \\
- g_i \frac{\mu_t}{\rho P_{rt}} \frac{\partial \rho}{\partial x_i} + \rho \epsilon
\end{aligned} \tag{6}$$

$$\begin{aligned}
\frac{\partial}{\partial t}(\rho \epsilon) + \frac{\partial}{\partial x_j}(\rho \epsilon u_j) = \\
\frac{\partial}{\partial x_j} \left[\left(\mu + \frac{\mu_t}{\sigma_\epsilon} \right) \frac{\partial \epsilon}{\partial x_j} \right] + \rho C_1 S \epsilon - \rho C_2 \frac{\epsilon^2}{k + \sqrt{\nu \epsilon}} \\
- C_{1\epsilon} \frac{\epsilon}{k} C_{3\epsilon} g_i \frac{\mu_t}{\rho P_{rt}} \frac{\partial \rho}{\partial x_i}
\end{aligned} \tag{7}$$

$$- \rho \overline{u'_i u'_j} = 2\mu_t S_{ij} - \frac{2}{3} \delta_{ij} \left(\rho k + \mu_t \frac{\partial u_k}{\partial x_k} \right) \tag{8}$$

The constants for the turbulence model are [? ?] :

$$C_{1\epsilon} = 1.44, C_2 = 1.9, \sigma_k = 1.0, \sigma_\epsilon = 1.2, P_{rt} = 0.85 \tag{9}$$

$$C_1 = \max \left[0.43, \frac{Sk/\epsilon}{Sk/\epsilon + 5} \right], S = \sqrt{2S_{ij}S_{ji}}, C_{3\epsilon} = \tanh \left(\frac{u_g}{u_p} \right) \tag{10}$$

$$\mu_t = \frac{\rho C_\mu k^2}{\epsilon} \tag{11a}$$

$$C_\mu = \frac{1}{A_0 + \frac{A_1 k U^*}{\epsilon}} \tag{11b}$$

$$U^* \equiv \sqrt{S_{ij}S_{ji} + \Omega_{ij}\Omega_{ji}} \tag{11c}$$

$$A_0 = 4.04 \tag{11d}$$

$$A_1 = \sqrt{6} \cos \left[\frac{1}{3} \cos^{-1} \left(\sqrt{6} \frac{S_{ij}S_{jk}S_{ki}}{(S_{ij}S_{ji})^{\frac{3}{2}}} \right) \right] \tag{11e}$$

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \tag{11f}$$

$$\Omega_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right) \tag{11g}$$

Where the u_g and u_p are the velocity component parallel and perpendicular to gravity respectively.

The simulations were validated against experimental results and two such comparisons are shown in ???. The locations of the plots are normalized and the normalization method is shown in ?? where L is the width of the plume.

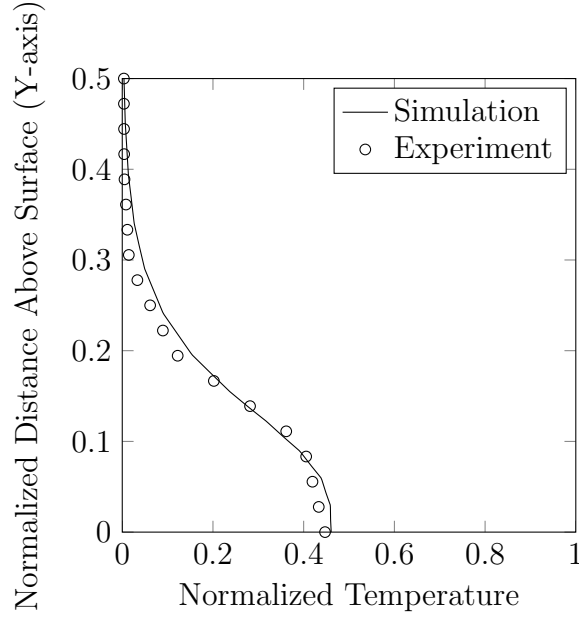


Figure 2: Validation of the simulation: Experiment versus simulation at $X = 1.6$ [?]]

