5-2017

Maximum Likelihood Temperature/Emissivity Separation of Hyperspectral Images with Gaussian Distributed Downwelling Radiance

David A. Neal
Utah State University

MAXIMUM LIKELIHOOD TEMPERATURE/EMISSIVITY SEPARATION OF HYPERSPECTRAL IMAGES WITH GAUSSIAN DISTRIBUTED DOWNWELLING RADIANCE

by

David A. Neal

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Electrical Engineering

Approved:

Todd K. Moon, Ph.D. Major Professor	Gus P. Williams, Ph.D. Committee Member
Jake Gunther, Ph.D. Committee Member	Scott Budge, Ph.D. Committee Member
Rajnikant Sharma, Ph.D. Committee Member	Mark R. McLellan, Ph.D. Vice President for Research and Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2017

ABSTRACT

Maximum Likelihood Temperature/Emissivity Separation of Hyperspectral Images with Gaussian Distributed Downwelling Radiance

by

David A. Neal, Doctor of Philosophy

Utah State University, 2017

Major Professor: Todd K. Moon, Ph.D.
Department: Electrical and Computer Engineering

Hyperspectral images contain information regarding temperature and emissivity of the materials in the images. Temperature/emissivity separation describes a class of algorithms that extract this information from the hyperspectral data. These algorithms are often only accurate to within a scaling or shift parameter due to the bi-linearity of the hyperspectral measurement model. A new hyperspectral image model is developed to resolve this ambiguity. By modeling the downwelling radiance as Gaussian distributed, emissivity is introduced into the variance of the observations separate from temperature. It is demonstrated that this indeed reduces ambiguity in the estimates.

A maximum likelihood approach is used to estimate the temperature and emissivity. In characterizing the objective function and testing traditional algorithms, a ridge feature is identified, upon which the likelihood must be maximized. A new optimization approach is developed, not existing in the current body of literature, to find the optimal value along such a ridge which can be applied to both this problem and any other problem with an optimal value located on a ridge.

This new optimization approach forms the basis for an algorithm that quickly finds temperature and emissivity estimates. Algorithm performance is characterized and estimator bias is considered.

(196 pages)

PUBLIC ABSTRACT

Maximum Likelihood Temperature/Emissivity Separation of Hyperspectral Images with Gaussian Distributed Downwelling Radiance

David A. Neal

Hyperspectral images are made up of energy measurements at different wavelengths of light. The case is considered where these measurements are dependent on temperature, the self-emitted energy (emissivity), and reflected energy (downwelling radiance) from the surroundings. The process where the downwelling radiance is fixed and the temperature and emissivity are estimated is referred to as temperature/emissivity separation.

Due to the way these terms mix, for a given set of measurements, there exist many pairs of temperatures and emissivities that satisfy the model. This creates ambiguity in the solution that must be resolved for the result to have any significance.

A new model is developed which reduces this ambiguity. This model is used to form an objective function. The temperature and emissivity which maximize the value of the objective function are solved for given a set of measurements.

As part of the solution, a new algorithm is developed which exploits the shape of the objective function to estimate the temperature and emissivity quickly and accurately. Extensive testing of this algorithm is performed to gain an understanding of its average speed and accuracy.

To those who await me in my future. I have completed this work in the hope that it will prove to be to your benefit.

ACKNOWLEDGMENTS

I want to thank my parents for always seeing me better than I am and encouraging me every step of the way, my advisor Dr. Todd Moon for helping me to discover the wonders of signal processing, the joys of doing crazy math, and for supporting me and helping me work through the easy and hard parts of my research, Andrew Pound, Mohammad Shekaramiz, and Trevor Landeen for reviewing my math, code, and sanity as was necessary, and especially my Father in Heaven, for making me equal to this task that so often seemed beyond me.

David Alexander Neal

CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

ACRONYMS

IR	Infrared
LWIR	Long-Wave Infrared
ML	Maximum Likelihood
TES	Temperature/Emissivity Separation

CHAPTER 1

INTRODUCTION

Remote sensing is the art of extracting information from an object without coming into contact with it [1]. Hyperspectral imaging is a type of remote sensing that can be used to detect features or materials in terrain, including such applications as identifying man-made materials in natural scenes to assist in search and rescue, detecting specific plants in counternarcotics operations, and detection of military vehicles in defense applications [2]. Other applications can include food safety, monitoring of pharmaceutical processes, forensic applications [3], and mineral mapping [4, 5].

Hyperspectral images are obtained through the use of hyperspectral cameras, which detect incident radiance at high spectral resolution across every image pixel. This radiance is sensed at many wavelengths, so the data can be seen as a cube, as shown in Figure 1.1, with two spatial dimensions and the third dimension being wavelength. Each pixel naturally has a spectrum associated with it. Spectral signatures can often be used to identify materials, as many materials have identifiable features in their spectral signatures. Due to the interaction between the atmosphere, the various energy sources, and the materials in an image, actually determining what materials are present in a hyperspectral image can be quite challenging. A variety of techniques exist for extracting or estimating the materials in an image [2, 3, 6–17].

1.1 Hyperspectral Image Model

The relationship between material emissivities in a scene and the incident radiance detected by a hyperspectral camera requires the understanding of the many radiance terms, how the atmosphere impacts them, and how the emissivity of an object relates to the radiance emitted. In order to provide understanding of these different factors, the different terms will be considered individually before summarizing the information into the modeling equation that will be used.

It is important to understand that the critical terms vary with wavelength and that some terms become trivially small at certain wavelengths. While the initial development will include all terms,

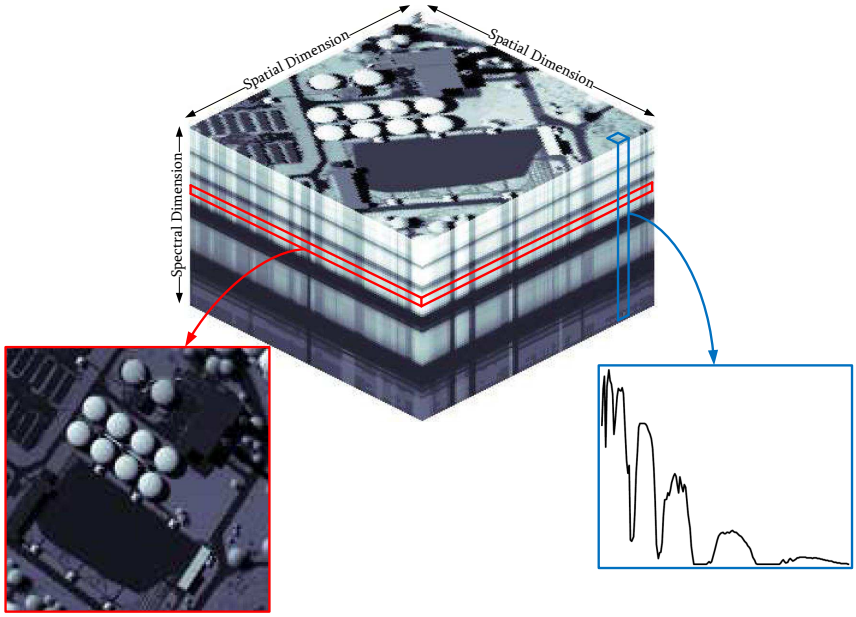

Fig. 1.1: Example Hyperspectral Data Cube

this dissertation will focus on the long-wave infrared region (LWIR). This region, sometimes referred to as the thermal infrared (IR) region, is typically considered to be between 7 and 15 μm, but the actual region considered will depend on the bandwidth of the hyperspectral camera being used [1, 18].

At the highest level, the various sources of radiance detected by a hyperspectral camera may be considered. Figure 1.2 shows the most commonly considered radiance sources. The sun is a major source of radiance detected in hyperspectral images. Source A represents radiance from the sun reflected off of the ground. Source B is radiance from the sun scattered onto the ground and reflected into the camera. Source C is radiance from the sun reflected off the atmosphere into the camera. These are considered the major sources of solar radiance. Radiance from the sun reflected off of other objects, such as buildings, may also reflect off the ground and into the camera, as in source G, but this is typically ignored due to the high attenuation of radiance after the multiple reflections and longer path loss through the atmosphere. Obviously, more convoluted paths to the sensor with more reflections could be considered, but these are even more heavily attenuated than source G and are not considered [1, 18].

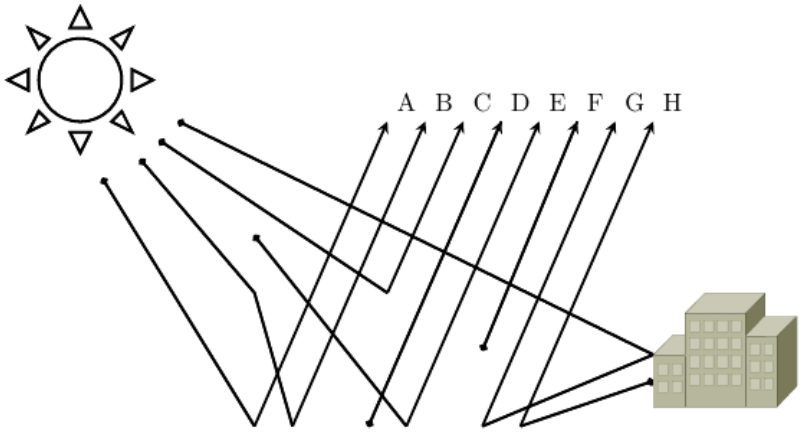

Fig. 1.2: Major Radiance Paths

Objects other than the sun also contribute radiance to the image. Source D indicates the direct path radiance from an object in the scene to the camera. Source E is radiance from the atmosphere reflected off the ground into the camera. Source F is atmospheric radiance directly into the camera. Source H is radiance from some other source reflected off the ground. Source H is typically too small to be considered, similar to source G. Again, more complicated paths could be considered, but are considered sufficiently attenuated to be ignored [1, 18, 19].

The sources listed are typically broken into two groups, solar sources, comprising sources A, B, C, and G, and thermal sources, comprising sources D, E, F, and H. In the thermal IR region, thermal sources dominate the solar radiance, so only these terms are typically considered. For the purposes of this research, source H will also be ignored. Thus, a simple model for the radiance detected in a pixel of a hyperspectral image is

$$y(\lambda) = L_{SourceD}(\lambda) + L_{SourceE}(\lambda) + L_{SourceF}(\lambda), \tag{1.1}$$

where $y(\lambda)$ is the measured radiance and the other terms are the radiance from the indicated sources. These terms are listed as functions of λ, but in reality, these terms depend on many factors and are instead simply measured at a wavelength λ, which is what the notation intends to capture [1, 18].

In order to expand on this model, each term will be considered separately. First, the $L_{SourceD}(\lambda)$ term will be considered. Three factors contribute to the resulting radiance. The objects have self-

emissivity. This is handled by the spectral emissivity signature and is typically expressed as $\epsilon(\lambda)$. Next, the radiance from the object is affected by the object's temperature. This effect is accounted for by the Planck black body function, which is

$$B(\lambda, T) = \frac{2\pi hc^2}{\lambda^5(e^{\frac{hc}{\lambda kT}} - 1)}. \tag{1.2}$$

Here, T is the temperature, k is the Boltzmann gas constant ($1.38 \times 10^{-23}\text{JK}^{-1}$), h is the Planck constant ($6.62 \times 10^{-34} m^2 kg/s$), and c is the speed of light ($3.0 \times 10^8 m/s$). The final consideration is that radiance emitted from the objects passes through the atmosphere, which attenuates it. This is accounted for with the atmospheric attenuation factor $\tau(\lambda)$. Combining these, the term from source D can be expressed as

$$L_{SourceD}(\lambda) = B(\lambda, T)\epsilon(\lambda)\tau(\lambda). \tag{1.3}$$

This version assumes that only a single object is present in a pixel. This assumption is referred to as the pure-pixel assumption [1, 3, 18]. In many cases, multiple materials are present, so some kind of model is required to address how the materials mix together. However, for the purpose of this dissertation, we will focus on the pure pixel case.

Moving on to source E, this term is the result of downwelling radiance from the atmosphere reflecting off of the material in the scene into the sensor. This means that a downwelling radiance term, $L_d(\lambda)$ is present. This term reflects off of the materials in the scene, so the reflectance $\rho(\lambda)$ must be accounted for. Finally, the reflected radiance must pass through the atmosphere. The atmospheric path is identical to the one for source D, so the same $\tau(\lambda)$ is used [1, 18]. The radiance for source E can be written as

$$L_{SourceE}(\lambda) = \tau(\lambda)L_d(\lambda)\rho(\lambda). \tag{1.4}$$

The two material properties $\rho(\lambda)$ and $\epsilon(\lambda)$ tie (1.3) and (1.4) together. That is, the material is the same in both, so the emissivity and reflectivity must match that material. Further, Kirchoff's law states that for opaque objects, $\rho(\lambda) = 1 - \epsilon(\lambda)$. Thus, we can actually replace the $\rho(\lambda)$ in (1.4) to reduce the number of variables in cases where the materials are opaque [1, 18]. This assumption

will not always hold, but it is true for many materials.

The last term in (1.1) is the upwelling radiance. This term is a combination of all the atmospheric radiance that enters the sensor directly. This term is not critical to identifying the materials, as it does not interact with them. The atmospheric and other factors that impact its value are simply lumped into this term and it is simply expressed as $L_u(\lambda)$.

Combining (1.3), (1.4), and $L_u(\lambda)$, (1.1) can be expressed in more detail as

$$y(\lambda) = \tau(\lambda)B(\lambda, T)\epsilon(\lambda) + \tau(\lambda)L_d(\lambda)\rho(\lambda) + L_u(\lambda). \tag{1.5}$$

Further, as noted above, when only opaque objects are considered, the reflectivity can be replaced by 1 minus the emissivity.

1.2 Temperature/Emissivity Separation

The model, then, becomes a function of the given wavelength, the temperature of the material, the emissivity of that material, the downwelling radiance incident on the pixel, the upwelling radiance incident on the hyperspectral camera, and the atmospheric attenuation, which depends on the distance to the camera.

In trying to extract or estimate the material in a pixel using (1.5), the temperature must either be assumed or extracted as well. The process of estimating this temperature along with the emissivity is referred to as Temperature/Emissivity Separation (TES). A variety of approaches to TES exist [12, 20–30]. This dissertation is focused on developing TES approaches.

One of the major issues with TES algorithms is the bi-linearity of (1.5). Due to the way that the Planck function (or in other words, the temperature) and emissivity enter into the equation, the magnitude of one variable can be changed and a corresponding change in the other variable will result in an identical, or almost identical, output. In other words, there are multiple solutions to (1.5), which can result in solutions with an arbitrary shift from the true value. This is due to a fundamental ambiguity in the TES problem [31–33].

To illustrate this, consider fixing the atmospheric parameters $\tau(\lambda)$, $L_d(\lambda)$, and $L_u(\lambda)$ through

some independent estimation, as is often done in other algorithms [10–12, 18]. For a given observation, the emissivity is determined to be

$$\epsilon(\lambda) = \frac{y(\lambda) - L_u(\lambda) - \tau(\lambda)L_d(\lambda)}{\tau(\lambda)(B(\hat{T}, \lambda) - L_d(\lambda))}. \tag{1.6}$$

There is nothing to tie a measurement down. If a value is selected for the temperature, a corresponding value for emissivity is found. For another value of temperature, a different emissivity is found. The only way a solution is rejected is if the emissivity is not between 0 and 1.

A number of methods are used to perform TES and several approaches exist for resolving this ambiguity. For some, the ambiguity remains after a solution is found [18]. Approaches exist which remove the ambiguity using filters [26], make use of reference measurements at the pixel locations [24, 26, 32, 34–37], make use of expected known water spectral features to find reference points, which are used to provide a reference shift for the estimate [11, 38], or estimate black-body pixels in the scene to use as a baseline reference [10, 13].

Generally speaking some known or estimated quantity must be used as a reference point in order to adjust the estimate. In many cases, no reference measurements are available.

Further, estimating a quantity, such as a water feature or black-body pixel is subject to error. Noise in the measurements at the water feature wavelengths is added directly to the shift of all other wavelengths. Likewise, the assumption that a black-body pixel is present or even can be identified may be erroneous. In [10], for example, considerable effort is spent trying to improve the estimate of which pixel is actually the black-body reference.

These approaches have worked well, so it is unlikely that this error is large. However, the data itself can potentially be used to remove the ambiguity.

In this dissertation, it is intended to:

- Develop a model or constraint that makes use of the measured hyperspectral data to remove ambiguity in the estimation
- Develop an algorithm that makes use of this approach to perform TES
- Characterize the performance of this algorithm

1.3 Simplified Model and Previous Work

Initially, an algorithm was developed from the model found in [18]. As such, a similar line of logic was followed in order to reduce the model complexity. First, the sensor (camera) is assumed to be close to the scene of interest. This reduces $L_u(\lambda)$ to a negligible amount. Second, this proximity reduces the amount of atmospheric attenuation to zero, yielding $\tau(\lambda) \approx 1$.

This reduces the model to

$$y(\lambda) = B(\lambda, T)\epsilon(\lambda) + (1 - \epsilon(\lambda))L_d(\lambda). \tag{1.7}$$

In many TES approaches, an atmospheric compensation method is used to remove the atmospheric attenuation and upwelling radiance terms, so this could be considered equivalent to applying some approach to remove those terms [18, 39].

For the group of wavelengths in a hyperspectral image, (1.7) can be stacked as

$$\mathbf{y} = \Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon} + \Lambda_{(1-\boldsymbol{\epsilon})}\mathbf{L}_d, \tag{1.8}$$

where

$$\mathbf{y} = \begin{bmatrix} y(\lambda_1) \\ y(\lambda_2) \\ \vdots \\ y(\lambda_N) \end{bmatrix}, \tag{1.9}$$

with N being the number of wavelengths,

$$\mathbf{B}(T) = \begin{bmatrix} B(\lambda_1, T) \\ B(\lambda_2, T) \\ \vdots \\ B(\lambda_N, T) \end{bmatrix}, \tag{1.10}$$

$$\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon(\lambda_1) \\ \epsilon(\lambda_2) \\ \vdots \\ \epsilon(\lambda_N) \end{bmatrix}, \tag{1.11}$$

and

$$\mathbf{L}_d = \begin{bmatrix} L(\lambda_1)_d \\ L(\lambda_2)_d \\ \vdots \\ L(\lambda_N)_d \end{bmatrix}. \tag{1.12}$$

$\Lambda_{\mathbf{x}}$ is a diagonal matrix with the vector $\mathbf{x}$ down the diagonal and zeros in the off-diagonals.

In (1.8), observations are on hand and the estimate of T and $\boldsymbol{\epsilon}$ is to be found for some set of wavelengths. This requires knowing or finding $\mathbf{L}_d$, the downwelling radiance.

In most approaches, some sort of atmospheric model, such as MODTRAN, is used to generate an estimate of the downwelling radiance [10, 11, 13, 26, 34, 40, 41]. In contrast, [18] estimates an initial downwelling radiance from the data and then considers the downwelling as an additional parameter to estimate with the algorithm. In [42], it is shown that, for some algorithms, the estimate of temperature and emissivity is not particularly sensitive to error in the downwelling, but it does have some impact.

The algorithm developed in [18] linearizes the model and poses it as a convex problem around a current point. Some initialization is used as a starting point, the linearized convex model is solved, and the point is updated. This is repeated until convergence.

At this point, some method to remove the ambiguity in the model is needed. Some obvious approaches are to try similar registration methods with water spectral features or black-body pixels, as other algorithms use. A pixel could be fixed in the solution and then solving could proceed as normal. This might have some utility, but it suffers from all the potential problems previously mentioned. At this point, the model development diverges from the current literature and existing approaches.

1.4 A New Model

As a starting point, it is observed that, in general, only the downwelling radiance is being approximated. When estimating it from the data, there will be some noise, either from how we estimate it or from the observation noise. MODTRAN models, while accurate, will not provide the exact downwelling. There is strong likelihood that the true downwelling will be at least some small perturbation of the MODTRAN downwelling.

Further, across an image, the downwelling is usually assumed to be constant, having the same value for each pixel and for each observation. However, due to minor atmospheric variations, the downwelling component in each measured pixel may realistically be expected to have some variation in it. Based on this assumed variation, a statistical model is considered in which $\mathbf{L}_d$ is modeled as a random variable having a known mean and covariance. This additionally inherently accounts for the variation of the downwelling from whatever estimate that might have otherwise been used.

The covariance is assumed to have some correlated elements. The statistical structure structure of $\mathbf{L}_d$, its mean and variance, may be extracted, for example, from the radiative transfer model MODTRAN [43], or using atmospheric profile measurements.

If it is assumed that the downwelling is Gaussian distributed, the modeled variance of the downwelling can be used to advantage in the TES, since the temperature signal $B(T, \lambda)$ affects only the mean of the observations, while the emissivity $\epsilon(\lambda)$ affects both the mean and the variance of the observations. This variance provides an independent source of information about the emissivity which can reduce the model ambiguity.

Since this model has Gaussian distributed observations that are functions of the unknown parameters of the temperature and emissivity, one approach that can be used is maximum likelihood (ML). This is the approach that will be developed in this work.

1.5 Existing Algorithms for Comparison

As previously mentioned, a number of existing approaches for performing TES exist. The assumptions and algorithmic approaches for optimizing in each method are a mixture of common elements used by almost all techniques and unique tweaks and assumptions to provide simplification

or allow for better optimization. The model developed in this dissertation likewise contains unique elements and common features. While an exhaustive development of every approach is prohibitive, several algorithms have been chosen from the body of literature for a brief comparison with the model that will be developed in this dissertation.

While it is hard to say that the results of any particular set of models can truly be compared due to the assumptions explicit in each, these same models will be used later on as a performance reference. Critical features affecting the accuracy of each approach will be discussed to provide a better context for comparing the results in later chapters.

It is also important to note that the materials, atmospheric correction techniques, artificial versus measured data, and choice of sensor for measured data can all strongly impact performance. Relevant details will be discussed later in the results comparison.

1.5.1 Iterative Spectrally Smooth TES

Iterative Spectrally Smooth Temperature Emissivity Separation (ISSTES) is reviewed by Ingram and Muse to determine the error resulting from algorithmic assumptions [12]. The algorithm starts with the model found in (1.5). The upwelling radiance, downwelling radiance, and atmospheric attenuation are generated by MODTRAN. An inversion process is then used to map the downwelling radiance back to surface temperature and emissivity. A family of emissivity curves are generated for a number of temperatures. The algorithm uses a spectral smoothness measure to pick a curve from this family, which is the estimated emissivity with the corresponding estimated temperature. This selection is based on a number of assumptions about the smoothness of the emissivity curve in the true model [12].

While a similar starting model is used, a number of clear differences exist between ISSTES and the proposed algorithm. The ISSTES estimates the upwelling radiance, atmospheric attenuation, and downwelling radiance using MODTRAN. The former two are removed by assumption in the proposed approach and the latter is treated as a Gaussian random variable. It is also important to note that the key elements of the ISSTES algorithm are the inversion step and the fact that it seeks a spectrally smooth optimization. The ML optimum solution is likely not to be the same as the

spectrally smooth optimum.

1.5.2 Linearized Iterative Convex Optimization

The iterative convex approach proposed in [18] provides an interesting point of comparison. The general operation of the algorithm was previously discussed above, so the focus here is on the assumptions of the algorithm.