$$\phi = \frac{T - T_{\infty}}{T_S - T_{\infty}} \quad (12a)$$

$$X = \frac{x}{L} \quad (12b)$$

$$Y = \frac{y}{L} \quad (12c)$$

The inverse solution begins with the acquisition of n samples from within the domain, where n may be as few as 3. The relative location between the samples must be known, that is Δ_i must be known for all samples. The simulation data is used to construct inverse interpolation functions m and

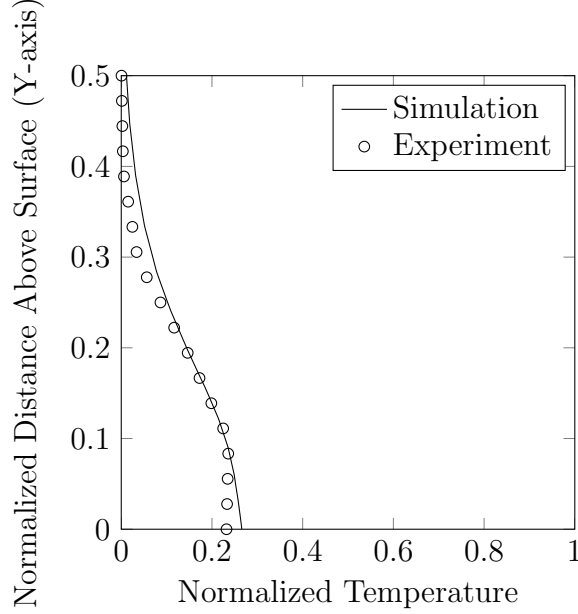


Figure 3: Validation of the simulation: Experiment versus simulation at $X = 2.25$ [?]]

b. Only two simulations are required due to the linear nature of ???. F is the objective function for the predictor stage of the methodology. The minimization of F is performed utilizing only a subset a samples of the original n samples. The value of a is chosen to be $\frac{1}{2}n$. While this may not be optimum for larger values of n , for small values of n it ensures both predictor and corrector steps have data to work with. The vector \mathbf{r}_{SP}^* is the solution to the minimum of F . If such a solution can be found, then ??? may be used to calculate the strength of the source plume [?]] .

$$F(\mathbf{r}) = \sum_{i=1}^a [m(\mathbf{r} + \Delta_i) T(\mathbf{r}_i) + b(\mathbf{r} + \Delta_i) - m(\mathbf{r}) T(\mathbf{r}_0) - b(\mathbf{r})]^2 \quad (13)$$

$$T_{SP}^* = \frac{1}{a} \left\{ \sum_{i=0}^a [m(\mathbf{r}_{SP}^* + \Delta_i) T(\mathbf{r}_i) + b(\mathbf{r}_{SP}^* + \Delta_i)] \right\} \quad (14)$$

The corrector phase of the method begins with correcting ?? for the now predicted source strength, which creates ??. Minimizing ?? by utilizing the rest of sampling points results in the source location \mathbf{r}_{SP} . A sample point \mathbf{r}_0 must be reused for ?? to still be valid, otherwise the \mathbf{r}_0 solved for will change

its definition during the solution method. The corrected source strength can then be calculated using ???. A flowchart of the methodology is found in ?? [?]. A typical inverse calculation requires approximately 5 minutes on a single thread of a 3.2Ghz processor. If the minimization method selected may be multi-threaded, the time will reduce dramatically as the bulk of the process time is the minimization required.

$$F_{mod}(\mathbf{r}) = \sum_{i=0}^{n-a} [m(\mathbf{r} + \Delta_i) T(\mathbf{r}_i) + b(\mathbf{r} + \Delta_i) - T_{SP}^*]^2 \quad (15)$$

$$T_{SP} = \frac{1}{n-a} \left\{ \sum_{i=0}^{n-a} [m(\mathbf{r}_{SP} + \Delta_i) T(\mathbf{r}_i) + b(\mathbf{r}_{SP} + \Delta_i)] \right\} \quad (16)$$