Key in comparing this approach to the proposed model is that the same simplifications removing the upwelling radiance and atmospheric attenuation are present. This provides a common starting place. The model is then linearized, both by using a linear approximation of the Planck function and by using a Taylor series approximation of the overall model. As such, any comparison of results will be shaded by the accuracy of those assumptions versus the proposed model.

The iterative convex approach also estimates the downwelling radiance, while the proposed approach treats the downwelling radiance as a random variable. This provides an additional variable over which to optimize in comparison to the proposed model, which could provide variation in the results.

The choice of dataset is also critical. As will be developed later, the proposed model is currently only set up to use pixels with common temperatures and emissivities. These most certainly would represent pixels from a single image. The iterative convex approach specifically is set up to take advantage of pixels over time, where the temperature has varied in the scene. This provides additional variation in the model.

Finally, like the ISSTES algorithm, the iterative convex approach uses a specific optimality criterion. The convexity of the observations iterated into the linear model is not likely to be reach the same optimal point as an ML approach. It is also unclear how the linearization interacts with convex optimality. There is no guarantee that the convex optimum point after linearization corresponds to an optimal point before linearization.

1.5.3 TES Retrieval Using Linear Spectral Emissivity Constraint

This algorithm uses a linear fit of the emissivity as a constraint. The algorithm uses the same model for the observations, removing the atmospheric attenuation and upwelling radiance on the

assumption of accurate atmospheric correction. However, the downwelling radiance is not modeled as a random variable [24].

The algorithm begins by estimating the temperature from the observations. Any method is acceptable, but using the maximum surface brightness temperature is put forth as a likely approach [24].

A piecewise linear approximation of the emissivity is used to model the emissivity spectra over a subset of wavelengths. Since the temperature has already been estimated, this estimate is used in solving for the line parameters of the linear model. A MODTRAN estimate of the downwelling radiance is also used. A least squares approach can be applied to solve for the parameters [24].

Next, a cost function is used. The particular cost function is the summed squared error between the at-ground radiance and the estimated radiance using the temperature and emissivity estimates already obtained. The first-order partial derivative of the cost function is calculated with respect to the temperature and used as an incremental step of the temperature. If this step is small enough, the algorithm stops and the current temperature and emissivity estimates are accepted. Otherwise, the temperature estimate is updated and the algorithm begins again [24].

As far as comparison goes, this approach, like ISSTES, uses MODTRAN in order to estimate the downwelling radiance, as opposed to treating it as a Gaussian variable. The linear fit will introduce variation into the solution, depending on how much it varies from the true fit. Finally, this algorithm uses a summed squared error optimality condition, which again will be different than ML optimality.

1.5.4 Stepwise Refining of TES

Another approach to TES is found in [34]. This approach is of particular interest because the authors do a direct comparison with ISSTES [34].

This approach begins by assuming accurate atmospheric correction, so the model is the same as the proposed one, excepting the downwelling radiance is not considered Gaussian. Similar to [18], the Planck function is approximated with a linear function, although in this case, this is only performed over a small spectral region. Further, it is linearly approximated as a function of the wave-

length. Emissivity is considered to change slowly as a function of the wavelength, so it is approximated as a constant over this same small spectral region of a few wavelengths. Due to the high variability in the downwelling radiance, it cannot be approximated by a linear function of the wavelength. An initial guess of the emissivity is made and, if that guess is accurate, a residue calculated from the emissivity will be minimized. The downwelling radiance is estimated by MODTRAN. The temperature is estimated directly from the observations using the estimated downwelling radiance and emissivity, including inverting the Planck function. [34].

The actual approach is performed in several narrow spectral regions, each being optimized. As an optimal emissivity is estimated for each region, new residues are calculated around the newest estimate with smaller step sizes between each residue, allowing the algorithm to iteratively step closer to the best estimate and allowing the user to select the final resolution [34]

Again, this algorithm differs from the proposed model in the use of MODTRAN to directly estimate the downwelling radiance instead of treating it as a Gaussian random variable. The choice of assuming constant emissivity and linearizing the Planck function over the narrow bands will also introduce some inaccuracy, depending on how closely these assumptions match the true data set. Finally, the residue optimality again provides a different criterion for optimality over the ML approach.

1.5.5 MODTES

This method starts with (1.5) as its model. Atmospheric correction is performed by estimating the upwelling radiance and atmospheric attenuation using MODTRAN. Graybody pixels, identified by their normalized difference vegetation index, and a water vapor profile are used to estimate the brightness temperature. This estimate is used to refine the MODTRAN-estimated upwelling radiance and atmospheric attenuation. A model for downwelling radiance, involving scaled path radiance, transmittance, and viewing angle, is used to estimate the downwelling radiance [38].

With the atmospheric correction taken care of and the downwelling radiance estimated, the model is again similar to the proposed model, and again, the difference is in the downwelling radiance not being modeled as a Gaussian random variable. The temperature and emissivity estimation

proceeds by selecting a maximum value for emissivity that is common for all wavelengths. This is used with the estimated downwelling radiance to provide an initial temperature estimate for three wavelengths. The maximum temperature from among those estimates is then used to estimate the entire emissivity spectrum. Those emissivities are then used to replace the previous maximum value for emissivity at their corresponding wavelength. The process is then iterated until it converges [38].

A ratio is calculated of the converged emissivities and their average value. The difference between the maximum and minimum of these ratios is found and used to find, by regression, the minimum emissivity. This minimum emissivity is then used to rescale the estimated emissivity spectrum. Finally, the maximum of the estimated emissivities is used to estimate the temperature [38].

As with other methods, this approach differs from the proposed model in its use of MODTRAN to estimate the atmospheric parameters. It additionally has correction of the MODTRAN estimates, which would further differentiate the atmospheric correction. The downwelling radiance is, as previously noted, not a Gaussian random variable. The estimation technique will be optimal in a different sense. In this case, it is not entirely clear in what sense it will be optimal. Regardless, results will vary from the optimal ML solution.

1.5.6 Variation of Proposed Model from Existing Methods

While the list of methods reviewed above is far from exhaustive of the literature, the assumptions and approaches provide a good view of the amount of variation present in the various approaches. Primarily, existing methods estimate the downwelling radiance using MODTRAN or some other approach. The proposed model treats downwelling radiance as a Gaussian random variable. While there is some similarity in the way the proposed model treats the atmospheric attenuation and the upwelling radiance, the proposed model also allows for it to be ignored. This assumption impacts the model differently than estimating the upwelling radiance and atmospheric attenuation, as is done in the reviewed approaches above.

1.6 Dissertation Outline

Building on this, the objectives are refined. In this dissertation, it is intended to specifically contribute:

1. A new statistical model for hyperspectral image pixels
2. A novel optimization technique for finding a maximum located on a long, relatively flat ridge (or a minimum in a valley)
3. An algorithm to solve for the maximum likelihood solution given the new statistical model

Chapter 2 will develop the new statistical model. The desired parameters in the model, the temperature and emissivity, will be estimated using the ML approach. An exhaustive literature search has shown no work making use of this approach or using maximum likelihood techniques to perform TES.

The objective function will be examined to identify any potential pitfalls in optimization as well as verifying the presence of a unique solution.

Chapter 3 will develop the ML algorithm. Several classical optimization techniques will be applied to demonstrate the impact of the unique features, namely a long flat ridge, of the objective function on optimization.

The ridge feature of the objective will be dealt with using an improved algorithm that exploits classical techniques to provide robust initialization and a novel approach to optimization that provides superior convergence. This approach has the added utility that it can be applied to any objective with similar ridge-like characteristics to this objective function, regardless of the underlying models or motivations. A search of the literature has shown no other examples of this approach being used.

The new algorithm will be used to demonstrate the utility of the variance on the downwelling radiance in reducing ambiguity in the solution.

Chapter 4 will review testing of the algorithm with variation in key parameters to demonstrate its performance, potential, and limitations. It will also include a comparison of the performance of the algorithm with the methods discussed above.

Chapter 5 will address concerns regarding bias and will discuss a general approach to removing bias that could be applied to the algorithm in the future.

Chapter 6 is a summary of the important contributions and identification of potential areas of additional research.

CHAPTER 2
MAXIMUM LIKELIHOOD FRAMEWORK

2.1 Likelihood Functions

As a starting point, consider the multiple wavelength observation model given in (1.8), with the extra consideration of observation noise. For a set of pixels with the same material and temperature, the model becomes

$$\mathbf{y}_i = \Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon} - \Lambda_{(1-\boldsymbol{\epsilon})}\mathbf{L}_{d,i} + \mathbf{n}_i, \tag{2.1}$$

where i indicates the observation number and where $\mathbf{n}$ is zero mean Gaussian noise with variance σ^2. The downwelling radiance $\mathbf{L}_d$ is Gaussian distributed with mean $\boldsymbol{\mu}$ and covariance R. There will likely be some correlation in how the downwelling radiance varies as the mean shape will be consistent . As such, the covariance matrix R should have non-zero off-diagonal elements. This will add additional complexity in the likelihood function, but also potentially adds additional information.

A set of observations comes from the pixels in the same image, as shown in Figure 2.1. This figure shows a scene with some material of interest. There are a number of pixels of this material that have been observed, indicated in the figure as $\mathbf{y}_1$ and $\mathbf{y}_2$. These form the set of observations for the algorithm; in this case, there are 2 observations in the set. Each pixel has the identical model described in (2.1), with each pixel having an independent draw from the downwelling radiance and noise random variables. For this model, it is assumed that all pixels are the same temperature. A set of data could be obtained by trying to identify pixel clusters of the same material. Several clusters or even individual pixels would be sufficient. In this work, it is assumed that the pixels have been identified and observation data has been captured. Each observation is Gaussian distributed with mean $\Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon} - \Lambda_{(1-\boldsymbol{\epsilon})}\boldsymbol{\mu}$ and covariance $\Lambda_{(1-\boldsymbol{\epsilon})}R\Lambda_{(1-\boldsymbol{\epsilon})} + I_n\sigma^2$, where I_n is an $n \times n$ identity matrix.

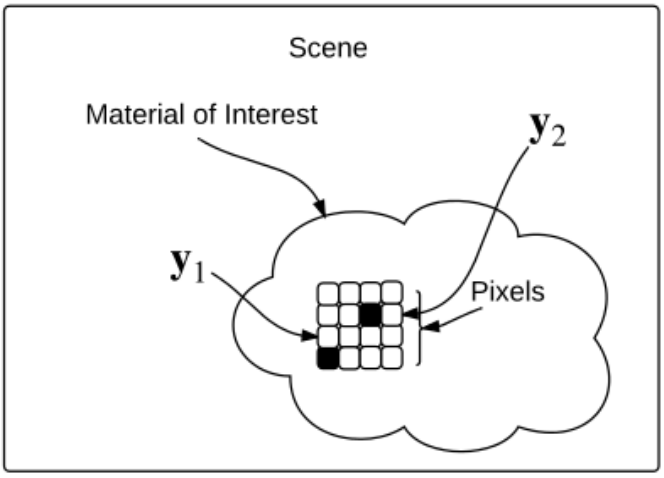

Fig. 2.1: Hyperspectral Image Pixel Observations in a Scene.

As part of the development, the noise-free observations will be considered as a reference of performance. The mean for this case remains the same and the covariance becomes $\Lambda_{(1-\boldsymbol{\epsilon})}R\Lambda_{(1-\boldsymbol{\epsilon})}$.

For both cases, the mean is a function of the temperature T and the vector of emissivities $\boldsymbol{\epsilon}$. In contrast, the covariance is only a function of $\boldsymbol{\epsilon}$. This is key for providing the extra information needed to remove ambiguity in the solution, meaning that across the set of observations, all variation in the observations comes from the downwelling radiance.

The likelihood function for the noise-free set of observations can then be written as

$$\begin{aligned} f(\mathbf{y}_1,\ldots,\mathbf{y}_n|T,\boldsymbol{\epsilon}) &= \frac{1}{(2\pi)^{nN/2}[\Lambda_{(1-\boldsymbol{\epsilon})}R\Lambda_{(1-\boldsymbol{\epsilon})}]^{n/2}}\times \\ \exp[-\frac{1}{2}\sum_{i=1}^{n}&(\mathbf{y}_i-\Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon}-\Lambda_{(1-\boldsymbol{\epsilon})}\boldsymbol{\mu})^T(\Lambda_{(1-\boldsymbol{\epsilon})}R\Lambda_{(1-\boldsymbol{\epsilon})})^{-1}(\mathbf{y}_i-\Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon}-\Lambda_{(1-\boldsymbol{\epsilon})}\boldsymbol{\mu}), \end{aligned} \tag{2.2}$$

where n is the number of observations and N is the number of wavelengths. In almost all cases, the log-likelihood is used, which is

$$\begin{aligned} L(\mathbf{y}_1,\ldots,\mathbf{y}_n|T,\boldsymbol{\epsilon}) &= \frac{nN}{2}\log(2\pi) - \frac{n}{2}\log([\Lambda_{(1-\boldsymbol{\epsilon})}R\Lambda_{(1-\boldsymbol{\epsilon})}]) \\ -\frac{1}{2}\sum_{i=1}^{n}&(\mathbf{y}_i-\Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon}-\Lambda_{(1-\boldsymbol{\epsilon})}\boldsymbol{\mu})^T(\Lambda_{(1-\boldsymbol{\epsilon})}R\Lambda_{(1-\boldsymbol{\epsilon})})^{-1}(\mathbf{y}_i-\Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon}-\Lambda_{(1-\boldsymbol{\epsilon})}\boldsymbol{\mu}). \end{aligned} \tag{2.3}$$

For the case with noisy observations, this changes the log-likelihood function to

$$L(\mathbf{y}_1,\ldots,\mathbf{y}_n|T,\boldsymbol{\epsilon}) = \frac{nN}{2}\log(2\pi) - \frac{n}{2}\log([\Lambda_{(1-\boldsymbol{\epsilon})}R\Lambda_{(1-\boldsymbol{\epsilon})} + I_n\sigma^2]) \\ - \frac{1}{2}\sum_{i=1}^{n}(\mathbf{y}_i - \Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon} - \Lambda_{(1-\boldsymbol{\epsilon})}\boldsymbol{\mu})^T(\Lambda_{(1-\boldsymbol{\epsilon})}R\Lambda_{(1-\boldsymbol{\epsilon})} + I_n\sigma^2)^{-1}(\mathbf{y}_i - \Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon} - \Lambda_{(1-\boldsymbol{\epsilon})}\boldsymbol{\mu}). \tag{2.4}$$

Having the log-likelihood functions in (2.3) and (2.4), the ML approach is used to estimate T and $\boldsymbol{\epsilon}$. The nature of solving this and the exact approach to be used will be addressed in the next chapter.

2.2 Likelihood Function Plots

In order to the maximize the likelihood with respect to the parameters, there must be an understanding of the nature of the likelihood function. While a Gaussian likelihood has a single maximum, the actual surface is the likelihood value for a set of observations as we change the temperature and emissivity values. Examining the actual function values can show the number of extrema in the function and any peculiar characteristics that it may have. Further, functions exist which are extremely difficult to optimize. Some of these, such as the infamous Rosenbrock function, require customized algorithms specifically suited to deal with the unique characteristics of the surface in question [44].

Since a hyperspectral image has many wavelengths, any plot will require $N + 2$ dimensions, one for each wavelength, one for temperature, and one for the likelihood value. Since a plot cannot represent more than three dimensions, a single wavelength is first considered. Ten observations were generated for a single wavelength at 8.3 μm, with the temperature set to 290°K and the emissivity being 0.8114. The downwelling radiance had a mean value of 547.8675 microflicks and covariance of 5.6. For the noisy case, the noise on the observations was $\sigma^2 = 1$.

Figures 2.2 and 2.3 show the likelihood surfaces for the noise-free and noisy observations, respectively. The noisy plot shows slightly more variation, but other than this, the surfaces are very similar. The main characteristic of interest is how they are very flat for low emissivities.

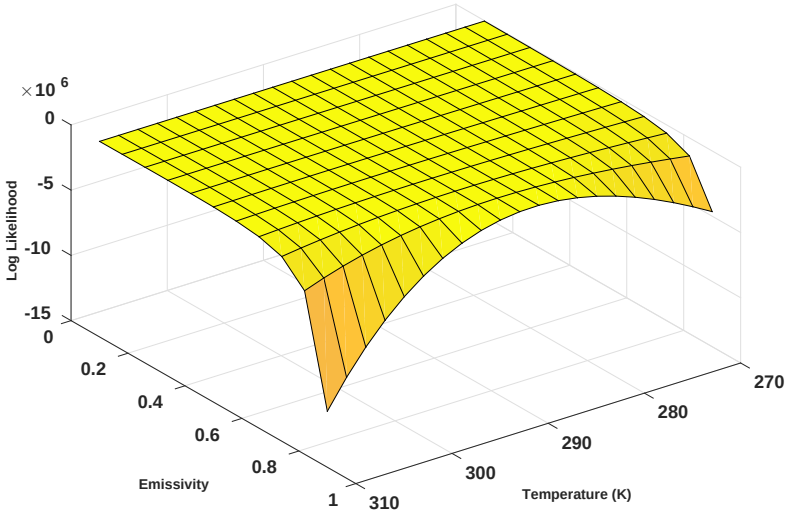

Fig. 2.2: Objective Likelihood Versus Temperature Versus Emissivity with Noise-Free Observations.

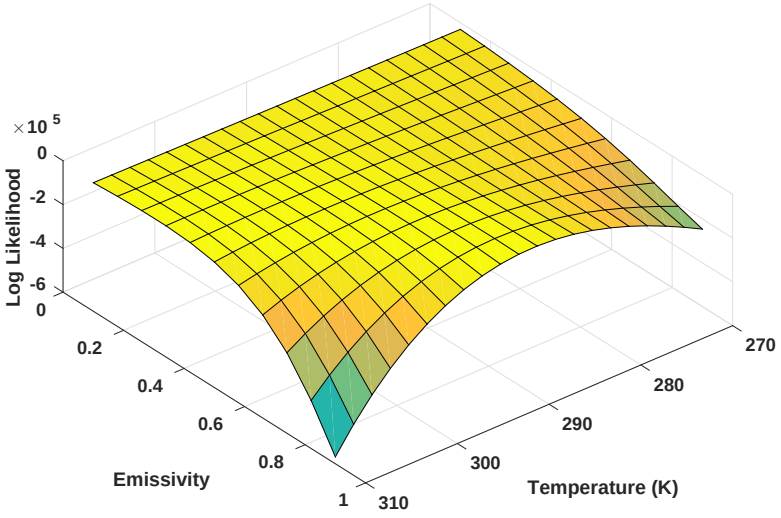

Fig. 2.3: Objective Likelihood Versus Temperature Versus Emissivity with Noisy Observations.

Examining this region closely shows that the region is not flat This is exhibited in Figure 2.4, where the scale is reduced and the emissivity scale is only from 0 to 0.5. The curve is much more pronounced on this scale.

Zooming in, as in Figure 2.5, the function is seen to have a ridge which is much higher than the rest of the surface. Examination of this ridge shows that the ridge peak is also very flat. There is a unique maximum value, but it is very close to the other values of the ridge peak compared to the rest of the surface off of the ridge.

For comparison, a small set of wavelengths was chosen to show how the surface changes when more than one wavelength is considered. In this case, ten observations were again generated, but this time for wavelengths from 8.3μm to 8.7μm in 0.1μm steps. The temperature was again set to $290°$K and the emissivity values were 0.8114, 0.7946, 0.7744, 0.7865, and 0.7549, respectively. The downwelling radiance had a mean vector of 547.8675, 572.2968, 596.0602, 561.5744, and 570.2581, all in μflicks. The covariance matrix was 5.6 down the main diagonal with 0.6 on the off diagonal and 0.5 for all other elements. For the noisy case, the noise on each element of the observations was $\sigma_n^2 = 1$. Again, since plotting seven dimensions is impossible, only two are shown, the first being the wavelength of emissivity and the second being temperature. The other four emissivity values that are not on any axis are set to their true values.

In Figures 2.6 and 2.7, there is a distinct ridge where a certain temperature gives maximum values for any value of emissivity. Also of note in the noise-free case is the distinct roll-off of the surface after a certain point. While the sharpness of this roll-off is due to the resolution of the plot, the surface does indeed decrease after this point. The noise on the observations smooths this out somewhat.

To ensure that there were not any irregularities, other wavelengths were also plotted versus temperature. Figures 2.8 and 2.9 show examples of these results. It is clear that these are almost identical to the other wavelength. Indeed, all five wavelengths only vary slightly in magnitude. As such, additional plots are not shown, as they are essentially the same as the plots below and above.

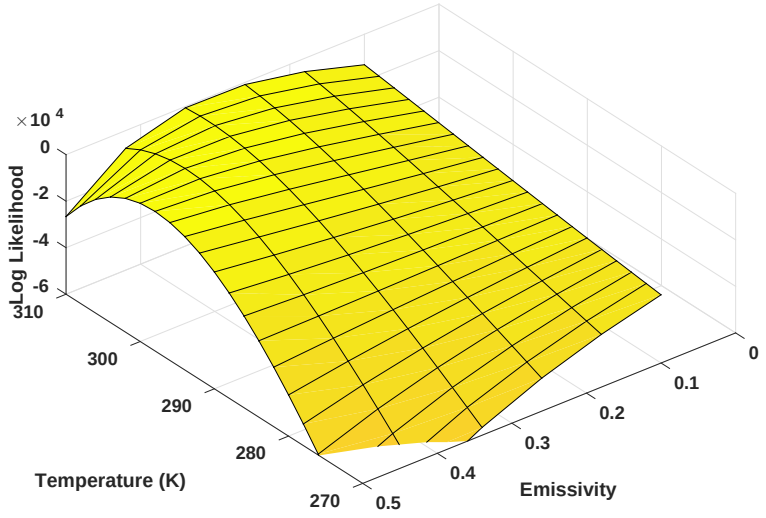

Fig. 2.4: Zoomed Objective Likelihood Versus Temperature Versus Emissivity with Noisy Observations.

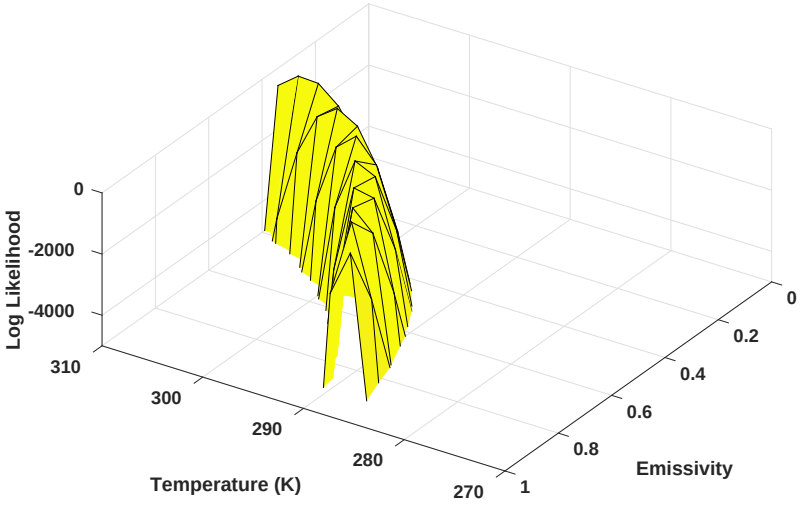

Fig. 2.5: Ridge Behavior of Objective Likelihood Versus Temperature Versus Emissivity with Noisy Observations.

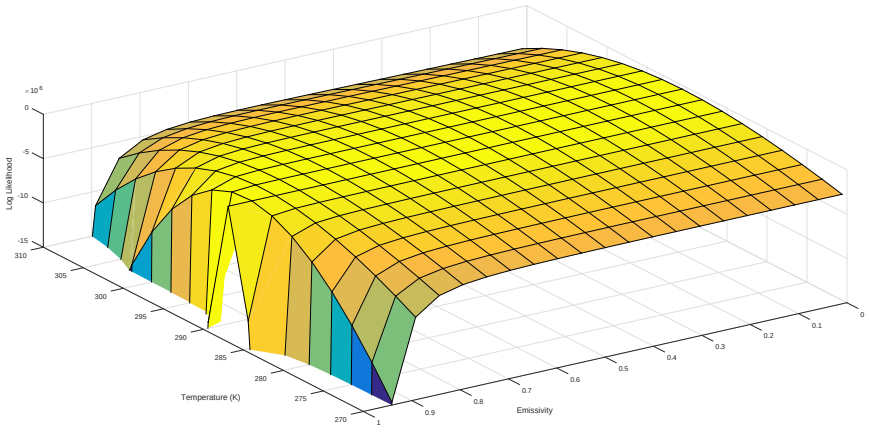

Fig. 2.6: Five Wavelength Objective Likelihood Versus Temperature Versus Emissivity with Noise-Free Observations.

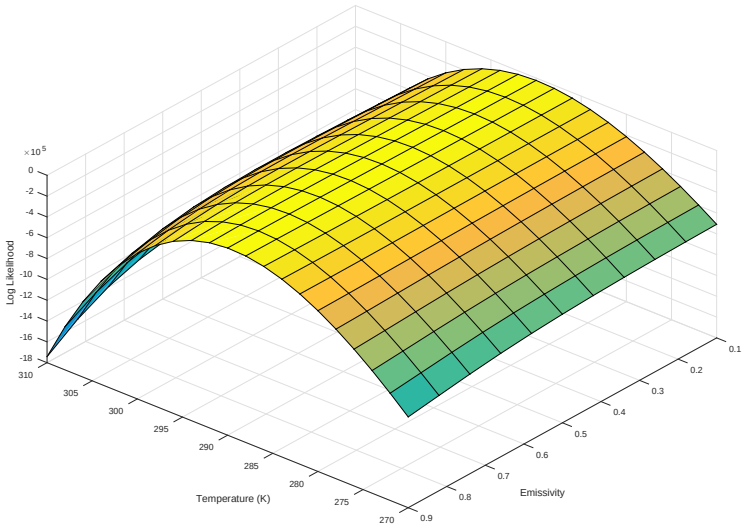

Fig. 2.7: Five Wavelength Objective Likelihood Versus Temperature Versus Emissivity with Noisy Observations.

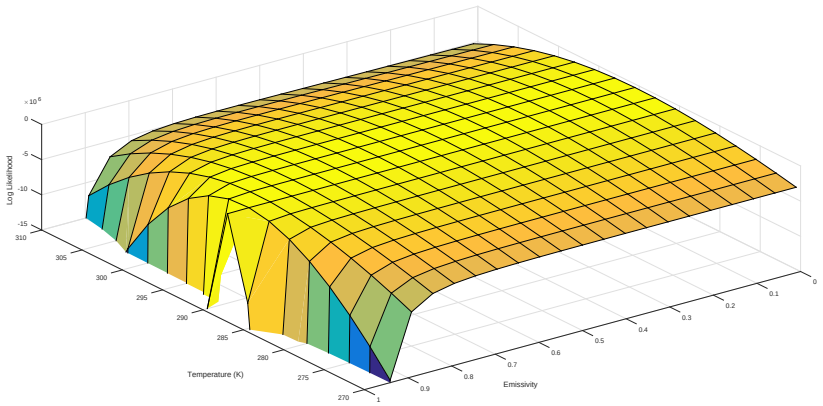

Fig. 2.8: Five Wavelength Objective Likelihood Versus Temperature Versus Emissivity with Noise-Free Observations at Different Wavelength.

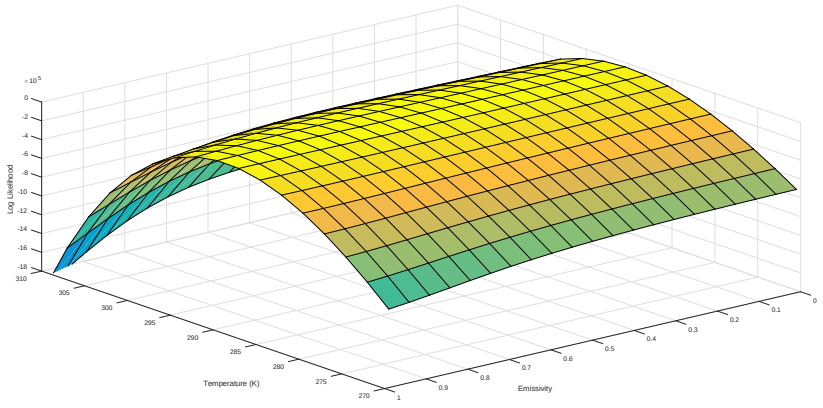

Fig. 2.9: Five Wavelength Objective Likelihood Versus Temperature Versus Emissivity with Noisy Observations at Different Wavelength.

To show how the emissivities relate to each other in the surface, two wavelengths are also plotted with the other emissivities and the temperature fixed to their true values. This provides an emissivity versus emissivity plot with the third dimension showing the likelihood.

Figures 2.10 and 2.11 show that instead of a ridge, there is a distinct maximal point. This suggests that the ridge-like behavior is due to interactions between temperature and emissivity. As have noted previously, there is a valid solution to (1.7) for (almost) any temperature. From the distinctness of the peaks in these emissivity versus emissivity plots it can be seen that for a given emissivity value at one wavelength, the maximum likelihood choice for the emissivities at other wavelengths is more distinct. There is still slight ridge-like behavior parallel to the axes where the other axis takes on the true value. Again a sharp roll-off is seen in the noise-free plot. In this case, the roll-off occurs after the emissivity passes the true value in one axis or the other.

Figures 2.12 and 2.13 zoom in on the likelihood-axis scale to show the peak values. It is clear from these figures that there is a peak value and it occurs at the same peak indicated in the emissivity versus emissivity figures.

It is also important to understand that the peak values in Figures 2.6 through 2.13 are only the peak values given the values to which the emissivities and/or temperatures are set that are not plotted on the shown axes. A maximum value exists for each parameter given the other values. Having determined this, an approach to finding temperature and set of emissivities that give us the maximum likelihood for a given set of observations can now be developed.

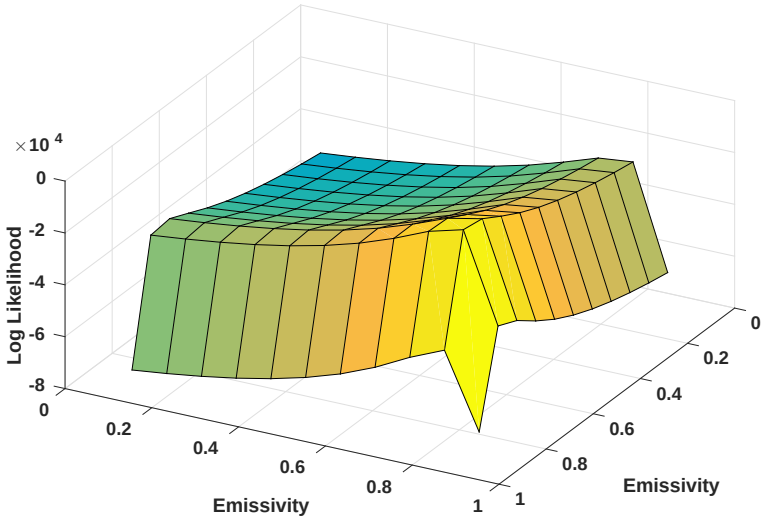

Fig. 2.10: Five Wavelength Objective Likelihood Versus Emissivity Versus Emissivity with Noise-Free Observations.

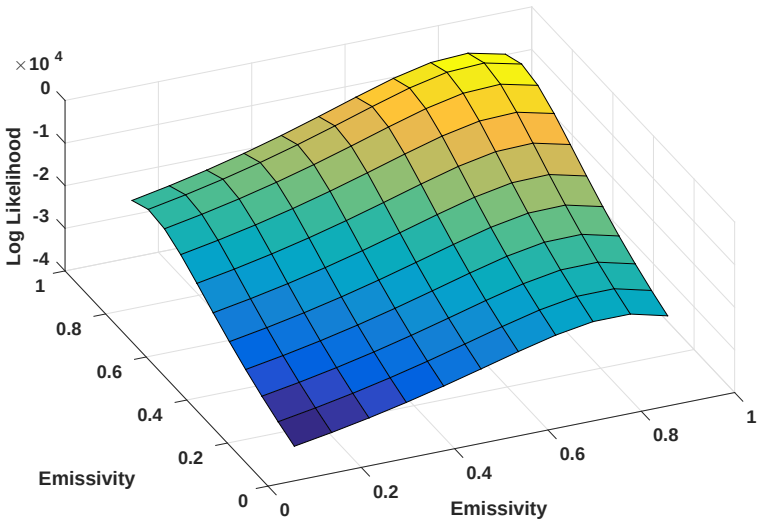

Fig. 2.11: Five Wavelength Objective Likelihood Versus Emissivity Versus Emissivity with Noisy Observations.

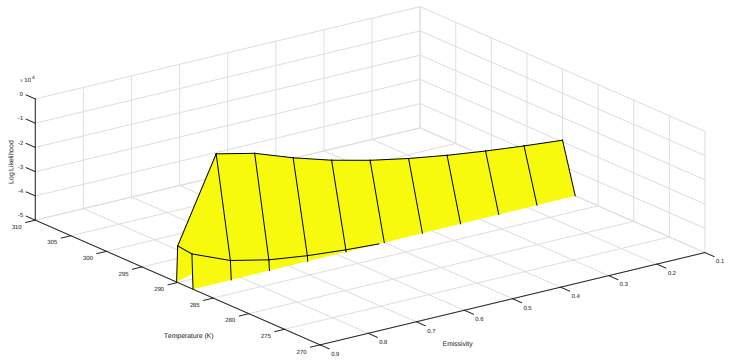

Fig. 2.12: Peak Value for Five Wavelength Objective Likelihood Versus Temperature Versus Emissivity with Noise-Free Observations.

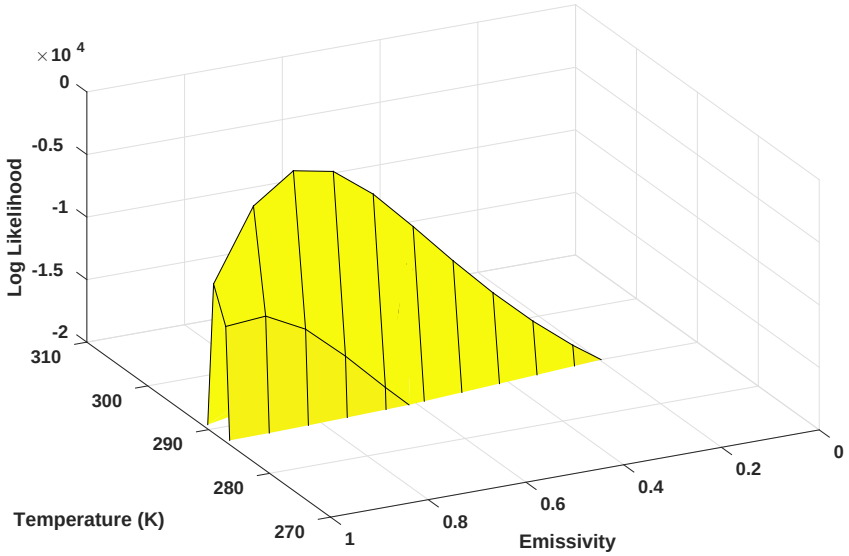

Fig. 2.13: Peak Value for Five Wavelength Objective Likelihood Versus Temperature Versus Emissivity with Noisy Observations.

CHAPTER 3
ALGORITHM DEVELOPMENT

3.1 Introduction

In this chapter, the objective and constraints are identified, and then several approaches to solving the problem are investigated. The overall goal is to get the best performance, but speed and computational efficiency are also desirable. No strict requirements were placed on these goals to start.

In order to find the maximum likelihood for a set of observations, the problem needs to be defined clearly to understand the objective function and its constraints. First, the goal is to maximize the log-likelihood. As for constraints, the only constraint is on the emissivity, which can only take on values between zero and one. As such, the problem can be stated as

$$\hat{\boldsymbol{\epsilon}}, \hat{T} = \underset{(\boldsymbol{\epsilon},T)}{\operatorname{argmax}} \quad L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon},T) \quad \text{s.t} \quad 0 \leq \epsilon_i \leq 1 \quad i = 0,1,...,n, \tag{3.1}$$

where $L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon},T)$ is (2.3) or (2.4), depending on the whether or not the noisy observations are used.

3.2 Gradient Ascent

Some initial attempts to directly solve were made. It is easy to see that the non-linearity of the Planck function will be problematic in solving for temperature. Additionally, the correlation in the downwelling radiance causes mixing between the emissivities at different wavelengths. Gradient ascent was tried next. This was initially explored in [45], but is more fully developed here.

Gradient ascent requires taking the partial derivative of the objective function with respect to all variables. A step is then taken in the direction of the gradient, which points uphill. A backtracking line search approach is used to adjust the step length and ensure that the likelihood function was

increasing at each step [46]. The line search was simplified somewhat to only compare the new likelihood value to the value at the previous iteration.

3.2.1 Partial Derivatives

The partial derivatives are derived in the appendix. For the noise-free case, the derivative with respect to ϵ_ℓ, the ℓth wavelength, is

$$\frac{\partial L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon},T)}{\partial \epsilon_\ell} = \frac{N}{1-\epsilon_\ell} - \sum_{i=1}^{N} \frac{(y_{i,\ell} - B(T)_\ell)}{(1-\epsilon_\ell)^2} \mathbf{e}_\ell^T R^{-1} \mathbf{w}_i, \tag{3.2}$$

where $\mathbf{w}_i = (\mathbf{y}_i - \Lambda_{B(T)}\boldsymbol{\epsilon} - \Lambda_{(1-\boldsymbol{\epsilon})}\boldsymbol{\mu})\Lambda_{1/(1-\boldsymbol{\epsilon})}$ and $\mathbf{e}_\ell$ is an elemental vector with a 1 in the ℓth position.

The partial derivative with respect to temperature is

$$\frac{\partial L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon},T)}{\partial T} = \sum_{i=1}^{N} \mathbf{w}_i'^T \Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})} R^{-1} \mathbf{w}_i, \tag{3.3}$$

where the kth element of $\mathbf{w}_i'$ is the derivatives of the Planck function at the kth wavelength, given as

$$\frac{\partial B(T)_k}{\partial T} = \frac{C_1 C_2 e^{C_2/(\lambda_k T)}}{\lambda_k^6 T^2 (e^{C_2/(\lambda_k T)} - 1)^2}. \tag{3.4}$$

For the noisy case, partial with respect to ϵ_ℓ can be written as

$$\begin{aligned}\frac{\partial L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon},T)}{\partial \epsilon_\ell} =& \frac{N}{1-\epsilon_\ell} - N\frac{\sigma^2 X^{-1}_{\ell,\ell}}{(1-\epsilon_\ell)^3} - \\ & \sum_{i=1}^{N} \frac{(y_{i,\ell} - B(T)_\ell)}{(1-\epsilon_\ell)^2} \mathbf{e}_\ell^T X^{-1} \mathbf{w}_i - \sigma^2/(1-\epsilon_\ell)^3 \mathbf{w}_i^T X^{-1} E_{\ell,\ell} X^{-1} \mathbf{w}_i,\end{aligned} \tag{3.5}$$

where $X = R + \sigma^2 \Lambda_{1/(1-\boldsymbol{\epsilon})^2}$.

The partial with respect to T is given as

$$\frac{\partial L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon},T)}{\partial T} = \sum_{i=1}^{N} \mathbf{w}_i'^T \Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})} X^{-1} \mathbf{w}_i, \tag{3.6}$$

In both the noise-free and noisy case, the n partial derivatives with respect to ϵ_ℓ and the single partial with respect to T are stacked into an $n+1$ length vector in order to form the gradient.

3.2.2 Enforcing the Emissivity Constraint

Initially, the emissivity constraint was enforced by manual checking while adjusting the step size in the backtracking line search. If the current step went out of bounds, the step size was reduced before checking the backtracking constraint. Once a valid step was obtained, the backtracking line search then proceeded as normal. This tended to run very slowly as there were a large number of iterations on the step size.

In an effort to remove this looping behavior, the sigmoid function was used by replacing the n elements of $\boldsymbol{\epsilon}$ with

$$\epsilon_\ell = \Phi(\alpha_\ell) = \frac{1}{1+e^{-\alpha_\ell}}. \tag{3.7}$$

The value of α_ℓ can take on any value while restricting the value of ϵ_ℓ to between 0 and 1. We then can iterate on $\boldsymbol{\alpha}$ and calculate $\boldsymbol{\epsilon}$ using (3.7). This turns the constrained optimization problem into an unconstrained optimization problem.

When calculating the gradient, this does not impact the partial derivative with respect to T, as the old $\boldsymbol{\alpha}$ is simply being used and thus $\boldsymbol{\epsilon}$ is on hand. However, the partial derivative with respect to ϵ_ℓ must be replaced with the partial derivative with respect to α_ℓ.

If the chain rule is applied, the partial derivative can be rewritten as

$$\frac{\partial L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon},T)}{\partial \alpha_\ell} = \frac{\partial L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon},T)}{\partial \epsilon_\ell}\frac{\partial \Phi(\alpha_\ell)}{\partial \alpha_\ell}. \tag{3.8}$$

As this shows, multiplying the (3.2) or (3.5) by the derivative of (3.7) with respect to α_ℓ finds the partial of the likelihood function with respect to α_ℓ. This derivative is

$$\frac{\partial \Phi(\alpha_\ell)}{\partial \alpha_\ell} = \Phi(\alpha_\ell)(1-\Phi(\alpha_\ell)) \tag{3.9}$$

3.2.3 Gradient Ascent Algorithm

The last missing piece for the algorithm is the initial conditions. For the algorithm, each

element of $\boldsymbol{\epsilon}$ was initially set to 0.5. There is no prior information, so it is assumed that the emissivity can be spread uniformly throughout the range 0 to 1, with the mean value chosen. Because of the iterating on $\boldsymbol{\alpha}$, each element of $\boldsymbol{\alpha}$ is set to 0 and the $\boldsymbol{\epsilon}$ values are then 0.5.

For the temperature, the choice of initial temperature is somewhat arbitrary. For most of the tests, several arbitrary starting values around the true temperature were selected. There were problems where the algorithm did not converge well for certain starting temperatures, which will be discussed in more detail below. Generally speaking, for testing the algorithm, an initial temperature was chosen that was five to ten degrees larger than the true value. For real data, the ambient temperature might be a good starting choice, assuming that it is available, but for some specific cases that were tested, even the true temperature was not an acceptable initial temperature, so temperature initialization could be of concern when using this algorithm.

Finally, the choice of initial step size was set to 10^{-3}. This was chosen based on the observed step size in the backtracking line search for preliminary tests. This was not an issue with the noisy observations. Generally speaking, the step size would be refined based on typical scaling required to get the gradient down to an appropriate size to ensure an increase in the objective function. However, the actual performance of the algorithm, which is discussed below, does not warrant tuning the gradient ascent algorithm more than it has been already.

For reference, an outline of the algorithm is given below. It assumes that a set of observations, the mean and variance of the downwelling radiance, and the variance of the measurement noise are available for calculations. The backtracking line search scaling on the step size is set to 0.9 in all the tests.

A value δ was chosen to define the stopping point. This value is a threshold on the amount of change allowable in the likelihood function. Once the change in the likelihood function is small enough, the algorithm stops iterating.

3.2.4 Example Gradient Ascent Performance

Tests were run using this algorithm for a measurement of five wavelengths. The wavelengths were from 8.3 to 8.7 μm in 0.1 μm steps. The true emissivity values were 0.8114, 0.7946, 0.7744,

Algorithm 1: TES Gradient Ascent

Input: Observations, Downwelling Radiance Parameters, Noise Variance

Initialize: Emissivity, Temperature, Step Parameters

while $|L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon}_j, T_j) - L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon}_{j-1}, T_{j-1})| > \delta$ **do**

  Calculate Gradients:

  $\Delta\boldsymbol{\alpha}$ (the ascent direction for $\boldsymbol{\alpha}$, found using (3.2) or (3.5) times (3.9))

  ΔT (the ascent direction for T, found using (3.3) or (3.6)

  Initialize Step Size: $t = 0.001$

  Backtracking Line Search:

  while $L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\alpha}_j + t\Delta\boldsymbol{\alpha}, T_j + t\Delta T) < L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\alpha}_j, T_j)$ **do**

    $t = \beta t$

  Update Values:

  $\boldsymbol{\alpha}_{j+1} = \boldsymbol{\alpha}_j + t\frac{\partial L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon}_j, T_j)}{\partial \boldsymbol{\alpha}_j}$

  $T_{j+1} = T_j + t\frac{\partial L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon}_j, T_j)}{\partial T_j}$

  $\boldsymbol{\epsilon}_{j+1} = \Phi(\boldsymbol{\alpha}_{j+1})$

  $j = j + 1$

0.7865, and 0.7549 at each wavelength (shown in Figure 3.1), respectively. The true temperature was set to 290°K. The downwelling radiance had a mean vector of 547.8675, 572.2968, 596.0602, 561.5744, and 570.2581, all in microflicks. The covariance matrix was 5.6 down the main diagonal with 0.6 on the off diagonal and 0.5 for all other elements. For the noisy case, the noise on the observations was set to $\sigma_n^2 = 1$ for each wavelength. The algorithm was run with δ set to 0. It was initialized with the emissivity as discussed above and with the temperature set to 295°K.

The test was first run with the noise-free data. Figure 3.2 shows the convergence of the of the temperature with respect to iteration. The temperature converges to 289.84°K, very close to the true value of the temperature. It takes about 130,000 iterations to converge, although by 60,000 iterations, it has already reached 290.02°K.

Examining the change in temperature more closely, the first 30 iterations or so can be seen to have a lot of fluctuation, as shown in Figure 3.3. After this point, the convergence is very slow, but smoother.

For reference, the value of t, the step size scaling, that was actually used by the backtracking line search is shown. Although t was initialized at 0.001, Figure 3.4 shows that the step size was reduced to a lower value until the algorithm had almost converged. This value could be used to

speed up convergence of the algorithm by reducing the number of loops in the backtracking line search. This would increase the number of iterations in the algorithm, since the step size increases near the end as the gradient decreases in magnitude.

Figure 3.5 shows the final emissivity value after convergence of the algorithm. The algorithm determines the emissivity to within about 0.01 for all wavelengths.