3. Methodology

The optimization of the predictor - corrector method begins with identifying the function needing to be optimized. The function desired needs to minimize the error associated with the prediction of the source temperature, since the predictor step defines the effectiveness of the corrector step. Thus, we start with ?? and determine the error of this function. ?? is rewritten here with a few modifications. The first aspect to identify is the change from \mathbf{r}_{SP}^* to \mathbf{r}_0 , indicating that \mathbf{r}_0 is fixed prior to the optimization. The Δ is now included in the local temperature function to keep the relative distance during the optimization process. Lastly is the addition of δ , that is used to calculate the error for the entire domain by varying the location of the sampled points.

$$T_{SO} = \frac{1}{n} \left\{ \sum_{i=0}^n [m(\mathbf{r}_0 + \Delta_i + \delta) T(\mathbf{r}_0 + \Delta_i) + b(\mathbf{r}_0 + \Delta_i + \delta)] \right\} \quad (17)$$

The error of ?? is written via ?? where ε is the percent error.

$$\left| \frac{T_{SO} - T_S}{T_S - T_\infty} \right| \times 100 = \varepsilon \quad (18)$$

The effectiveness of any one sampled data point is determined by how much it reduces the error at the test location and how much it increases the error

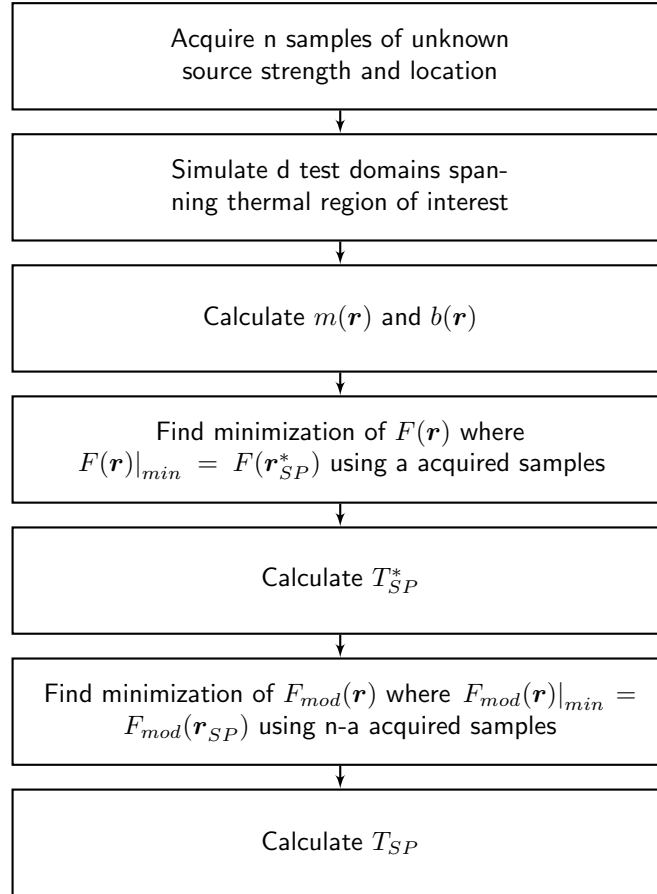


Figure 4: Flow chart of the predictor - corrector methodology [?]

everywhere else. This is accomplished with the area defined by ?? at a specified error ε .

The method used minimizes the area enclosed by ?? at a defined ε using one variable sample point. That is to say that the initial point is selected as \mathbf{r}_0 , which through the optimization yields, \mathbf{r}_1 . Repeating this process, using \mathbf{r}_0 and \mathbf{r}_1 , the optimization process can find \mathbf{r}_2 . If more accurate results are desired, this process may be continued indefinitely. For this particular inverse problem of the plume in a crosswind, three total sampling points were enough to achieve the desired accuracy, i.e. better than the experimental temperature error of 2%. This process is repeated for many iterations varying U_∞ , T_S , and \mathbf{r}_0 to cover the domain.