In order to see how the emissivity changes over iterations, Figure 3.6 shows the emissivity at different numbers of iterations. It is interesting to note that over the first few hundred iterations, the emissivity is simply changing shape, with only small amounts of change in the average value. At around $10,000$ iterations, it begins to change in average value more quickly. Similar to the temperature, as would be expected given the relationship between temperature and emissivity in the model, it begins to slow as it reaches $60,000$ iterations and eventually converges to its final value.

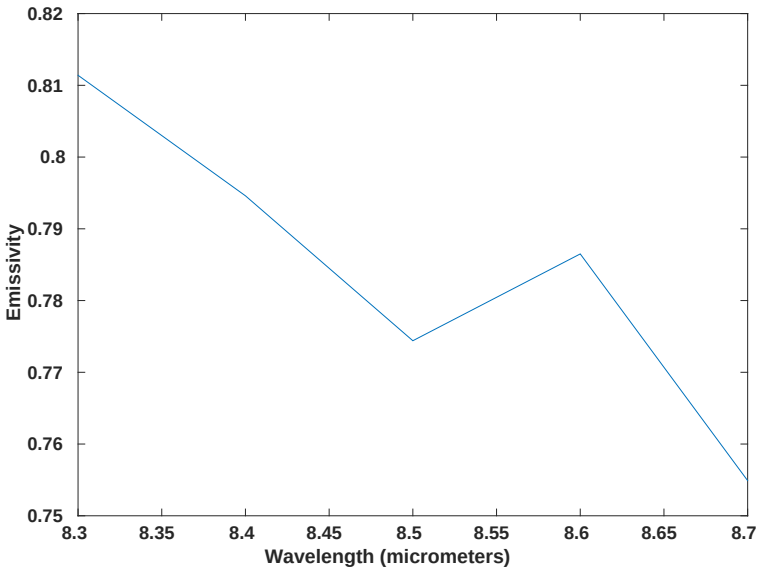

Fig. 3.1: True Emissivity Versus Wavelength For Algorithm Development.

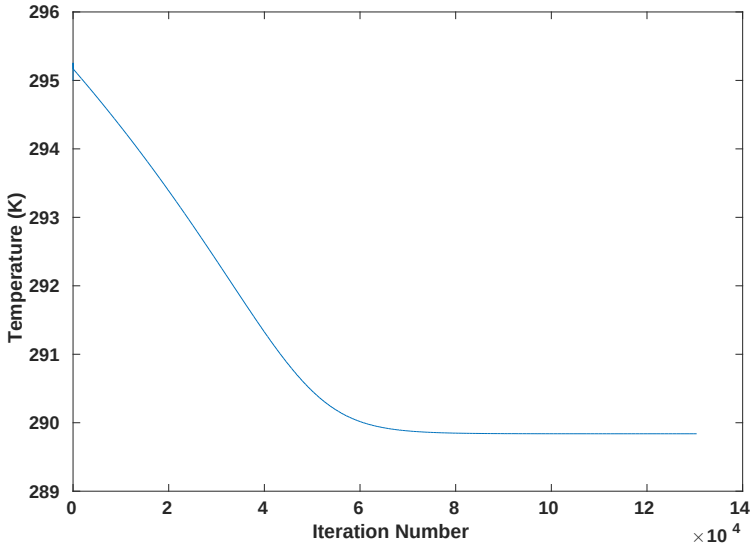

Fig. 3.2: Temperature Versus Iteration Number for Gradient Ascent with $\delta = 0$ and Noise-Free Observations.

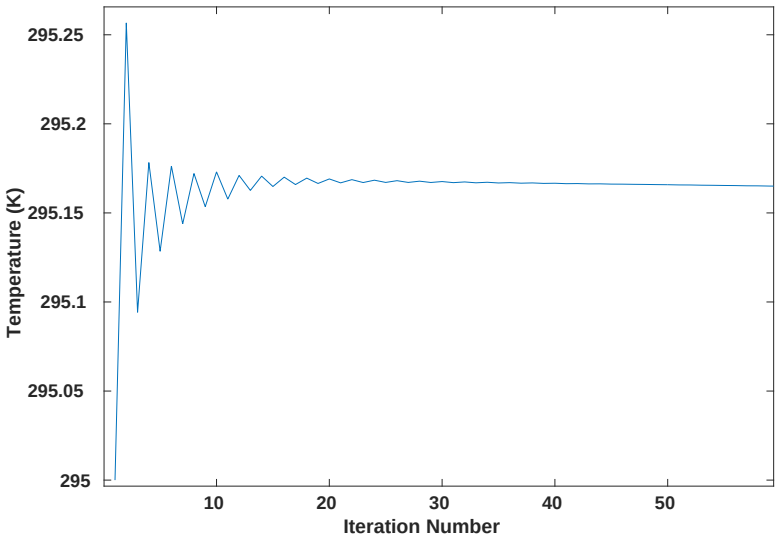

Fig. 3.3: First 60 Iterations of Temperature Versus Iteration Number for Gradient Ascent with $\delta = 0$ and Noise-Free Observations.

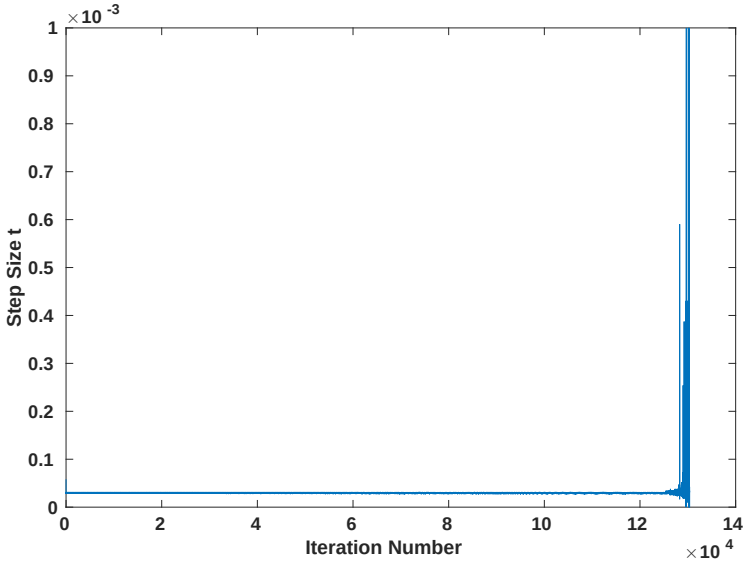

Fig. 3.4: Step Size t Versus Iteration Number with $\delta = 0$ and Noise-Free Observations.

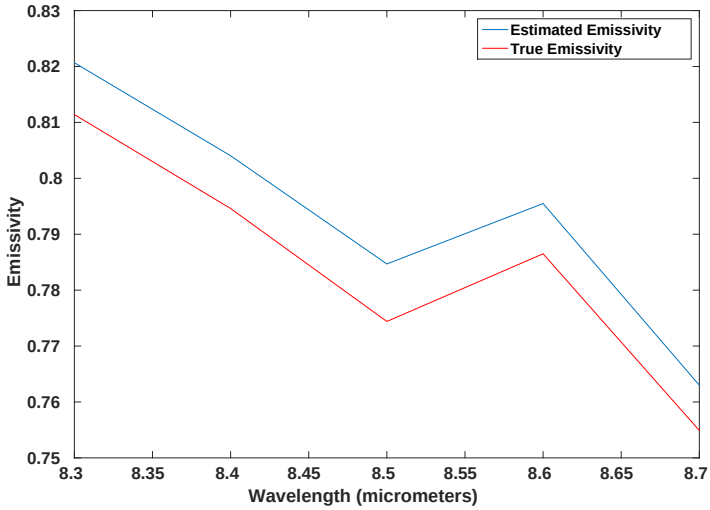

Fig. 3.5: Emissivity Comparison for Gradient Ascent with $\delta = 0$ and Noise-Free Observations.

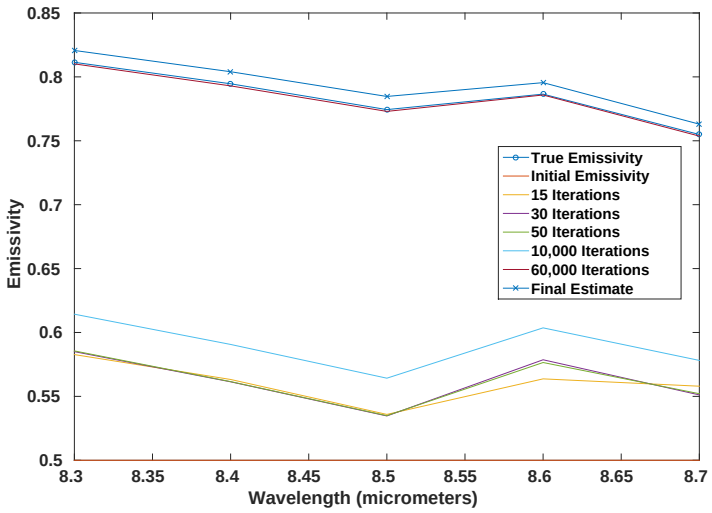

Fig. 3.6: Emissivity Comparison for Gradient Ascent with $\delta = 0$ and Noise-Free Observations.

The likelihood plot, shown in Figure 3.7, is not very informative, as the likelihood starts very small, around $-5,986.2626$ and increases to -38.8820 after only 20 iterations. The likelihood then increases very slowly, which makes sense given the flatness of the objective function. Figure 3.8 shows the first 50 iterations. After 50 iterations, the likelihood has reached -6.6373, but only increases to -6.6373 by the 1000th iteration. At $60,000$ iterations, when the temperature and emissivity finally begin to change more slowly, it has reached 11.5093 and the final value is 11.6340.

The test was also performed with noisy observations. The noisy tests were performed with the same observations, only with noise added on after generating them. This allows for close comparison of the sets of data. Other sets of data were tested with similar performance.

For the noisy case, Figure 3.9 shows the plot of change in temperature versus iteration number. Over $450,000$ iterations were required for convergence. The final temperature value was $290.4879°$K. As in the noise-free case, the algorithm has mostly converged much sooner, reaching $290.5030°$K by $250,000$ iterations. Regardless, this is more than three times as slow as the noise-free case, with larger error in the estimated temperature. Again, as in the noise free case, there is some initial oscillation. Looking more closely at the temperature curve in Figure 3.10, there is drastic oscillation in the initial temperature change. This takes about 30 iterations to occur, suggesting

that there is some initial adjustment by the algorithm.

Looking at the step size in Figure 3.11, the performance is similar to the noise-free algorithm as well. In Figure 3.12, the final emissivity curve compared to the true emissivity curve is shown. Just as the temperature is not as accurate, the emissivity is only about 0.04 of the true emissivity. Again, the likelihood changes very quickly, starting at $-3,866.0119$. The plot of the likelihood is essentially identical to Figure 3.7, except that the final convergence takes longer. The first 50 iterations are shown in Figure 3.13. The likelihood reaches -33.8172 after only 50 iterations, but only attains a final value of -31.3854, which is even flatter than noise-free case. After $100,000$ iterations, the likelihood has already reached -31.5180, and after $250,000$ iterations it is only -31.3854, which suggests that the likelihood has converged by this point. Looking at more significant digits at $250,000$, the likelihood is only changing in the fourth decimal place and these are the same due to rounding. The function is very close to numerically flat at this point. However, the temperature (and thus, emissivity) is still refining at this point, as the temperature accuracy is improved by over half a degree in taking another $200,000$ iterations. As before, the emissivity is plotted for different numbers of iterations. Figure 3.14 shows that, similar to the noise free case, it changes shape at first, then slowly begins to move towards the final value.

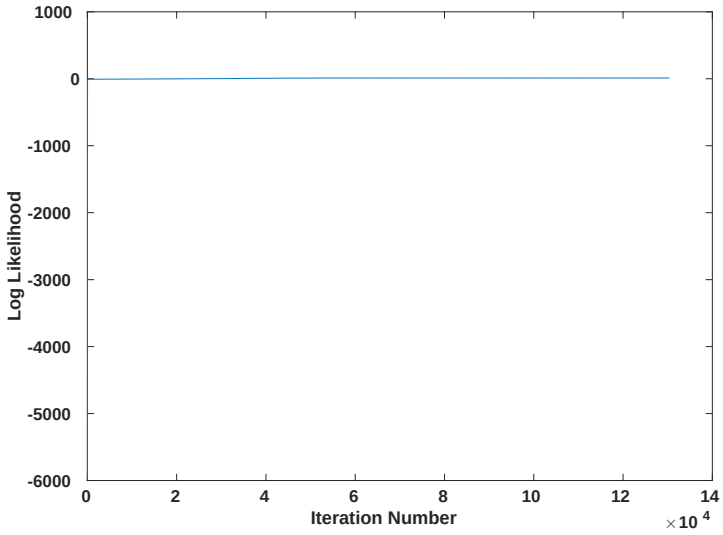

Fig. 3.7: Likelihood Versus Iteration Number for Gradient Ascent with $\delta = 0$ and Noise-Free Observations.

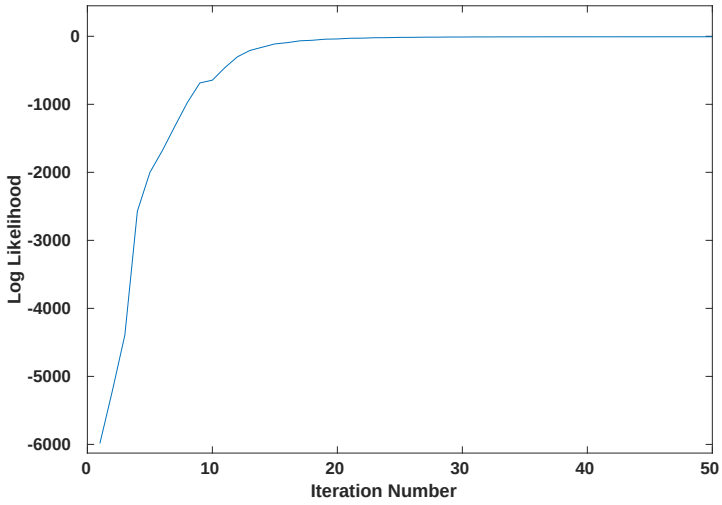

Fig. 3.8: First 50 Iterations of Likelihood Versus Iteration Number for Gradient Ascent with $\delta = 0$ and Noise-Free Observations.

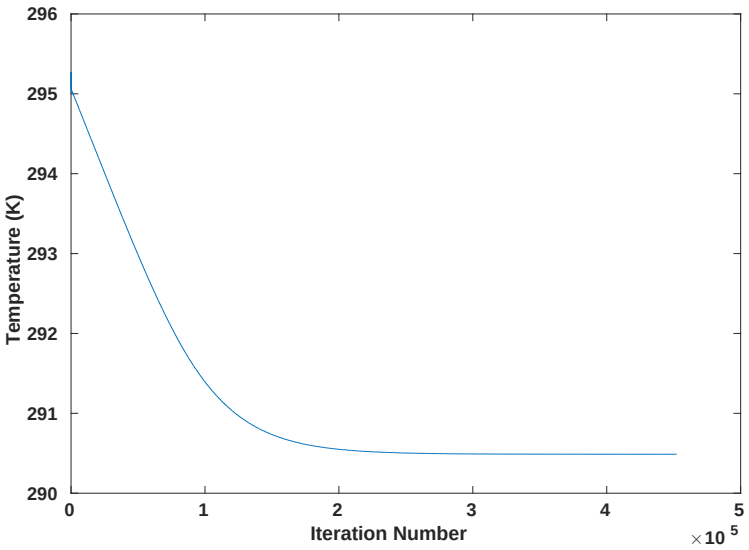

Fig. 3.9: Temperature versus Iteration Number for Gradient Ascent with $\delta = 0$ and Noisy Observations.

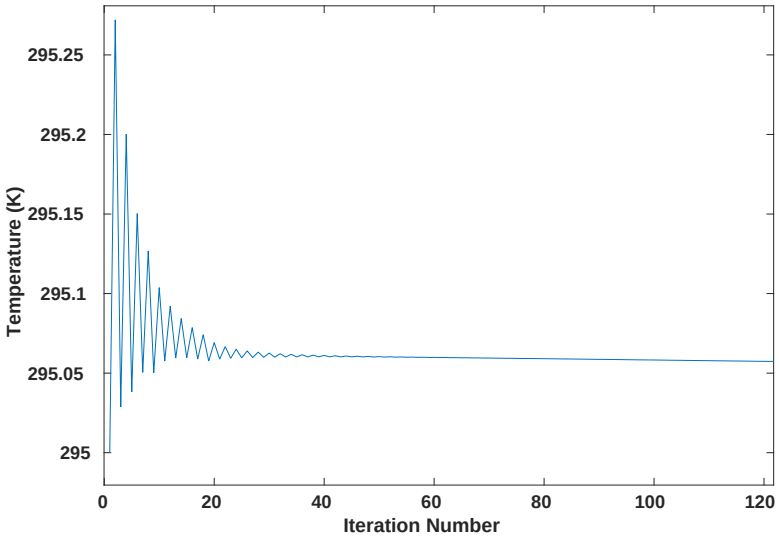

Fig. 3.10: Close up of Temperature versus Iteration Number for Gradient Ascent with Noisy Observations.

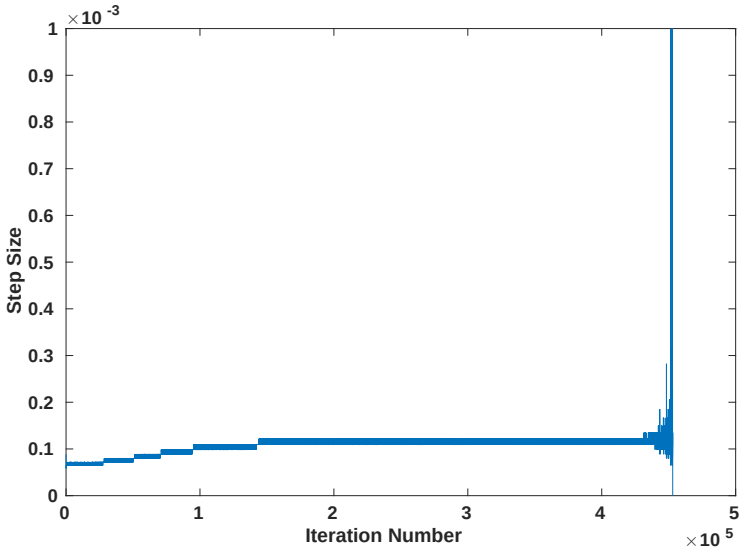

Fig. 3.11: Step Size t Versus Iteration Number with $\delta = 0$ and Noisy Observations.

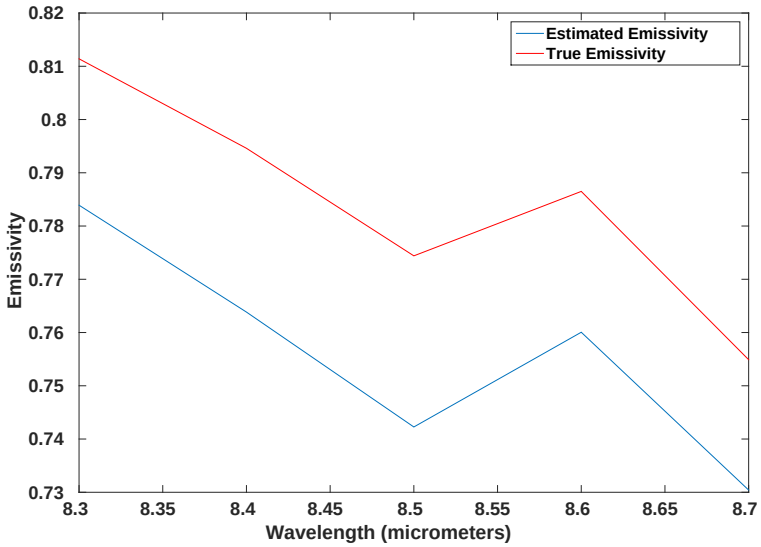

Fig. 3.12: Emissivity Comparison for Gradient Ascent with $\delta = 0$ and Noisy Observations.

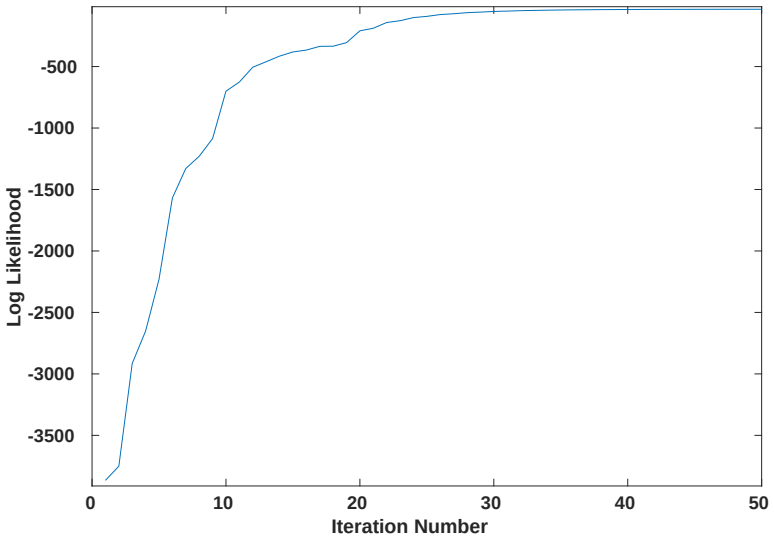

Fig. 3.13: First 50 Iterations of Likelihood Versus Iteration Number for Gradient Ascent with $\delta = 0$ and Noisy Observations.

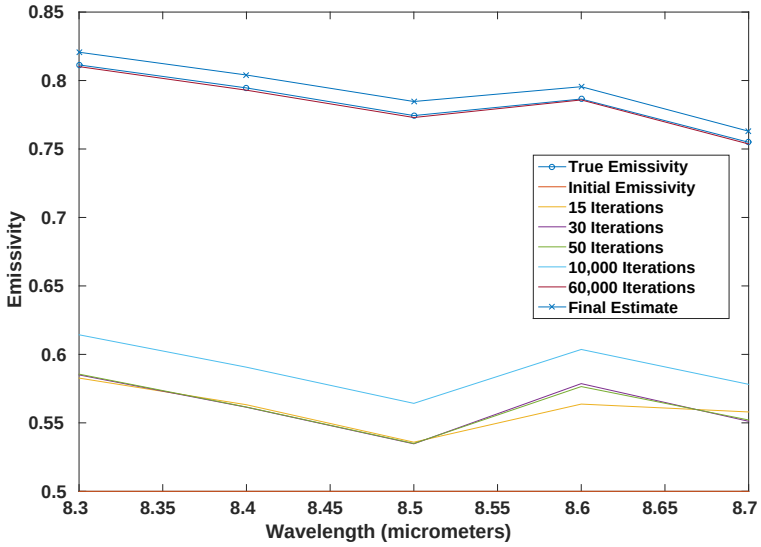

Fig. 3.14: Emissivity Comparison for Gradient Ascent with $\delta = 0$ and Noisy Observations.

3.2.5 Gradient Ascent Algorithm Conclusions

The general performance of the gradient ascent algorithm is good in the sense that the accuracy seems good. However, the number of iterations, especially for noisy observations, is excessive. Since this would be the case for all real data, this algorithm is probably insufficient. The additional problems from the initial temperatures make it a poor choice.

3.3 Newton Step

The Newton step approach was explored next in order to try and improve on the convergence speed. The Newton step makes use of the inverse Hessian of the objective function. By left-multiplying the gradient by the inverse Hessian, the step points more directly towards the maximum value of the function [46]. This should reduce the number of iterations required to achieve the maximum value, hopefully speeding up the algorithm.

3.3.1 Second Partial Derivatives

Since the Hessian must be formed first, the second partial derivative of the gradients (3.2)

and (3.3) (and their noisy counterparts) must be found with respect to the emissivity at each wavelength and the temperature. In addition, the second partial derivative must be found with respect to the parameters α_ℓ to account for the constraints on emissivity. The derivatives are derived in the appendix.

The derivatives of (3.2) and (3.5) are taken first. The derivative with respect to ϵ_ℓ is considered first, with the same emissivity as before, ϵ_m, a different emissivity from before (at a different wavelength), and the derivative with respect to T. For the noise free case, there are three derivatives,

$$\frac{\partial^2 L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon},T)}{\partial \epsilon_\ell^2} = \frac{N}{(1-\epsilon_\ell)^2} - \sum_{i=1}^{N} \frac{2(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^3}\mathbf{e}_\ell^T R^{-1}\mathbf{w}_i + \frac{(y_{i,\ell}-B(T)_\ell)^2 R^{-1}_{\ell,\ell}}{(1-\epsilon_\ell)^4}, \quad (3.10)$$

$$\frac{\partial^2 L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon},T)}{\partial \epsilon_m \partial \epsilon_\ell} = \sum_{i=1}^{N} \frac{(y_{i,\ell}-B(T)_\ell)(y_{i,m}-B(T)_m)R^{-1}_{\ell,m}}{(1-\epsilon_\ell)^2(1-\epsilon_m)^2}, \quad (3.11)$$

$$\frac{\partial^2 L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon},T)}{\partial T \partial \epsilon_\ell} = \sum_{i=1}^{N} \frac{w'_{i,\ell}}{(1-\epsilon_\ell)^2}\mathbf{e}_\ell^T R^{-1}\mathbf{w}_i + \frac{(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^2}\mathbf{e}_\ell^T R^{-1}\Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})}\mathbf{w}'_i. \quad (3.12)$$

For the noisy case, these become

$$\begin{aligned}
\frac{\partial^2 L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon},T)}{\partial \epsilon_\ell^2} = {} & \frac{N}{(1-\epsilon_\ell)^2} + \frac{2N\sigma^4(X^{-T}E_{\ell,\ell}X^{-T})_{\ell,\ell}}{(1-\epsilon_\ell)^6} - \frac{3N\sigma^2 X^{-1}_{\ell,\ell}}{(1-\epsilon_\ell)^4} \\
& - \sum_{i=1}^{N} \frac{2(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^3}\mathbf{e}_\ell^T X^{-1}\mathbf{w}_i \\
& - \sum_{i=1}^{N} \frac{2\sigma^2(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^5}\mathbf{e}_\ell^T X^{-1}E_{\ell,\ell}X^{-1}\mathbf{w}_i + \frac{(y_{i,\ell}-B(T)_\ell)^2 X^{-1}_{\ell,\ell}}{(1-\epsilon_\ell)^4} \\
& + \sum_{i=1}^{N} \frac{3\sigma^2}{(1-\epsilon_\ell)^4}\mathbf{w}_i^T X^{-1}E_{\ell,\ell}X^{-1}\mathbf{w}_i + \frac{2\sigma^2(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^5}\mathbf{e}_\ell^T X^{-1}E_{\ell,\ell}X^{-1}\mathbf{w}_i \\
& - \sum_{i=1}^{N} \frac{4\sigma^4}{(1-\epsilon_\ell)^6}\mathbf{w}_i^T X^{-1}E_{\ell,\ell}X^{-1}E_{\ell,\ell}X^{-1}\mathbf{w}_i,
\end{aligned} \quad (3.13)$$

$$\begin{aligned}\frac{\partial^2 L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon},T)}{\partial\epsilon_m\partial\epsilon_\ell} &= \frac{2N\sigma^4(X^{-T}E_{\ell,\ell}X^{-T})_{m,m}}{(1-\epsilon_\ell)^3(1-\epsilon_m)^3} + \sum_{i=1}^{N}\frac{2\sigma^2(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^2(1-\epsilon_m)^3}\mathbf{e}_\ell^T X^{-1}E_{m,m}X^{-1}\mathbf{w}_i \\ &- \sum_{i=1}^{N}\frac{(y_{i,\ell}-B(T)_\ell)(y_{i,m}-B(T)_m)X^{-1}_{\ell,m}}{(1-\epsilon_\ell)^2(1-\epsilon_m)^2} \\ &\sum_{i=1}^{N}\frac{2\sigma^2(y_{i,m}-B(T)_m)}{(1-\epsilon_\ell)^3(1-\epsilon_m)^2}\mathbf{e}_m^T X^{-1}E_{\ell,\ell}X^{-1}\mathbf{w}_i \\ &- \sum_{i=1}^{N}\frac{4\sigma^4}{(1-\epsilon_\ell)^3(1-\epsilon_m)^3}\mathbf{w}_i^T X^{-1}E_{m,m}X^{-1}E_{\ell,\ell}X^{-1}\mathbf{w}_i,\end{aligned} \tag{3.14}$$

$$\begin{aligned}\frac{\partial^2 L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon},T)}{\partial T\partial\epsilon_\ell} &= \sum_{i=1}^{N}\frac{w'_{i,\ell}}{(1-\epsilon_\ell)^2}\mathbf{e}_\ell^T X^{-1}\mathbf{w}_i + \frac{(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^2}\mathbf{e}_\ell^T X^{-1}\Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})}\mathbf{w}'_i \\ &- \sum_{i=1}^{N}\frac{2\sigma^2}{(1-\epsilon_\ell)^3}\mathbf{w}_i'^T\Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})}X^{-1}E_{\ell,\ell}X^{-1}\mathbf{w}_i.\end{aligned} \tag{3.15}$$

The partial of (3.6) and (3.3) with respect to T must also be found. Technically, these derivatives must also be found with respect to ϵ_ℓ, but they will be identical to (3.12) and (3.15) due to symmetry in the Hessian. The derivative is then

$$\frac{\partial^2 L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon},T)}{\partial T^2} = \sum_{i=1}^{N}\mathbf{w}_i''^T\Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})}R^{-1}\mathbf{w}_i + \mathbf{w}_i'^T\Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})}R^{-1}\Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})}\mathbf{w}'_i, \tag{3.16}$$

where $\mathbf{w}_i''$ is the sum of

$$\frac{-C_1C_2^2e^{\frac{C_2}{\lambda_k T}}}{\lambda_k^7T^4(e^{C_2/(\lambda_k T)}-1)^2}, \tag{3.17}$$

$$\frac{-2C_1C_2e^{\frac{C_2}{\lambda_k T}}}{\lambda_k^6T^3(e^{C_2/(\lambda_k T)}-1)^2}, \tag{3.18}$$

and

$$\frac{C_1C_2^2e^{\frac{2C_2}{\lambda_k T}}}{\lambda_k^7T^4(e^{C_2/(\lambda_k T)}-1)^3}. \tag{3.19}$$

For the noisy case, this is

$$\frac{\partial^2 L(\mathbf{y}_1..\mathbf{y}_N|(\boldsymbol{\epsilon},T)}{\partial T^2} = \sum_{i=1}^{N}\mathbf{w}_i''^T\Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})}R^{-1}\mathbf{w}_i + \mathbf{w}_i'^T\Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})}R^{-1}\Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})}\mathbf{w}'_i. \tag{3.20}$$

As before, the substitution of the α_ℓ terms must now be dealt with in order to deal with the constraints on emissivity. For (3.12) and (3.15)), the α_ℓ term only enters in to the derivative with respect to ϵ_ℓ. Because of this, the same single derivative can be used as before, multiplying these terms by $\Phi(\alpha_\ell)(1-\Phi(\alpha_\ell))$. For (3.11) and (3.14), there are two derivatives with respect to different α_ℓs, so they can simply be multiplied together, yielding $\Phi(\alpha_\ell)(1-\Phi(\alpha_\ell))\Phi(\alpha_m)(1-\Phi(\alpha_m))$. For (3.16) and (3.20), there is no element of α_ℓ, so it is not of concern. The last set of terms, given by (3.10) and (3.13), things become more complicated, since the product rule must be used in order to account for the previous product that was calculated in the first derivative. The result is

$$\frac{\partial^2 L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon},T)}{\partial\epsilon_\ell^2}(\frac{\partial\Phi(\alpha_\ell)}{\partial\alpha_\ell})^2+\frac{\partial L(\mathbf{y}_1..\mathbf{y}_N|(\boldsymbol{\epsilon},T)}{\partial\epsilon_\ell}\frac{\partial^2\Phi(\alpha_\ell}{\partial\alpha_\ell^2}=$$
$$\frac{\partial^2 L(\mathbf{y}_1..\mathbf{y}_N|(\boldsymbol{\epsilon},T)}{\partial\epsilon_\ell^2}(\Phi(\alpha_\ell)(1-\Phi(\alpha_\ell)))^2+\frac{\partial L(\mathbf{y}_1..\mathbf{y}_N|(\boldsymbol{\epsilon},T)}{\partial\epsilon_\ell}(1-\Phi(\alpha_\ell))(\Phi(\alpha_\ell)-2\Phi^2(\alpha_\ell)). \tag{3.21}$$

The derivative is finished by stacking the various second partial derivatives into the Hessian matrix. In calculating these values, it is useful to note that, due to symmetry, only the upper triangle of the Hessian must be calculated.

3.3.2 Newton Step Algorithm

This algorithm works very similarly to the gradient ascent algorithm. The Hessian as well as the gradient are calculated first, then the Newton step is determined. The initialization values are identical. Unfortunately, the temperature initialization also has similar issues. Again, the issues will be addressed in more detail later.

3.3.3 Newton Step Algorithm Performance

The same setup, initial conditions, and dataset are used as for the gradient ascent algorithm. Figure 3.15 shows that that the algorithm converges in only seven iterations. Similar to the gradient ascent algorithm, there is a quick jump in likelihood before it settles at the final value of -6.9447. This is the value reached after about 50 iterations of the gradient ascent algorithm, showing that the Newton step does indeed move more quickly towards the true value. Unfortunately, this final value

Algorithm 2: TES Newton Step

Input: Observations, Downwelling Radiance Parameters, Noise Variance
Initialize: Emissivity, Temperature, Step Parameters
while $|L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon}_j, T_j) - L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon}_{j-1}, T_{j-1})| > \delta$ **do**
  Calculate Gradients and Hessians:
  Gradient with respect to $\boldsymbol{\alpha}$, found using (3.2) or (3.5) times (3.9)
  Gradient with respect to T, found using (3.3) or (3.6)
  Form Hessian using second partial derivatives
  Calculate Ascent Direction:
  $\Delta[\boldsymbol{\alpha}, T]^T = (\nabla^2 L(\mathbf{y}_1..\mathbf{y}_N|(\boldsymbol{\alpha}_j, T_j))^{-1} \nabla L(\mathbf{y}_1..\mathbf{y}_N|(\boldsymbol{\alpha}_j, T_j))$
  Initialize Step Size: $t = 1$
  Backtracking Line Search:
  while $L(\mathbf{y}_1..\mathbf{y}_N|(\boldsymbol{\alpha}_j + t\Delta\boldsymbol{\alpha}, T_j + t\Delta T) < L(\mathbf{y}_1..\mathbf{y}_N|(\boldsymbol{\alpha}_j, T_j)$ **do**
    $t = \beta t$
  Update values:
  $[\boldsymbol{\alpha}_{j+1}, T_{j+1}] = [\boldsymbol{\alpha}_j, T_j] - t\Delta[\boldsymbol{\alpha}, T]$
  $\boldsymbol{\epsilon}_{j+1} = \Phi(\boldsymbol{\alpha}_{j+1})$
  $j = j + 1$

is still much lower than the final value in the gradient ascent case. Convergence is faster, but does not go as far towards the optimal point. This is due to the Hessian approximating the function as quadratic, but the ridge shape not matching this well. This prevents the Newton step from moving along the ridge reliably.

Looking at the temperature plot in Figure 3.16, the temperature is much farther from the true value, converging to $295°$K. The figure shows that the algorithm is not even moving in the right direction, as all the steps in temperature are away from the true value. Careful examination of the algorithm showed that this was, in fact, the direction to step based on the Newton step. As expected, the emissivity is also farther away. Figure 3.17 shows the final emissivity, which is about 0.25 lower than the true value. This is an order of magnitude larger error than for the gradient ascent algorithm. This curiously poor performance is explained when a plot of the step size from the backtracking line search is examined. Shown in Figure 3.18, the first five steps are either a full step or a slightly scaled full step. By the sixth step, the step size becomes zero.

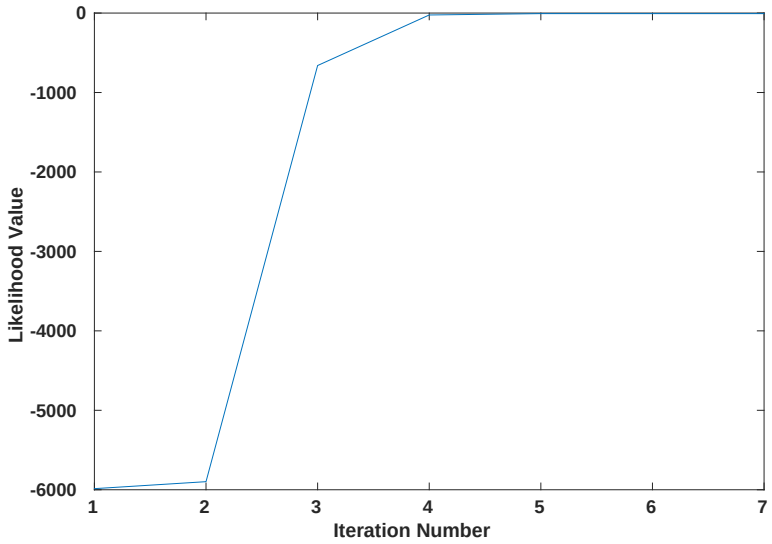

Fig. 3.15: Likelihood Function Value Versus Iteration Number for Newton Step Algorithm with Noise-Free Observations.

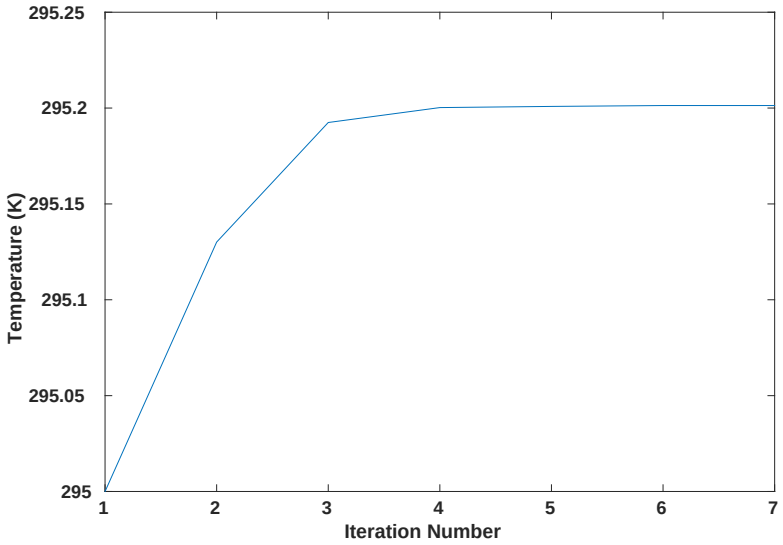

Fig. 3.16: Temperature Versus Iteration Number for Newton Step Algorithm with Noise-Free Observations.

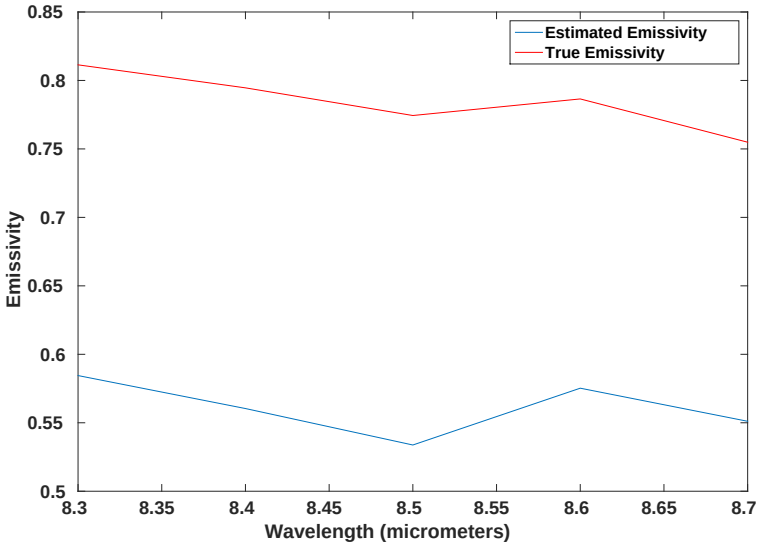

Fig. 3.17: Emissivity Versus Wavelength for Newton Step Algorithm and Noise-Free Observations.

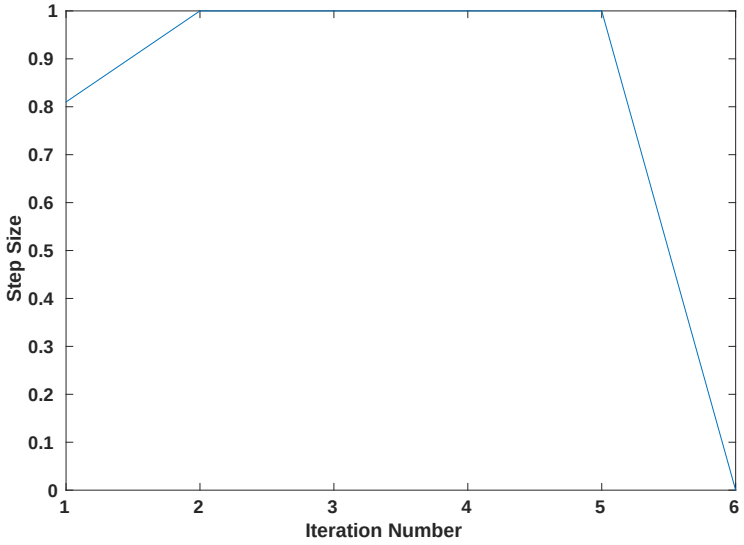

Fig. 3.18: Step Scaling Versus Iteration for Newton Step Algorithm and Noise-Free Observations.

The algorithm is programmed such that the step size is set to zero when the backtracking line search is unable to find a valid step size that increases the likelihood.

It appears that the direction of the Newton step is not necessarily in the direction of increasing likelihood. Efforts to remove the backtracking line search in order to allow further convergence did allow the algorithm to move more. However, it tends to move away from the true temperature, further decreasing the accuracy of the algorithm. Decreasing the initial step size from 1 allowed the algorithm to move slightly towards the true temperature value before eventually converging away from it.

For the noisy case, the performance is almost identical, reaching a likelihood of -33.4558 after only 6 iterations. The temperature and emissivity are similar to the values obtained by the gradient ascent algorithm after about 50 iterations.

3.3.4 Newton Step Algorithm Conclusions

Overall, the Newton step algorithm is extremely disappointing. The convergence is quick, but the algorithm is unable to point the step direction towards the maximum value.

What is of interest is how fast the algorithm is able to reach the emissivity shape. In the gradient ascent algorithm, this was a slow process, taking many iterations. The Newton step is able to get the shape accurately after only a few steps.

3.4 Algorithm Motion on the Likelihood Surface

Given the number of optimization algorithms that have been developed, th process could be continued, trying new algorithms until a preferable one with improved performance was found. The gradient ascent algorithm and its slow convergence could alternatively simply be accepted.

As an alternative, how the two algorithms move across the surface of the likelihood function may be considered. In order to do this, a single wavelength is again examined. This allows a three dimensional plot of the likelihood versus temperature and emissivity to be made. The step locations across this surface can be plotted and see how the algorithm moves in convergence.

3.4.1 Gradient Ascent

The gradient ascent algorithm is considered first. The same setup is used as in the previous chapter, with ten observations that were generated for a single wavelength at 8.3 μm, the temperature set to $290°$K, and the emissivity being 0.8114. The downwelling radiance had a mean value of 547.8675 microflicks and covariance of 5.6. For the noisy case, the noise on the observations was 1.

The likelihood surface was generated and then selected four initial temperatures. To help understand the performance and initial condition problems that were previously seen, one was selected that does not work well, $285°$K, and three that should work better, $290°$K, $295°$K, and $300°$K. The regular step size of the actual algorithm was not used, as plotting several hundred or thousand points could be hard to interpret. Instead, a step of 0.05 in the emissivity direction was taken, scaling the temperature as necessary. While this does not guarantee the the likelihood function will increase, since the gradient ascent algorithm never reached zero step scaling before converging, the movement of the algorithm as it converges should be apparent. For all cases, ten steps were taken.

Figures 3.19 and 3.20 show the results for the noise-free and noisy observations, respectively. There are several observations. First, for the $285°$K case, the steps run to the edge of the surface without stopping. This is the problem that we saw with the algorithm for these temperatures: that the emissivity would run to the boundaries. There is almost no temperature change.

For the other initial temperatures, especially at $295°$K and $300°$K, there are fewer than ten steps. Ten steps are actually present, but the steps start to oscillate back and forth between very close points at some point in the convergence process.

By zooming in on these points, as shown in Figures 3.21 and 3.22, the points can be seen to oscillate around the ridge. That is to say, when approaching the ridge, the gradient tends to bounce back and forth on the ridge (given the step size applied for this test).

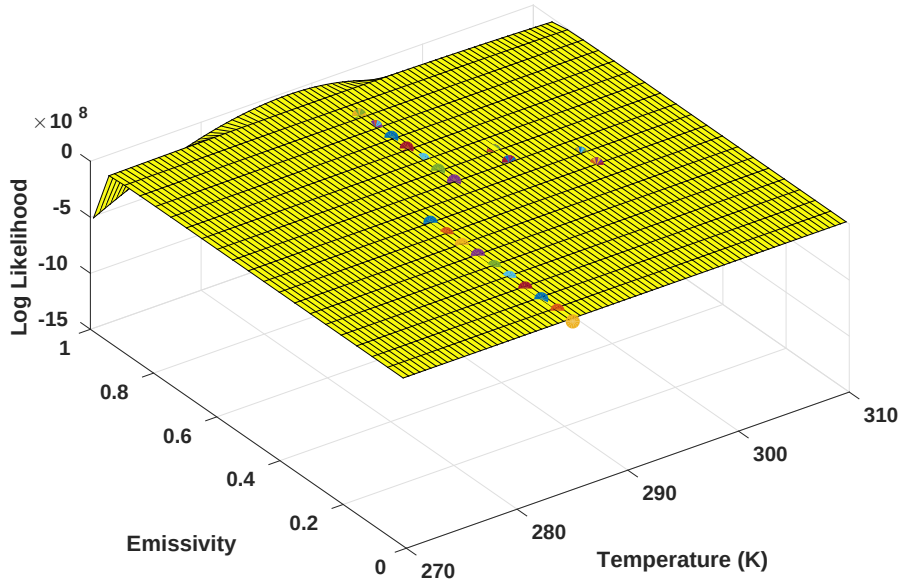

Fig. 3.19: Gradient Ascent Steps Across the Likelihood Surface for Noise-Free Observations and One Wavelength.