While any global optimization method would likely work the genetic algorithm was chosen due to its robustness and ease of use. A canned genetic algorithm within the Mathworks Matlab[?] optimization toolkit was utilized. The genetic algorithm is a bio-mimetic optimization strategy specifically designed to be robust and solve for global minimums [?]. The genetic algorithm roughly mimics the evolution of a species with the design parameters called chromosomes. The process starts by generating a large pool of chromosomes, each is checked against function needing to be optimized and ranked. A selection process then chooses pairs of parents to generate the next generation. Information from each of the chromosomes is used to make children chromosomes in what is known as crossover. These new chromosomes have a small chance at mutation, which randomly changes a design parameter. This reduces the chances of stagnation of the chromosome pool. This process repeats until a stopping criteria is met, usually the total number of iterations or a count of iterations where the best performing chromosome does not change. In the case of the Matlab version those default numbers are 100 and 50 respectively.

The only parameters changed from the Matlab defaults are listed in ?. Due to the repetitive nature and large total iteration count of the methodology, the population size of the genetic algorithm was kept small to reduce computation time. While this could reduce the effectiveness of the genetic algorithm the large sample size reduces the effects. Several test cases were performed with much larger populations and generation counts and found that the resultant solution was very similar to the reduced population version. The computation time was approximately 2 weeks and run on a twin processor, twelve core computer at 3.2GHz with 32GB of RAM. A flow chart of the methodology is shown in ?.

Parameter	Value
PopulationSize	60
Generations	200

Table 1: Genetic algorithm parameters

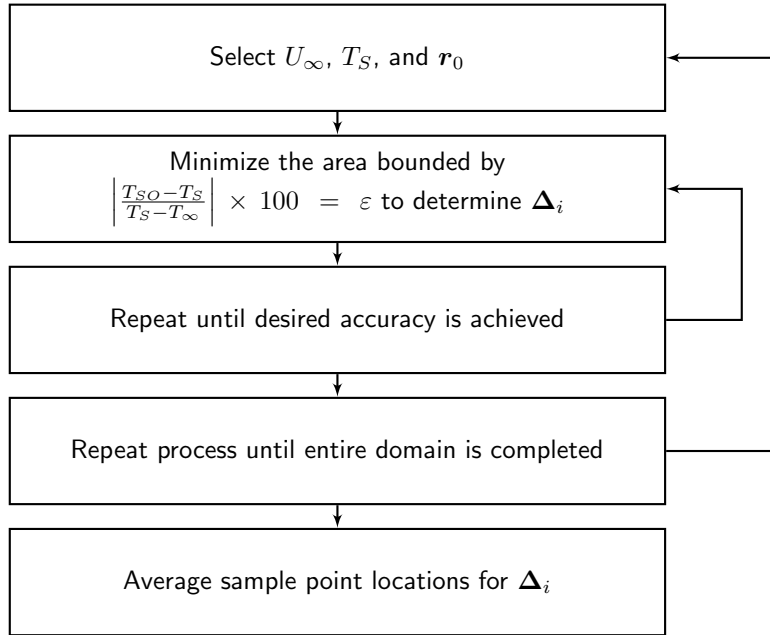


Figure 5: Flow chart of the presented methodology

4. Results and discussions

The methodology requires a large sampling of the domain of interest to function properly. The selected parameters are varied as shown in ???. These parameters were selected to thoroughly cover the experimental domain, which has physical limitations. The material properties of the wind tunnel limit T_S to 450 K. Which in turn limits U_∞ to approximately 1.0, m/s due to the plume being pushed too close to the lower boundary and becoming difficult to measure accurately. The parameter r_{axial} is the coordinate in the direction of the free stream, and r_{perp} is the coordinate perpendicular to the free stream. The ranges for both r_{axial} and r_{perp} were selected to completely cover the effective range of the plume. The end result of using these selected parameters is 1050 total samples in the domain.

Parameter	Minimum	Maximum	Increments
U_∞ (m/s)	0.0	1.0	0.2
T_S (K)	350	450	25
r_{axial} (mm)	0	150	25
r_{perp} (mm)	0	10	2.5

Table 2: Variable domain parameters used for optimization

The results of this optimization is a “perfect” search shape for each of the aforementioned domain parameters. However, the point of the inverse problem is such that the domain parameter is not known and determining the domain parameters are in fact the desired result of the inverse methodology. Therefore to make the optimization worthwhile all of the \mathbf{r}_1 ’s are averaged together and similarly for \mathbf{r}_2 .