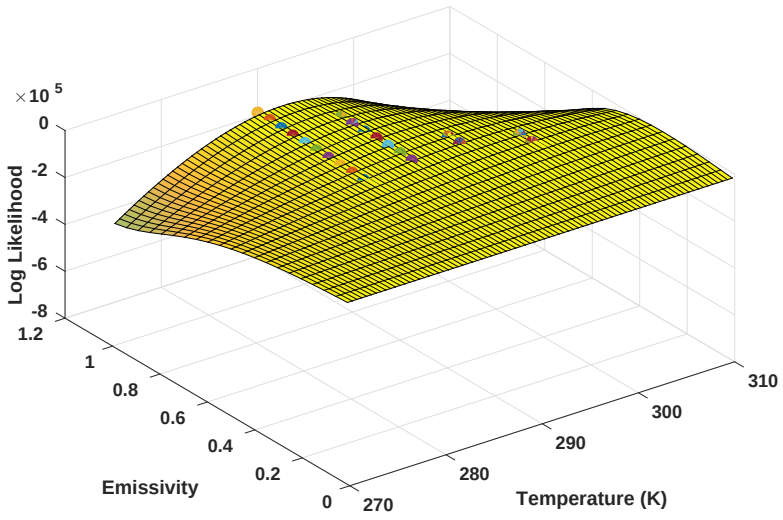

Fig. 3.20: Gradient Ascent Steps Across the Likelihood Surface for Noisy Observations and One Wavelength.

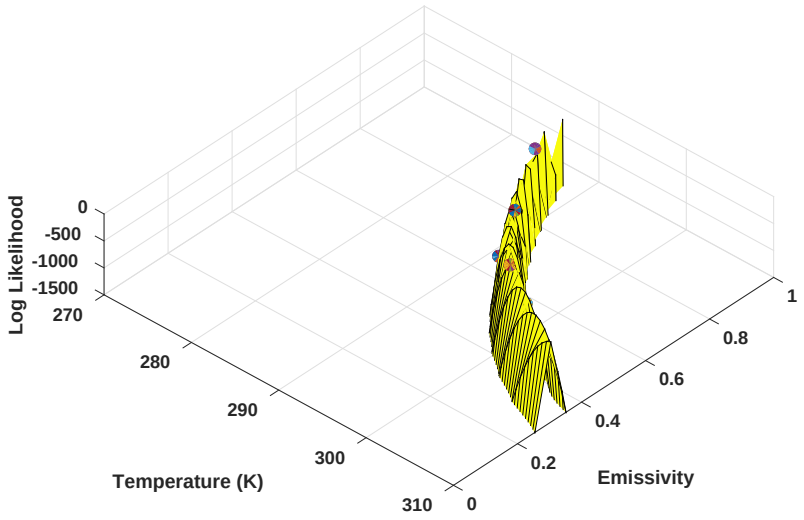

Fig. 3.21: Close-Up of Gradient Ascent Steps Across the Likelihood Surface for Noise-Free Observations and One Wavelength.

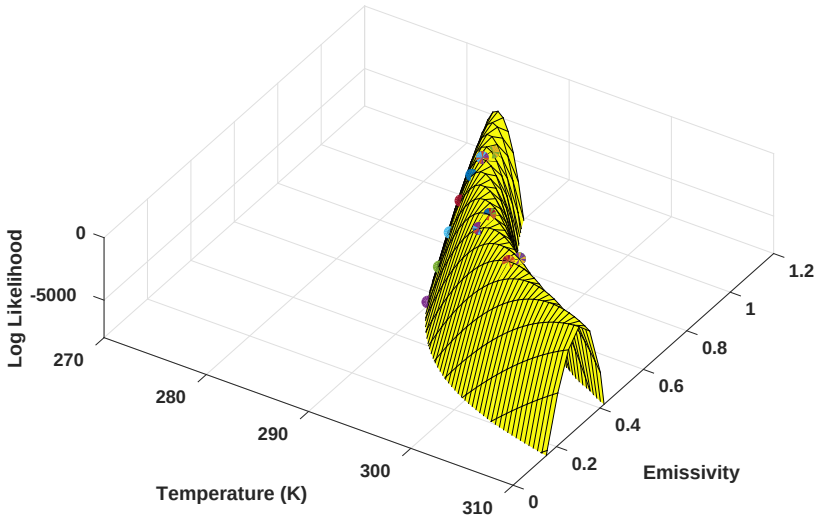

Fig. 3.22: Close-Up of Gradient Ascent Steps Across the Likelihood Surface for Noisy Observations and One Wavelength.

3.4.2 Newton Step

Next, the same test is performed with the Newton step. The same setup and initialization were applied, including the scaling of the step size and removal of the backtracking line search

Figures 3.23 and 3.24 show the Newton step results. Both the 285°K and 290°K cases run off the edge, with the 290°K case jumping in temperature to move off the edge almost in lock step with the 285°K steps. The other two cases seem to oscillate as before, although the oscillations are more distinct, as seen in the series of points in the example shown in Figure 3.25. Reducing the scale, Figure 3.26 shows that the points seem to be migrating slowly along the ridge.

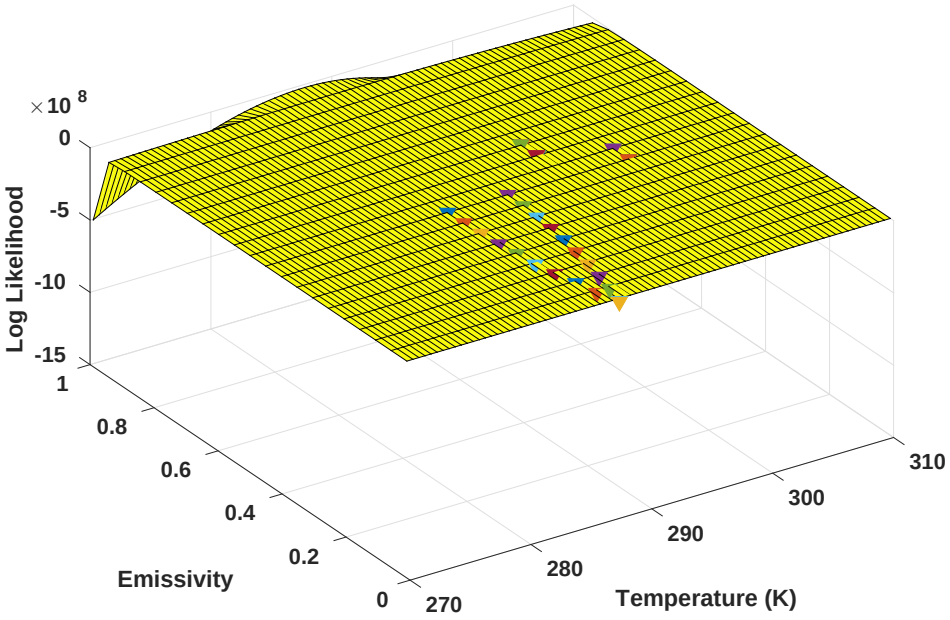

Fig. 3.23: Newton Steps Across the Likelihood Surface for Noise-Free Observations and One Wavelength.

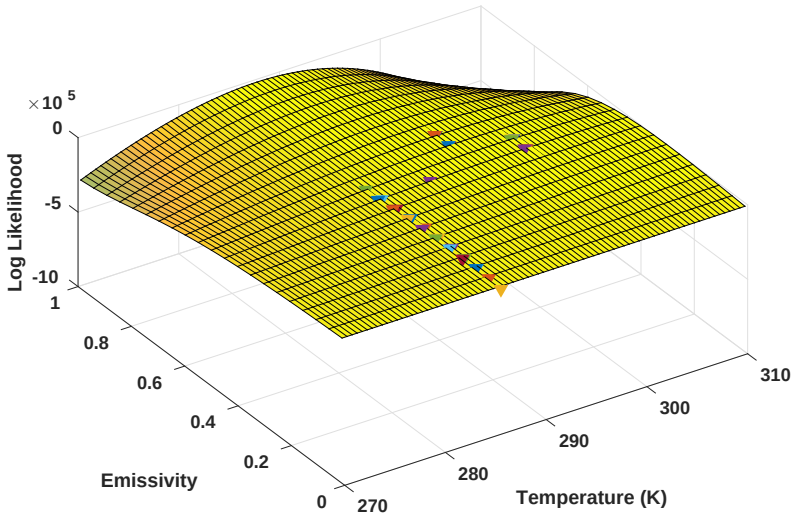

Fig. 3.24: Newton Steps Across the Likelihood Surface for Noisy Observations and One Wavelength.

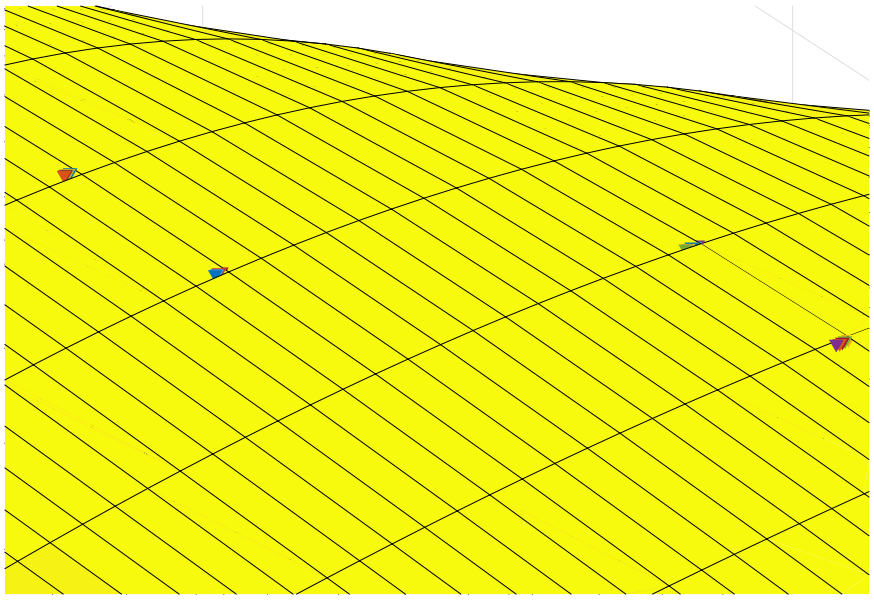

Fig. 3.25: Close-Up of Newton Steps Across the Likelihood Surface for Noisy Observations and One Wavelength.

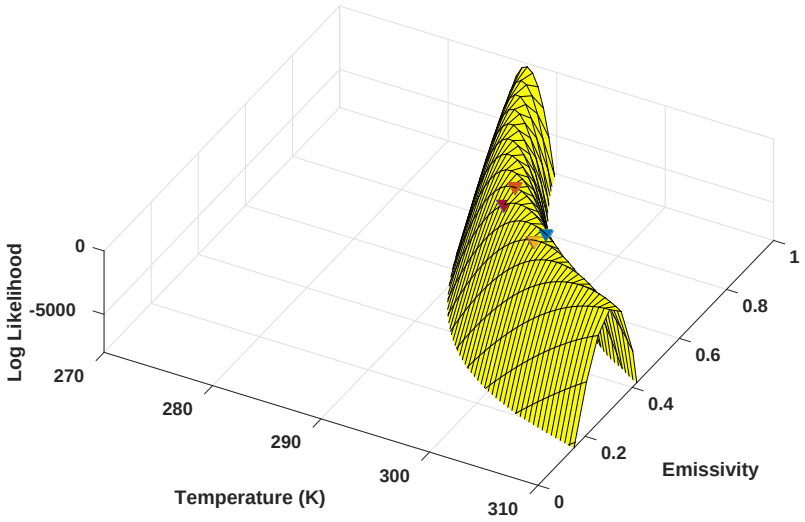

Fig. 3.26: Scale Reduction Close-Up of Newton Steps Across the Likelihood Surface for Noisy Observations and One Wavelength.

3.4.3 Step Performance Discussion

Considering the direction that the Newton steps are moving, that is, that they are going back and forth across the ridge, it is of note that the backtracking line search would have the effect of stopping the Newton steps when they come close to the top of the ridge nearest the starting point in the Newton step direction, as going past the top would reduce the likelihood. So the Newton step algorithm can find the top of the ridge. However, the Hessian is not able to point the step along the ridge.

The gradient ascent algorithm also finds this ridge peak and, as noted before, takes about 50 steps compared to the six or seven for the Newton step (at least in our examples), to reach about the same ridge point. Unlike the Newton step algorithm, this algorithm is able to move along the ridge to the maximum value, although it takes a long time.

Since gradient ascent and Newton step have been tried, it would make sense to try Levenberg-Marquardt. This approach has a tuning parameter that essentially allows us to interpolate between gradient ascent and Newton steps to find a different, and hopefully better, step choice [47].

Picking only between the two, Newton step could be used to get to the ridge, then the gradient ascent could be used to move along it. This would have the benefit of the quick Newton stepping to the ridge, but would still take several hundred thousand iterations to move along the ridge, as the gradient ascent moves slowly in this way.

Since Levenberg-Marquardt allows us to pick steps that are interpolations between the other two, experiments with tuning the interpolation parameter could allow better steps to be found. These steps would expected to be somewhere between the slow gradient convergence and the non-existent Newton convergence, which is not very promising.

Another common approach is the conjugate gradient algorithm. This method avoids calculating the Hessian and so provides a generally faster approach than the Newton step [48]. However, in implementation, this method performed extremely poorly, stalling quickly and not providing significant results.

In consideration of the problems mentioned above, none of these ideas was further developed, in deference to the algorithm described below, which is specifically suited to this optimization problem.

It is possible that these algorithms could, in fat, be tuned to converge, or at least work more quickly than the gradient ascent. However, looking at the ridge shape that the function has, for example, in Figure 2.5, the ridge is slightly curving. The most direct path will require stepping off the ridge across the edge of the ridge and back up to the maximum. Once off the ridge, the gradient ascent or Newton step will focus on going back up the ridge, as it is the steepest direction by far.

Intuitively, the maximum must be somewhere on the ridge. Since the top of the ridge can be found quickly (using Newton step), an algorithm that moves quickly along the ridge would be useful, with no tuning necessary. The optimal step direction could be found directly, at least inasmuch as stepping along the ridge direction is optimal.

This has an obvious advantage over the Newton step quadratic approximation, since it considers the true shape of the ridge.

3.5 Ridge Optimization

In order to move along the ridge, the direction of the ridge must be known. In order to develop some theoretical understanding, a function of two variables must be used, this time with a completely flat ridge. This function will have no unique maximum. A generalized equation for such a function is

$$f(x,y) = (ax - by)^2 = -a^2x^2 - b^2y^2 + 2aby. \tag{3.22}$$

This function has a ridge along the line $ax = by$. The Hessian of this function is the matrix

$$H = \begin{bmatrix} -2a^2 & 2ab \\ 2ab & -2b^2 \end{bmatrix}. \tag{3.23}$$

The eigenvectors of H are $[b/a, 1]$ and $[-a/b, 1]$ with associated eigenvalues 0 and $-2a^2 - 2b^2$, respectively.

While this does not tell much in this form, an example with real numbers is insightful. As an example, consider $a = b = 1$. This gives (normalized) eigenvectors $[\sqrt{2}, \sqrt{2}]$ and $[\sqrt{2}, -\sqrt{2}]$ with eigenvalues 0 and -4, respectively. Figure 3.27 shows this ridge along with the eigenvectors plotted along the surface, centered at zero. Note that the zero eigenvalue has an associated eigenvector that points along the ridge (the ridge is on $x = y$). Any other ridge with different values of a and b is equivalent within a rotation of coordinates, which can easily be accomplished by a unitary transform.

The Hessian describes the local quadratic nature of the function and the smallest magnitude eigenvalue indicates the eigenvector which points in the direction of the least amount of curvature. This was verified in the simple example above. In actual implementation, the smallest positive (including zero as positive) eigenvalue yielded the correct eigenvector, while selecting the smallest magnitude eigenvalue sometimes caused the algorithm to step to the edge, off of the ridge.

For the likelihood function, it was noted in chapter 2 that the ridge was very flat on the top, almost to within numerical precision. Further, the sides of the ridge were very steep. As such, if the Hessian is found at a point on the ridge, the eigenvector of that Hessian with the smallest magnitude eigenvalue should point along the ridge, as any other direction will have a large change in curvature.

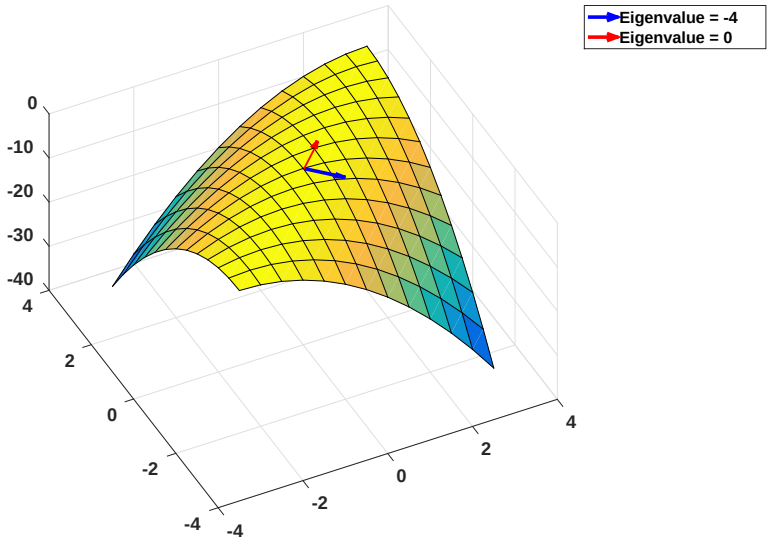

Fig. 3.27: Ridge Function with Eigenvectors.

A step can be taken in this direction and move along the ridge. In the case where the ridge is curved, this step may lead off the ridge, but the Newton step, as previously seen, can move back onto the ridge.

3.6 Proposed Approach

As a method to optimize more quickly than gradient ascent, the eigenvector with the smallest positive eigenvalue of the Hessian at the current ridge point to move in the direction of the ridge. Since the direction of the ridge is arbitrary, the algorithm will have to account for checking both directions along the vector to see which one provides the greatest increase in the objective function. There is some chance of stepping off the ridge, depending on how straight it is, so before checking this value, some approach will need to be used to step back to the top of the ridge. This combined step: stepping in the eigenvector direction and stepping back to the top of the ridge, then checking which sign to put on that step, we call the eigenstep. Generally speaking, any approach could be considered to reach the top of the ridge equivalent. However, use the specific approach of taking a Newton step will be used (due to it stepping more quickly to the ridge). Other methods could be substituted if they had desirable characteristics, but the general approach of this eigenstep and its

uniqueness in compared to other optimization approaches would remain.

Since eigenvectors are typically normalized to unity scaling is somewhat arbitrary in relation to our objective function, so some kind of step size must be determined and potentially reduced over time, through a backtracking search or other means, to allow convergence with greater precision.

This method is dependent on starting at the top of the ridge in the first place. As such, an approach must be found to reach the ridge. From the tests above, the Newton step proved to be effective at quickly stepping to the top of the ridge. The fact that it does not converge along the ridge would then be offset by stepping in the ridge (eigenvector) direction.

Making use of the Newton step algorithm to reach the ridge would have the additional difficulty of requiring a valid initial temperature, so as part of the algorithm development, this issue will have to be resolved.

3.6.1 Solving Initial Temperature Performance Issues

Initializing the Newton (or gradient ascent) algorithm requires a temperature value. As has been seen, certain initial temperature values result in ascent paths that do not reach the ridge. For whatever reason, the paths along the surface of the function seek to reach a value on the ridge that is outside of the constraints. The poor direction of these steps is most likely caused by the flatness of the objective function.

Interestingly, the temperature curves at the initialized emissivities is actually well behaved. Figure 3.28 shows a typical curve at the initial emissivity of 0.5. Regardless of the number of emissivities or the presence or absence of noise, these curves have a similar shape. The distinct maximum and parabolic shape of these curves lend themselves to gradient ascent techniques. While the maximum of this curve is not the true answer, tests show that a gradient ascent algorithm applied only to the temperature is able to quickly converge to the maximum value of the temperature, which can then be used as an initial temperature for the overall TES algorithm. Since temperature value is on the peak of the ridge, at least in relation to the initial emissivities, it is a valid starting point for the algorithm. Tuning of the algorithm increased the speed to be only a few steps. The algorithm is a variation on the previous gradient ascent algorithm, but for clarity, an algorithm outline is provided

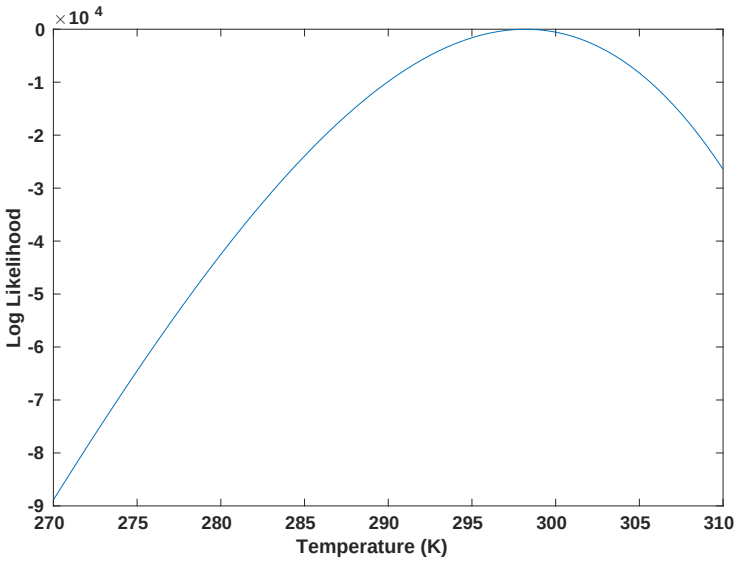

Fig. 3.28: Temperature Versus Likelihood Plot.

to make clear the parameters. Of key importance in the algorithm was the tuning of the initial

Algorithm 3: Temperature Gradient Ascent

Input: Observations, Downwelling Radiance Parameters, Noise Variance
Initialize: Emissivity, Temperature, Step Parameters
while $|T_j - T_{j-1}| > 0.01$ **do**
  Calculate Gradient:
  ΔT (the ascent direction for T, found using (3.3) or (3.6)
  Initialize Step Size: $t = 5 \times 10^{-4}$ (noise-free) or $t = 1 \times 10^{-3}$ (noisy)
  Backtracking Line Search:
  while $L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\alpha}_j + t\Delta\boldsymbol{\alpha}, T_j + t\Delta T) < L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\alpha}_j, T_j)$ **do**
    $t = \beta t$
  Update Values:
  $T_{j+1} = T_j + t\frac{\partial L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon}_j, T_j)}{\partial T_j}$
  $j = j + 1$

step size. Different values worked better for both the noise-free and noisy cases. This allowed the algorithm to converge quickly.

It is also important to note that this algorithm only changes the temperature value. All other initial conditions remain the same.

If the algorithm is run until the backtracking line search cannot find a valid step, then this allows it to take about $20 - 25$ steps. Smaller or larger initial step sizes can increase this by an order of magnitude or more. The actual stopping condition was based on convergence in temperature and allowed for much faster convergence. Figures 3.29 and 3.30 show typical plots for the typical five wavelength case used in this chapter with noise-free and noisy observations. The initial temperature for this part of the algorithm was set to $270°$K, which is quite far from the true value.

The figures show convergence within ten iterations. For both cases, the temperature converged to around $297°$K. Generally speaking, picking a closer initial value for T does not reduce the number of iterations significantly. However, the reach of this algorithm is quite far and it will find the maximum temperature quickly even from several hundred degrees away. It is recommended that initialization of the temperature for this algorithm be based on any available atmospheric temperature data, which should allow for quick convergence of the algorithm.

After running this algorithm, a temperature is reached that will allow a Newton step algorithm to find the ridge of the objective function.

3.6.2 Finding the Ridge

In order to find the ridge, the Newton step algorithm as defined previously will be used. The stopping condition was set to a change in the likelihood function of 10. While more precision could perhaps be obtained, these Newton steps yield almost no change in the temperature overall and mostly tune the emissivity to its correct shape. Multiple steps were not found to change the emissivity significantly. Since only a few steps are being taken, no backtracking line search was used and, in the example, only two (noisy) or three (noise-free) full Newton steps were taken. The main result from these steps is the change in emissivity, with the results being shown in Figures 3.31 and 3.32.

While the results are far from the true value, which is expected given that the goal is finding the top of the ridge, the proper shape of the emissivity curve has been found, which indicates that the point if convergence is at or near the top of the ridge.

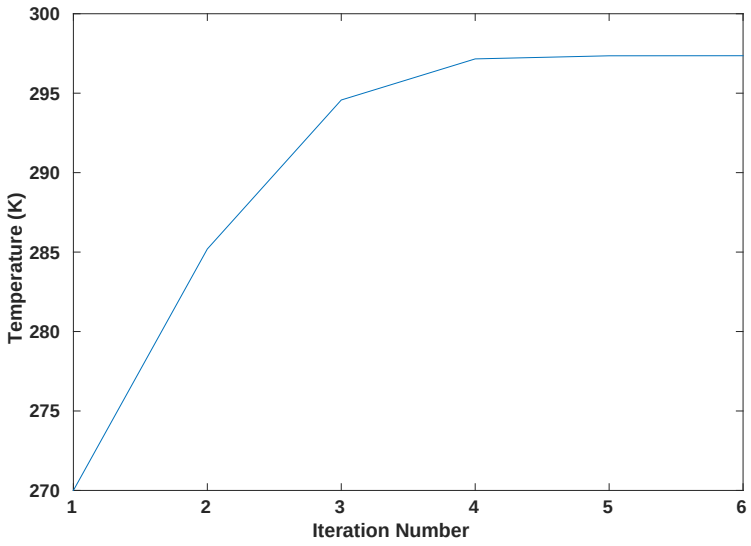

Fig. 3.29: Temperature Gradient Ascent Algorithm Convergence with Noise-Free Observations.

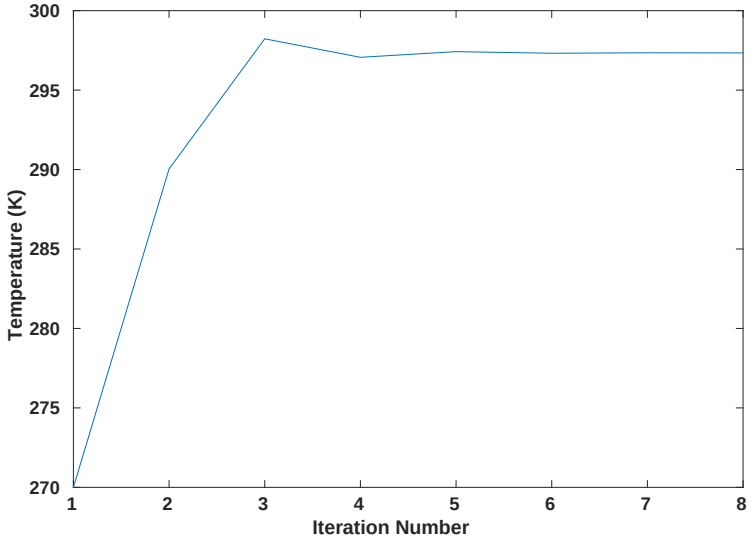

Fig. 3.30: Temperature Gradient Ascent Algorithm Convergence with Noisy Observations.

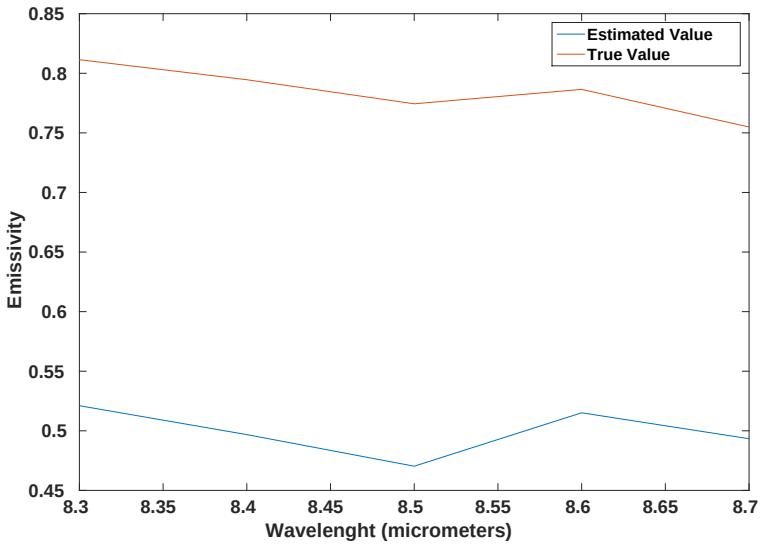

Fig. 3.31: Estimated Emissivity Versus Wavelength After Initial Newton Stepping with Noise-Free Observations.

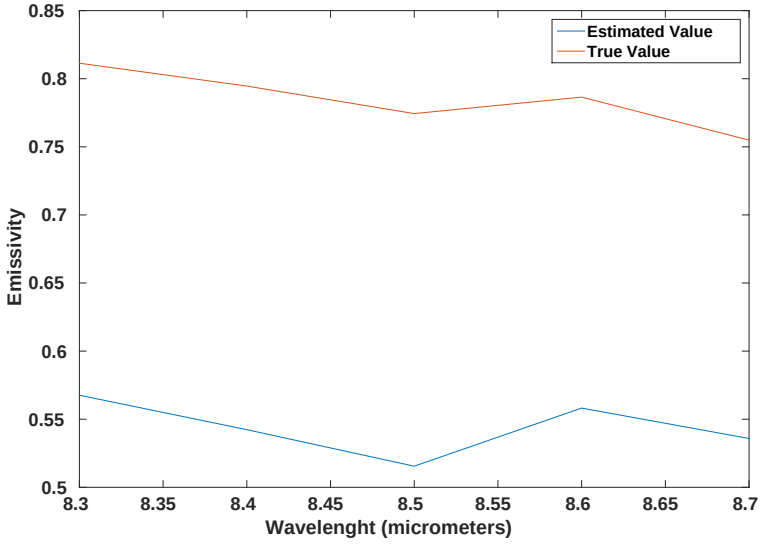

Fig. 3.32: Estimated Emissivity Versus Wavelength After Initial Newton Stepping with Noisy Observations.

3.6.3 Eigenstep Algorithm

Having climbed to the top of the ridge, the eigenstep approach can now be employed. First, find the Hessian at the current point. Compute the eigen decomposition and select the eigenvector corresponding to the smallest positive eigenvalue. Then select some step size and step the in the eigen direction scaled by that step size. From that point, apply the Newton step to ensure the the algorithm is back on top of the ridge. At this point, evaluate the likelihood function for likelihood value in the first direction.

Next, take a step in the opposite direction of the eigenstep and Newton step back to the top of the ridge. Evaluate the likelihood here and compare this to the first likelihood value. Select the step direction corresponding to the larger likelihood value and accept that point as the new temperature and emissivity.

Then, repeat the process, finding the Hessian at the new point and so forth.

3.6.4 Stopping Conditions

In order to get the algorithm to terminate, some kind of stopping criterion must be chosen. In the gradient ascent and Newton steps, the backtracking line search eventually reduces the step size to zero, allowing the algorithm to converge. As shown in the Newton step case, this can cause premature convergence.

As a simple test, using the same setup as other tests, the temperature gradient ascent was run starting from $295°$K, the ridge was found using Newton step, and two hundred eigensteps were taken. The test was run with a fixed step size of 0.1. The results for the noise-free observations are shown in Figure 3.33. The results are very similar to the noisy version. Note in the figure that the algorithm has roughly converged by about 80 iterations. This is much better speed than any algorithm that has been tried and includes the setup iterations for the temperature gradient ascent and Newton step initialization. The likelihood achieved is 11.504, which is a little more than 0.1 less than the value achieved by the gradient ascent algorithm. This suggests that refinement of the step size will be necessary as the optimal point is approached.

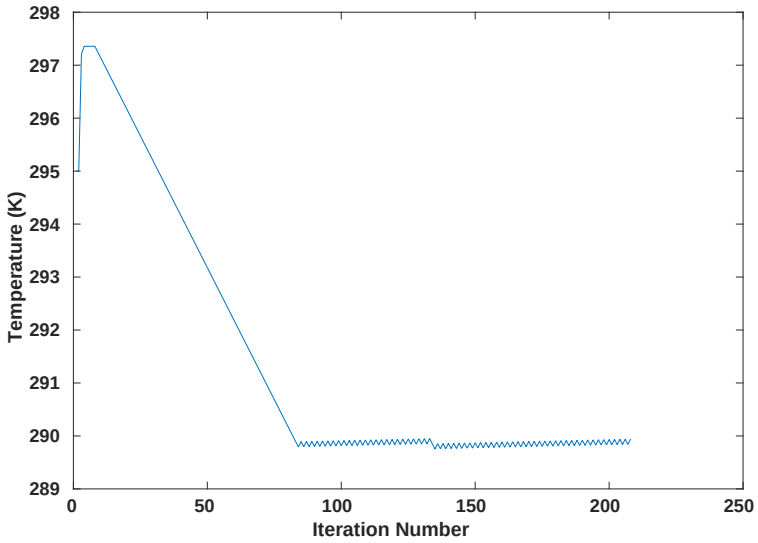

Fig. 3.33: Temperature Versus Iterations for Eigenstep Approach with 0.1 Step Size and a Fixed 200 Iterations.

The oscillation seen after about 80 iterations appears to be a feature of this approach and is present in every test. This means that a way will be needed to refine the step size over time.

Similar tests with larger step sizes were conducted and converged to this oscillating phase much faster. As would be expected, the oscillation was larger. Additionally, there was a drift in the oscillations, suggesting there is some underlying convergence still taking place, even after 200 iterations.

These results present several issues. First, as identified, the step size needs to be refined as the algorithm progresses. Due to the necessity of determining direction by calculating both directions, implementing a backtracking line search is infeasible. Second, a stopping condition is needed, but there is oscillation in the temperature. This will additionally prevent the likelihood function from ever ceasing to change.

As a result, a way to reduce step size naturally as the algorithm converges is needed or else the algorithm will suffer the speed loss due to using a small step size to begin with. Also, a condition to identify the stopping point of the algorithm is needed.

The proposed solution combines the two. A set of step sizes is chosen, with the last one being chosen to provide some overall level of accuracy. The algorithm will start with the largest value.

When the eigenstep reverses direction in temperature, it implies that we have passed the maximum value, at least with regards to temperature. At this point, the algorithm switches to the next smallest step size. This reduction in step size will search back over the region of the function that has just been passed. The temperature is the focal point due to the surface being less flat along the temperature direction. This process will be repeated every time the algorithm switches directions until the last step size is used and the algorithm will have to be considered to have converged.

This approach has the advantage of allowing the accuracy of convergence to be specified, within the accuracy of the objective function overall. Additionally, it provides a natural stopping condition that does not have to deal with the impact of the oscillations.

3.7 Proposed Algorithm

The TES Eigenstep Algorithm, outlined below, is initialized in the same manner as the Newton step and Gradient Ascent algorithms. It is important to note that the use of the temperature gradient ascent algorithm renders the choice of initial temperature moot, except in that it changes convergence time of that portion of the algorithm. Whatever starting point, it will still converge to approximately the same value. For clarity, the gradients and Hessian include the sigmoid function to account for the constraints on emissivity.

3.7.1 Performance

The algorithm was used with the same set of data as before. For the noise-free data, we start with the plot of temperature versus iteration number, shown in Figure 3.34. The temperature converges to $289.84°$K, which is not only close to the true value, but identical (to the second decimal) to the gradient ascent algorithm. In contrast to gradient ascent, this algorithm converges in only 59 iterations. Further, while there is only a little convergence after about 50 iterations, the cost of the extra 9 iterations is low.

Although it is not entirely clear from the figure, the first four steps are the temperature gradient ascent. The next two steps are the Newton steps onto the ridge. The rest of the steps are all eigensteps. When the algorithm changes directions, large spikes are seen, after which the step size changes and the algorithm continues on.

Algorithm 4: TES Eigenstep Algorithm

Input: Observations, Downwelling Radiance Parameters, Noise Variance
Initialize: Emissivity, Temperature, Step Parameters
Adjust the initial temperature: Run the Temperature Gradient Ascent Algorithm
Step to the top of the ridge: Use the TES Newton Step Algorithm with $\delta = 10$
Step along the ridge:
Initialize $k = 0$ and $j = 0$
while $k <$ *Number of step sizes* **do**
    Calculate the Hessian
    Take eigen decomposition of the Hessian
    Select the eigenvector associated with the smallest positive eigenvalue of the Hessian
    Set the step size to the kth value in the list of step sizes
    Scale the eigenvector so that the temperature step is the magnitude of the step size
    Take two separate steps, one in the direction of the scaled eigenvector and one in the opposite direction
    For both steps, take one step using the Newton Step Algorithm
    Evaluate the likelihood function for both results
    Select the step with the higher likelihood
    Update the values based on that step:
    Update values:
    $[\boldsymbol{\alpha}_{j+1}, T_{j+1}] = [\boldsymbol{\alpha}_j, T_j] - t\Delta[\boldsymbol{\alpha}, T]$
    $\boldsymbol{\epsilon}_{j+1} = \Phi(\boldsymbol{\alpha}_{j+1})$
    $j = j + 1$
    if *Direction of the selected temperature step reversed* **then**
        $k = k + 1$ (This moves to the next step size)

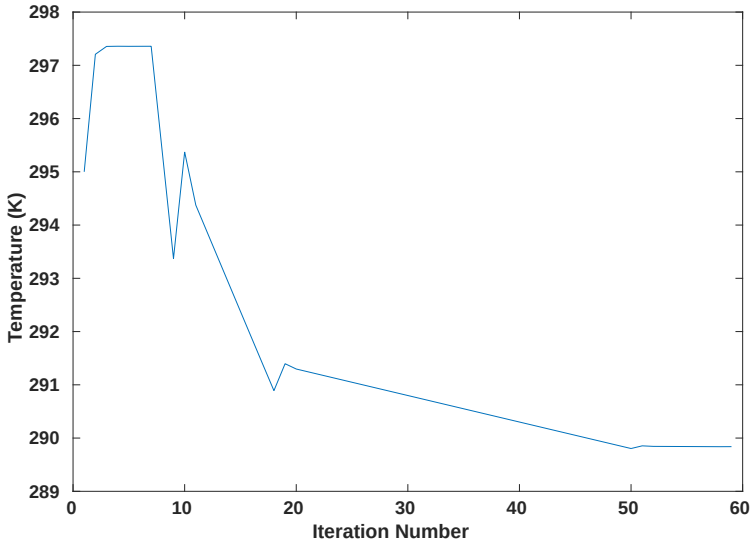

Fig. 3.34: Temperature Versus Iteration Number for Eigenstep Algorithm with Noise-Free Observations.

Figure 3.35 shows the final estimated emissivity value after convergence of the algorithm. The algorithm gets within about 0.01 for all wavelengths which, again, is almost identical to the gradient ascent algorithm.

The likelihood is shown in Figure 3.36. The final log likelihood is 11.634, which is the same as the gradient ascent algorithm. After 50 iterations, it has already reached 11.6167. The convergence of the temperature gradient ascent algorithm can be easily seen where it stops at the first shoulder of the curve. The second shoulder is where the Newton step onto the ridge finishes. As before, the majority of the iterations result in only a small increase in likelihood, while the temperature and emissivity continue to change.

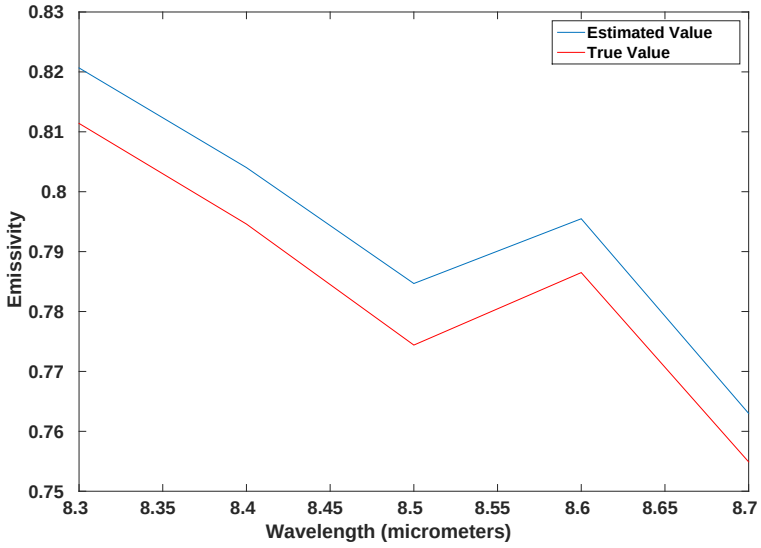

Fig. 3.35: Emissivity Comparison for Eigenstep Algorithm with Noise-Free Observations.

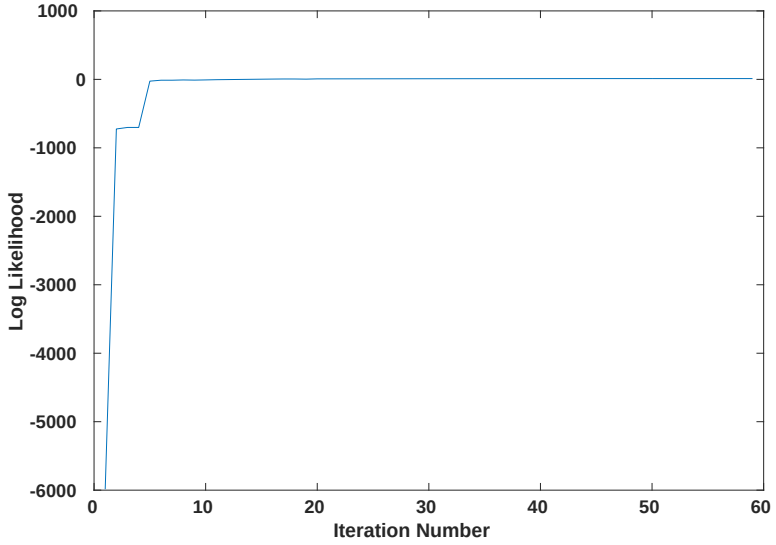

Fig. 3.36: Likelihood Versus Iteration for Eigenstep Algorithm with Noise-Free Observations.

For the noisy case, Figure 3.37 shows the plot of change in temperature versus iteration number. As in the gradient ascent, more iterations are needed, although it is only a few more, with the eigenstep converging in 64 iterations. The final temperature value was $290.4882°$K, which is within 0.002 of the gradient ascent algorithm. As in the noise-free case, the algorithm has mostly converged sooner, reaching about $290.4882°$K after 40 iterations.

For reference, the temperature gradient converges after 7 steps and the Newton step to the top of the ridge only takes 2 steps.

Figure 3.38 shows the final emissivity curve compared to the true emissivity curve. Again, as in the gradient ascent case, all the values are within 0.04 of the true emissivity.

The likelihood is shown in Figure 3.39. The final log likelihood is -31.3854, which is again the same as the gradient ascent algorithm. After 40 iterations, the likelihood has already reached -31.3855. The figure shows essentially the same kind of performance as in the noise-free case.

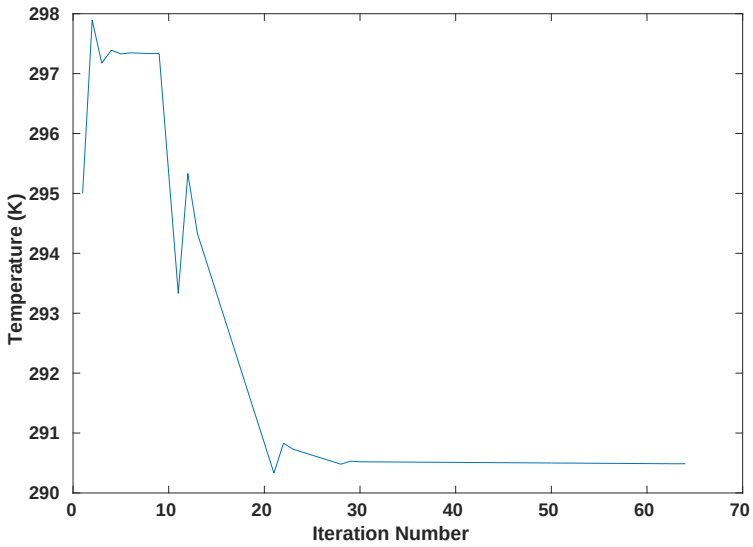

Fig. 3.37: Temperature Versus Iteration Number for Eigenstep Algorithm with Noisy Observations.

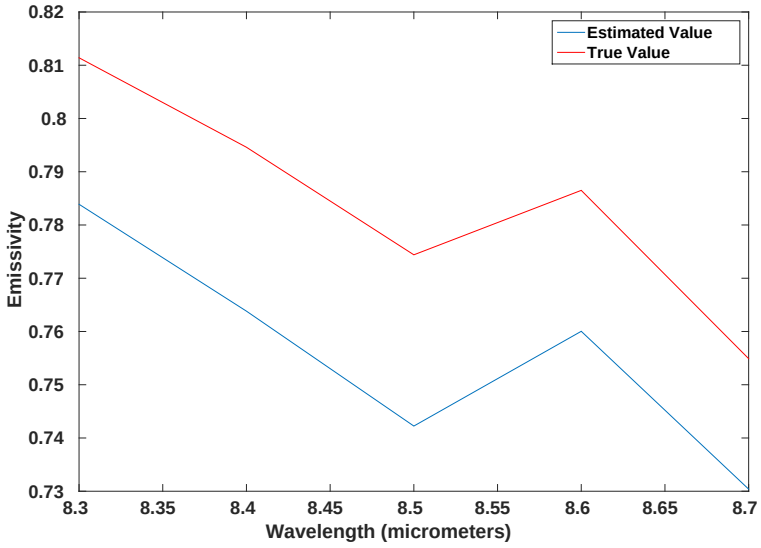

Fig. 3.38: Emissivity Comparison for Eigenstep Algorithm with Noisy Observations.