The result from averaging the 1050 domain optimization is shown in ???. While the methodology is capable of providing more sample points, the accuracy provided by the three sample points was more than enough to accurately solve the described inverse problem. The predictor step uses the first and second samples, while the corrector step uses the first and third samples. That is to say \mathbf{r}_0 is reused as previously described.

For comparisons sake, ??? contains the original search shape used in the inverse solution. This search shape was selected based upon intuition only. The original samples one through three were used for the predictor stage and one, four, and five were used in the corrector stage.

Sample Point	Δ (mm)
0	(0.0, 0.0)
1	(1.7, 3.5)
2	(2.8, 0.6)

Table 3: Search shape results of averaging over the domain

Sample Point	Δ (mm)
1	(0.0, 0.0)
2	(1.0, 0.0)
3	(2.0, 0.0)
4	(0.0, 1.0)
5	(0.0, 2.0)

Table 4: Original search shape for this inverse solution

The error plots of ????????? demonstrate the effectiveness of the new sampling points. These plots are explained as the predicted source temperature error if an incorrect location was predicted. The correct result location is at $\mathbf{r}_0 = (100.0, 2.0) \text{ mm}$ and is indicated by an ‘x’. The shaded regions indicate how much error is associated with the prediction stage as it searches for a match within the domain. That is to say it is a plot of ε from ???. The x and y distances are normalized to the heater width of 25.4 mm . The size and scale of each plot is kept the same for comparison’s sake. The largest contour region is empty space and indicates an error of 20% or more.

?? is an error plot for 1 sample point. This is the same for both the original sampled data and the optimized sampled data. The darkest shaded regions indicate a 5% or less error in source temperature for those locations, indicating that any of those regions are viable answers. The biggest issue is that there are many local minimums, which will never converge to the correct solution. ???? and ???? show the differences between the original sampled data pattern and the optimized sampled data pattern. In both cases the error from the optimized sampled data is smaller and more confined than the original sampled data.

The area enclosed by the 5% error line is presented in ??. This quickly

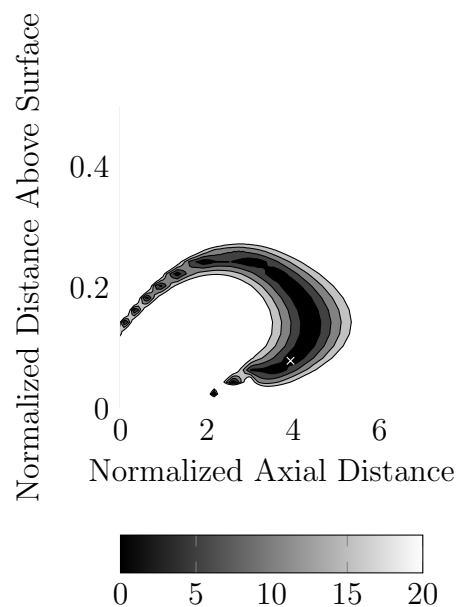


Figure 6: Contours of source prediction error, utilizing 1 point of information indicated by the 'x'

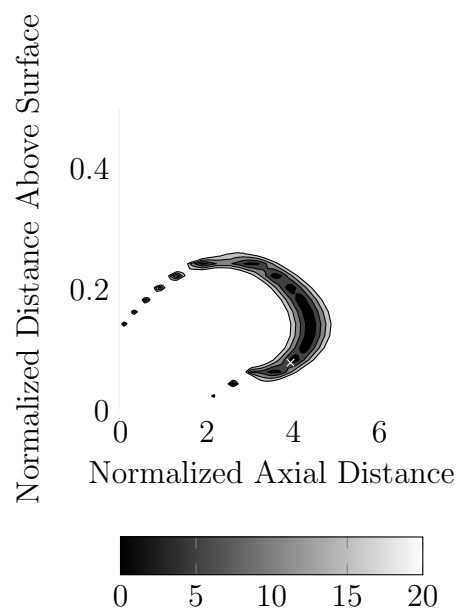


Figure 7: Contours of source prediction error, utilizing 2 points of sampled data from the original search shape

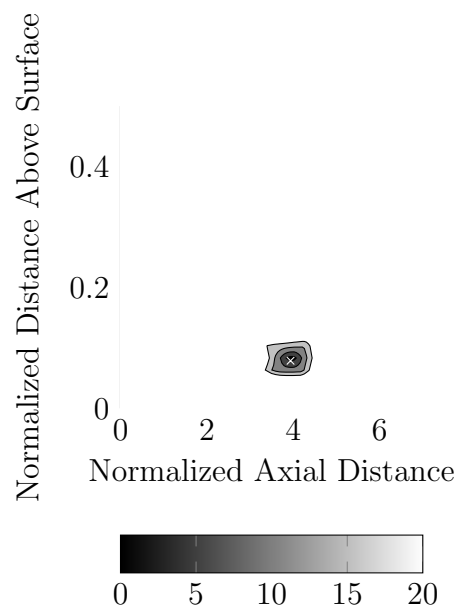


Figure 8: Contours of source prediction error, utilizing 2 points of sampled data from the optimized search shape

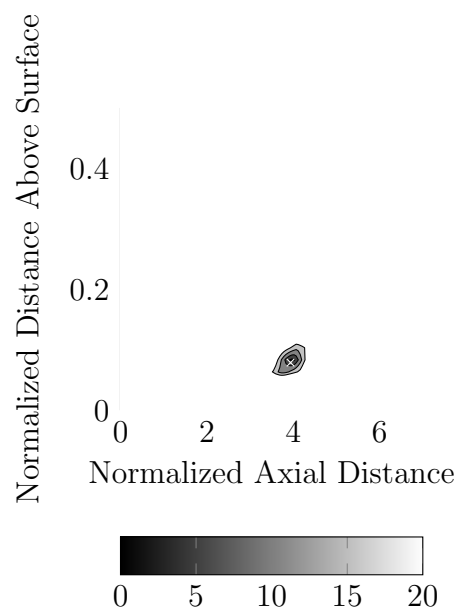


Figure 9: Contours of source prediction error, utilizing 3 points of sampled data from the optimized search shape

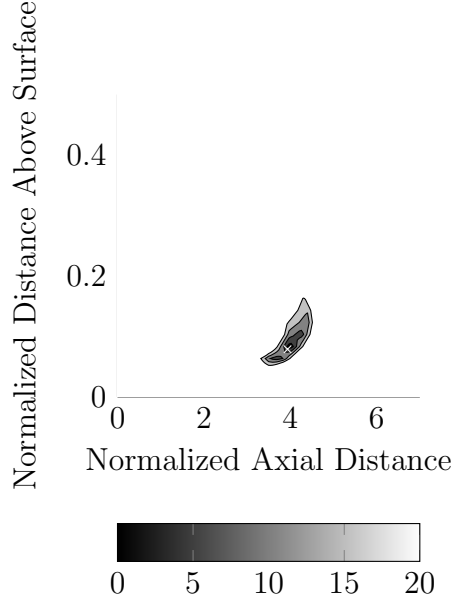


Figure 10: Contours of source prediction error, utilizing 3 points of sampled data from the original search shape

shows the effectiveness of the optimized search shape. The optimized search shape is more accurate with two sample points than the original search shape was with three sample points. Only at five sample points does the original sample set surpass the effectiveness of the optimized version. This, however, does not tell the whole story, as the lack of alternative local minimums in the optimized data are of much more importance. This allows the use of quicker and more efficient optimization techniques in the inverse methodology, while guaranteeing the solution does not converge to an alternative local minimum. This shows how effective the optimization for the search shape is, while highlighting the affects of not doing so. Unfortunately, this means for every experimental configuration a optimized search shape needs to be developed for maximum effectiveness of the inverse algorithm.

The previously presented original and optimized sampled data set comparison had the following test conditions, ??.

Lastly let us make a comparison of the results from the inverse methodology shown in ??. The parameters are labeled on the left. The original search shape solutions are in the middle under “original”. The optimized search shape solutions are on the right under “optimized”. Under the parameters

Sample Point	Original Search Shape	Optimized Search Shape
1	2375	2375
2	1142	60
3	71	30
4	39	-
5	13	-

Table 5: Comparison of enclosed area for 5% error (0.01 mm^2)

Parameter	Value
$T_\infty (K)$	293
$P_\infty (kPa)$	101.32
$T_S (K)$	425
$U_\infty (m/s)$	0.6

Table 6: Test parameters for the sampled set comparison

heading U_∞ is the free stream velocity, while T_S is the source temperature (not known a priori) for that case. Also under the parameters heading is \mathbf{r}_0 , this is the location of the datum point (not known a priori). Under the heading original T_S is the % error the original search shape had for predicting the source temperature. The next column \mathbf{r}_0 is the % error for predicting the location of the plume source used the original search shape. The last two columns are % error associated for predicting the source temperature and source location using the optimized search shape. The data presented here is a small subset of the complete data, but contains a good representation of the complete data. Percent error is calculated using the following ??.

$$error_{temp}(\%) = \frac{|T_{SP} - T_S|}{T_S - T_\infty} \times 100 \quad (19a)$$

$$error_{\mathbf{r}}(\%) = \frac{|\mathbf{r}_{SP} - \mathbf{r}_S|}{\mathbf{r}_S} \times 100 \quad (19b)$$

$$(19c)$$

Parameters			Original		Optimized	
$U_\infty (m/s)$	$T_S (K)$	$\mathbf{r}_0 (mm)$	T_S	\mathbf{r}_0	T_S	\mathbf{r}_0
0.4	375	(100.0, 1.0)	0.12	(0.2, 0.0)	0.12	(0.1, 0.0)
0.6	375	(100.0, 1.0)	0.36	(0.3, 10.0)	0.12	(0.1, 0.0)
0.6	400	(100.0, 1.0)	0.84	(0.3, 0.0)	0.09	(0.1, 0.0)
0.6	425	(100.0, 1.0)	0.61	(0.1, 10.0)	0.07	(0.1, 0.0)
0.6	425	(140.0, 8.0)	0.68	(3.4, 0.0)	0.53	(0.0, 0.0)
0.6	425	(140.0, 2.0)	0.68	(1.9, 0.0)	0.15	(0.0, 0.0)
0.8	425	(140.0, 8.0)	0.61	(1.9, 1.3)	0.53	(0.0, 0.0)

Table 7: Results of the inverse methodology (% error)

5. Conclusions

A predictor - corrector method for solving inverse convection problems was optimized and analyzed against numerical results. The inverse method was previously developed and the search shape was originally based upon physical intuition [? ?]. A new search shape was selected based upon optimization results. The new search shape reduces the number of required sampling points from five to three while eliminating the alternative local minimums. The elimination of false minimums also has the effect of significantly reducing the run time of the inverse methodology by allowing for more efficient optimization techniques. The new data points decreases the error in location prediction from a maximum of 10% [?] down to less than 1%.

While the predictor - corrector method was furthered by this study, the methodology can still be advanced. One of the potential paths to continue is a full error analysis of the domains, developing an envelope of acceptable error. That is to say, determine the range of parameters in which the method will result in a certain acceptable error. Another option would be to extend the methodology to similar inverse convection problems such as the jet in a crosswind or a plume in an enclosure.

The optimize search shape highlights the requirements of needing a new optimization search shape for each experimental configuration to get the

maximum performance of the inverse algorithm. Another potential problem with the algorithm is it does not easily lend itself to transient conditions and thus requires steady state situations only. This comes with an upside as it is able to solve for many boundary conditions simultaneously.

Fundamentally this predictor - corrector method is very sound, however it needs to be matured to a full three dimensional system with the possibility of transient solutions as well. If these features could be developed, the method would have much more far reaching applications such as fires in urban environments [? ?].