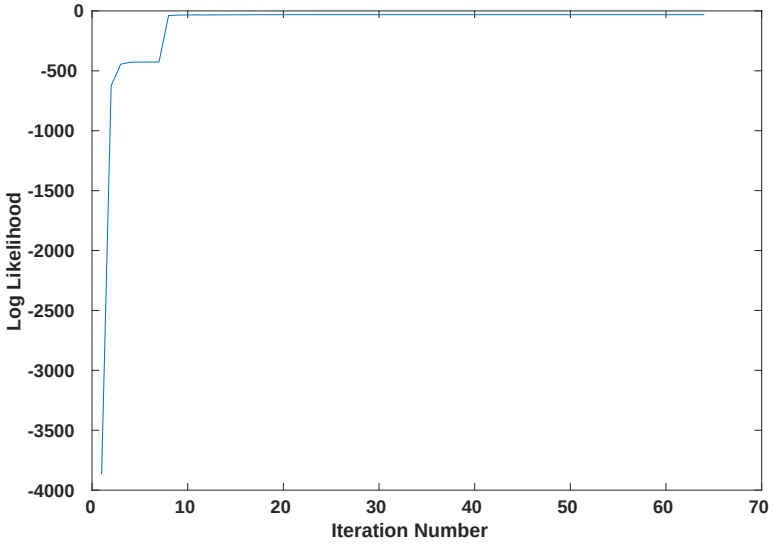

Fig. 3.39: Likelihood Versus Iteration for Eigenstep Algorithm with Noisy Observations.

3.7.2 Conclusions

The eigenstep algorithm provides about four orders of magnitude reduction in the number of iterations while yielding almost identical performance to the gradient ascent algorithm. The addition of the temperature gradient ascent algorithm resolves the problem with initial temperatures. While the gradient ascent algorithm could be used for ridge finding as well, it would require waiting several hundred thousand iterations for the gradient ascent algorithm to converge after adjusting the initial temperature. This algorithm is taken as the primary approach to solving for the maximum likelihood solution to (3.1)

3.7.3 Model Utility

As the model has been developed, the hope has been that the addition of the emissivity to the variance of the observations will provide extra information, improving the performance of the estimate. As a check on this, the variance was removed on the downwelling radiance and a fixed value was used. The performance in this case will help see how much improvement the additional look at the variance has provided, as this model will only have variance in the observation noise. With regards to estimating the temperature and emissivity, this will reduce the model to only having information from the mean. To be clear, the distribution on the observations with a fixed downwelling radiance is

$$\mathbf{y} = \Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon} - \Lambda_{(1-\boldsymbol{\epsilon})}\boldsymbol{\mu} + \mathbf{n}, \tag{3.24}$$

with the observations having a Gaussian distribution with mean $\Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon} - \Lambda_{(1-\boldsymbol{\epsilon})}\boldsymbol{\mu}$ and variance $\sigma^2 I$.

Derivative Changes

In running the eigenstep algorithm, the noise-free variant cannot be used, as that would leave no variation, and thus no likelihood function. In the noisy variant, the covariance must be reduced to only the noise variance.

This does not change the temperature derivatives, but does change the emissivity derivatives and the any Hessian elements with an emissivity derivative.

For the derivatives, the derivative with respect to ϵ_ℓ becomes

$$\frac{\partial L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon},T)}{\partial \epsilon_\ell} = \sum_{i=1}^{N} -\frac{(y_{i,\ell} - B(T)_\ell)}{\sigma^2}\mathbf{e}_\ell^T\mathbf{w}_i. \tag{3.25}$$

For the Hessian, we consider the second derivative with respect to T, which is

$$\frac{\partial^2 L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon},T)}{\partial T^2} = \sum_{i=1}^{N} \frac{\partial B(T)_k/\partial T}{\sigma^2}\mathbf{e}_\ell^T\mathbf{w}_i - \frac{(y_{i,\ell} - B(T)_\ell)}{\sigma^2}\mathbf{e}_\ell^T\mathbf{w}_i', \tag{3.26}$$

where $\partial B(T)_k/\partial T$ is defined in (3.4). The second derivative with respect to ϵ_ℓ is

$$\frac{\partial^2 L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon},T)}{\partial \epsilon_\ell^2} = \sum_{i=1}^{N} -\frac{(y_{i,\ell} - B(T)_\ell)^2}{\sigma^2}. \tag{3.27}$$

The second derivative with respect to ϵ_m is

$$\frac{\partial^2 L(\mathbf{y}_1..\mathbf{y}_N|\boldsymbol{\epsilon},T)}{\partial \epsilon_\ell \partial \epsilon_m} = 0. \tag{3.28}$$

This is due to the lack of correlation between elements.

Performance

The performance for the algorithm when the downwelling variance is removed is significantly different. Figure 3.40 shows that there is a long period of initial oscillation. This is the temperature gradient ascent, which we did not tune for the new algorithm. After 113 iterations, the gradient ascent converges. A long flat period follows with little change. These 376 iterations are the convergence of the Newton step onto the top of the ridge. The constraint on likelihood change had to be tightened to 1 from 10 due to the function not finding the ridge top without this change. This constraint takes longer to reach, hence the extra iterations. After this, there is some jumping around as the eigenstep tries to converge. A backtracking line search was required on the Newton steps in the eigenstep portion of the the algorithm in order to get the algorithm to work, as the Newton step had difficulty finding the top of the ridge in this case as well. The final converged temperature was $302.1065°$K, which is neither close to the true value or the value achieved with downwelling

variance present.

Since this temperature is far from the true value, the emissivity would likewise be expected to be inaccurate. Figure 3.41 shows this to be the case. The shape is good, but the emissivity is a little more the 0.35 away from the true value.

The likelihood plot, shown in Figure 3.42, is also interesting, as the Newton step onto the ridge does not actually cause a jump in likelihood. This means that there is little change in the likelihood between the initial emissivity and the ridge emissivity, which should be better due to the improved shape. This suggests that the surface of the ridge is flatter and wider than it is with variance in the downwelling.

It is also interesting that when the eigenstep-portion of the algorithm begins, the likelihood drops significantly, but then climbs back up. This suggests that, despite the ridge direction being a theoretically good direction in which to optimize, for this case, the algorithm is actually stepping off the ridge by quite a bit.

As a test to understand how the algorithm converges, the initial step size of the eigenstep was changed to see what effect this would have. This choice was made since only a few steps are taken with the eigenstep and they do not seem to be great step choices, as shown by the large reduction in the likelihood. The step size could impact the convergent point in this case.

In the case where there is downwelling variance, this slows or speeds up the convergence, but the final result is not changed, as the algorithm finds a maximum.

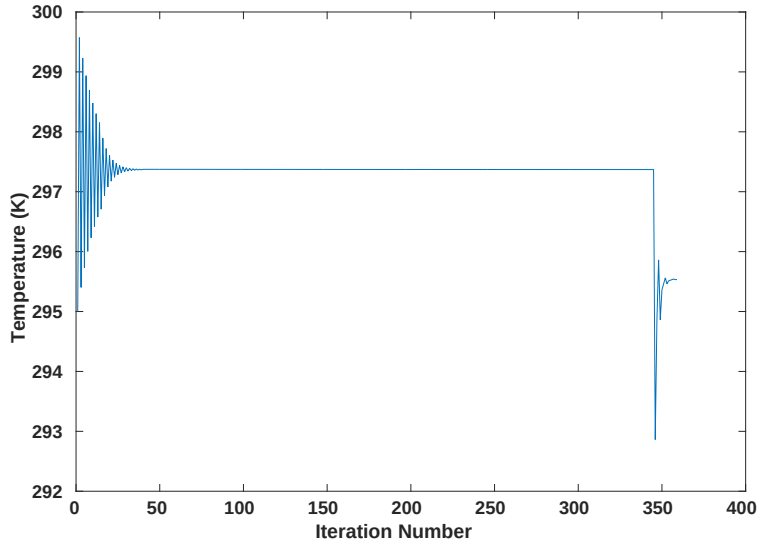

Fig. 3.40: Temperature Versus Iteration for Eigenstep Algorithm with No Downwelling Variance.

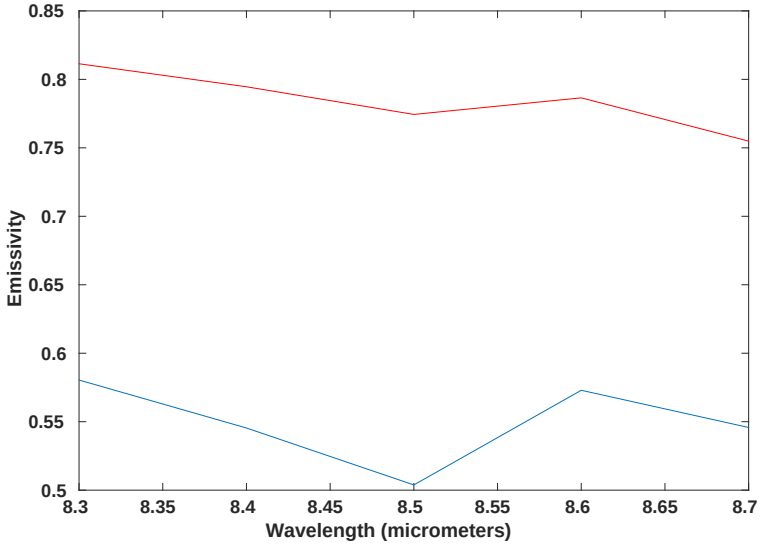

Fig. 3.41: Emissivity Versus Wavelength for Eigenstep Algorithm with No Downwelling Variance.

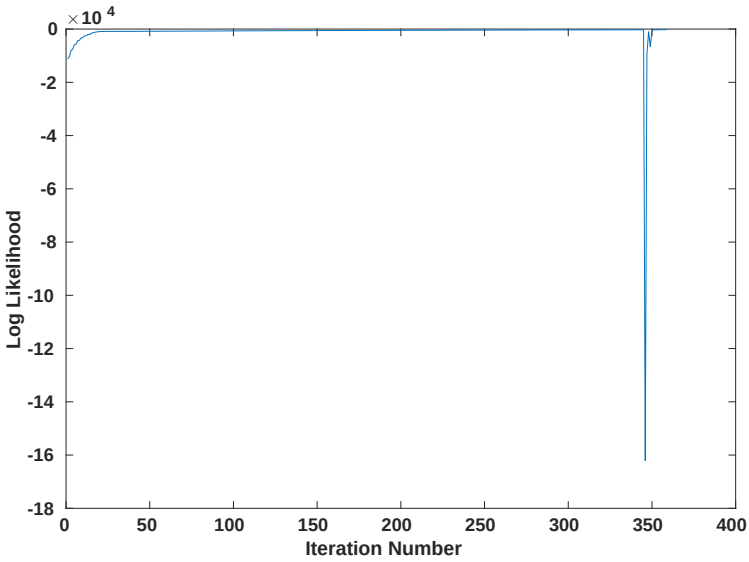

Fig. 3.42: Likelihood Versus Iteration for Eigenstep Algorithm with No Downwelling Variance.

In the case without downwelling variance, this actually changes the converged value of the temperature (and thus the emissivity). Table 3.1 summarizes the results in terms of the size of the initial step. The table shows that the step size does affect the final convergence. Interestingly, the final likelihood value is very different, suggesting that the real issue here is that the algorithm cannot converge to the true maximum.

As an alternative approach to finding the maximum, the gradient ascent approach was applied which, while working, was very slow in the previous applications. The algorithm is essentially the same as before, except with the new gradients being used.

Applying the gradient ascent algorithm in this case took over $350,000$ iterations. The plots are of little interest due to their flatness, although it was noted that initial oscillation in the temperature as the shape of the emissivity adjusts was still seen.

Interestingly, there is almost no change after the first 100 iterations. The final temperature is $295.1431°$K, while at the 100th iteration is $295.1424°$K. Not only does most of the iterating not affect the result, but we initialized at $295°$K, so there was almost no change in the temperature.

Similarly, the likelihood is -21.7871 at 100 iterations, but only improves to -21.7872 by the convergence. There is a lot of change in the lower decimal places, but this change is not partic-

Table 3.1: Eigenstep Algorithm Comparison with Differing Initial Eigenstep Sizes

Initial Step Size	Converged Temperature (K)	Converged Likelihood
2.5	300.0924	$-5,360.0261$
2	295.5332	-243.5009
1	296.1988	-90.3493
.5	296.7927	-44.3075

ularly significant, especially considering that the temperature (and the emissivity) are essentially converged at this point. There is significant change in the first 100 iterations, but this is due to the change in emissivity as the algorithm adjusts its shape. By 100 iterations, the emissivity is the correct shape, but is shifted significantly, as would be expected, given the difference in temperature.

As a comparison, the initial temperature was changed to $292°$K. In this case, instead of running to full convergence, the algorithm was run until the change in likelihood was $1e-7$, which gave slightly more than 100 iterations. Convergence would be expected by this point, based on the above performance.

The final temperature was $292.2332°$K and the final likelihood was -17.1183. The emissivity was appropriately shaped and shifted by slightly more than 0.1, which is consistent with the temperature.

A second comparison with the same setup, but the initial temperature set to $290°$K was also run. In this case, the final temperature was $291.9217°$K and the final likelihood was -17.4297. The emissivity was again about 0.1 off.

Conclusions

These three results suggest that the initial temperature has a strong effect on where the algorithm converges. Further, there appears to be significant numerical flatness, as the algorithm is unable to move from the -21 ranged likelihood when initialized at $295°$K to the -17 ranged likelihood that was achieved with lower temperatures. Given that change in likelihood all three of the points we selected was below the ten-thousandths place in the decimal, it seems likely that the function has some numerically flat regions.

In comparison with the results with downwelling variance, almost no change is seen in temperature or emissivity, only a change in the emissivity shape. In other words, the algorithm is able to find the true value within a shift (it is ambiguous).

When there is downwelling variance, even when there are small changes in likelihood, there are still significant steps in temperature or emissivity as the algorithm converges. Further, the algorithm converges to identical points for the same data, regardless of the step size or initialization.

The model with downwelling variance appears to add sufficient curvature to the surface and allows an algorithm to find a unique solution, as opposed to the multiple ambiguous solutions that might be found without it.

3.8 Real Data

Having developed the eigenstep algorithm and shown that it will converge for some simulated data, an important question remains: how does the algorithm perform with real data? Some data taken with the Denali sensor was used [49]. No ground truth was available for this data, which makes it difficult to check the results. Further, there is no knowledge of the observation noise or the downwelling variance. To make things more difficult, it is unclear if the pixels being observed are truly the pure material. These problems aside, the algorithm can still be applied to a set of real data. A material identified in the Denali data set as slate was used as a test material since slate was used in the other tests.

Based on this, some approximation of the slate emissivity curve was expected to be seen. The ambient temperature at the time of the tests was approximately 277°K. While the material may be much hotter or colder than this, we do expect something around this value.

An initial test run was made using the downwelling mean, covariance and noise variance tested above, but the algorithm ran up against the boundary for emissivity and the Hessian became ill conditioned.

A check of the estimated covariance showed that the diagonal elements were 6 to 10 times larger than the current values and the off-diagonal elements were even larger. In short, the downwelling covariance estimate is very poor.

For additional comparison, the values for the data set and the generated noisy data for five wavelengths are shown in Figure 3.43.

There is almost a 100 microflick difference between the values. The exact cause of this discrepancy is unclear. Possible sources include the downwelling mean not being exact for this time of day. There does not appear to be a significant difference in noise between the two curves, but the covariance estimate is large for the real observations, so it is possible that the change is due to this. It is also possible that the observation noise is very large, which would have a similar effect.

As a simple fix to the algorithm, the downwelling covariance was changed to the estimated covariance from the data. This allowed the algorithm to converge easily. The other parameters were left the same. Only 9 observations were available for the test.

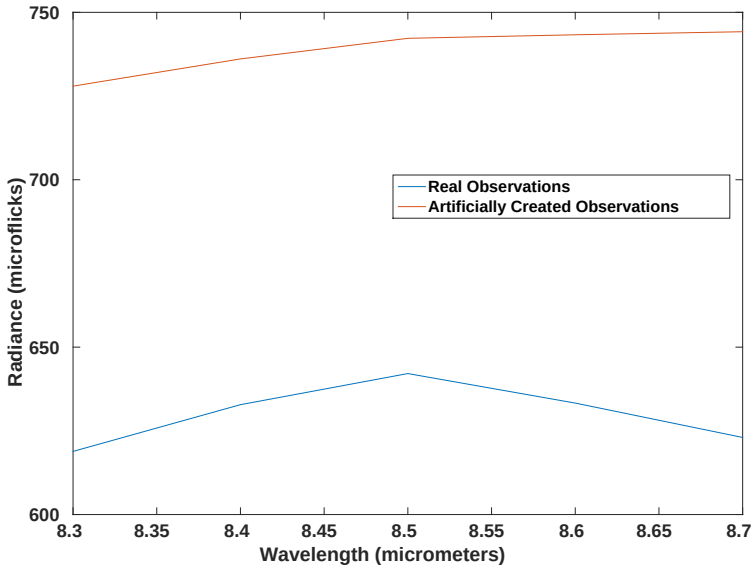

Fig. 3.43: Comparison of Real Data Observations and Artificially Generated Observations.

The likelihood plot is shown in Figure 3.44. The likelihood is constantly increasing, so the algorithm is moving in the correct direction. The algorithm takes about 60 iterations to converge, which is similar to the performance seen above. The first 31 steps are temperature convergence. Three steps are needed to reach the ridge. The remaining 26 steps are eigenstep iterations.

For the temperature, shown in Figure 3.45, the algorithm moves generally in the right direction. Once it reaches the eigenstep iterations, the algorithm moves in the wrong direction, at least based on the expectation of the temperature.

For emissivity, the expected emissivity of slate is the only point for comparison. This could be incorrect due to mixing in the pixels or variance in the real signature of the data and our library reference. The comparison is shown in Figure 3.46. The estimated shape is actually good, but the emissivity is shifted. This matches well with the temperature performance.

Based on these results, it appears that there is some bias. This bias could potentially be due to a shift in the downwelling or to a larger than expected amount of observation noise.

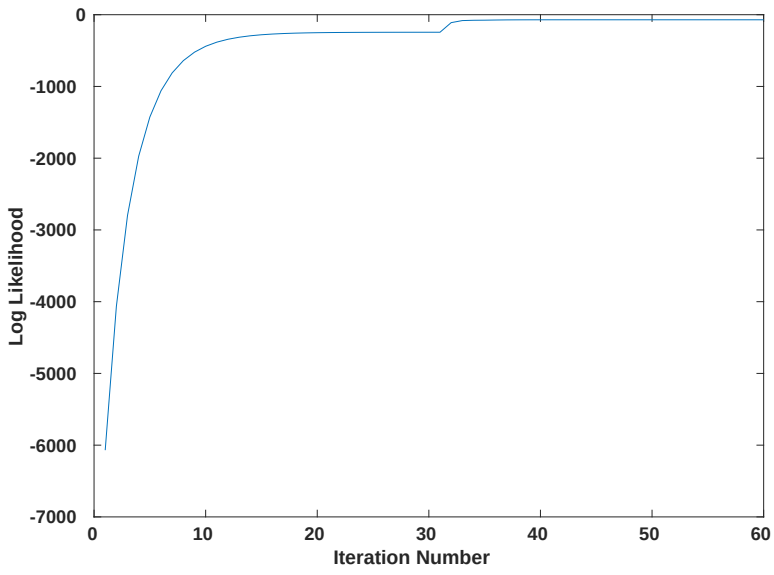

Fig. 3.44: Likelihood Versus Iteration Number for Real Observations with Estimated Downwelling Covariance.

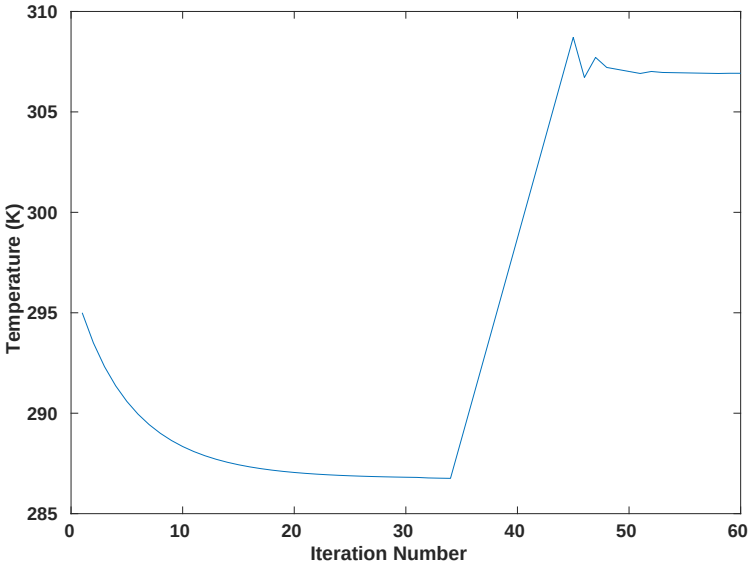

Fig. 3.45: Temperature Versus Iteration Number for Real Observations with Estimated Downwelling Covariance.

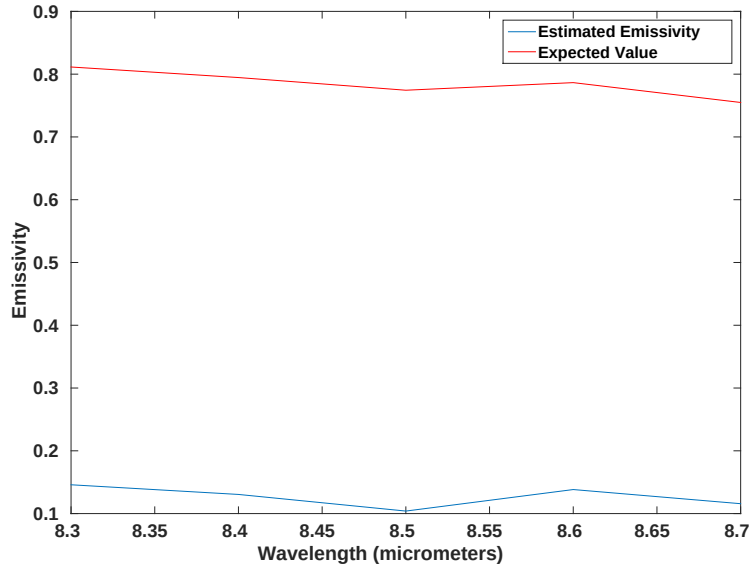

Fig. 3.46: Estimated Emissivity Compared to Expected Emissivity for Real Observations with Estimated Downwelling Covariance.

First, shifting the downwelling mean was explored to account for the shift in observations. There was reason to suspect this downwelling could introduce error, as it is an average across a portion of the day and may not reflect the exact mean at the time of measurement. All elements of the downwelling mean were shifted by 100 in order to account for the difference between the artificial and real data.

The results are essentially identical, except that the temperature shifts up by about $25°$K to about $330°$K. There is a corresponding shift in the emissivity. This suggests that this is not really providing much help.

Next, increasing the amount of noise was tried to help account for the shift in observation magnitude. The noise variance was set to $\sigma^2 = 75$. While this number seems excessively large, it is on the same order of magnitude as the downwelling covariance and also is on an order that would provide the shift that was seen in the observations.

The results are again almost identical, but are more accurate this time. The final temperature has shifted down to about $290°$K, which is much closer to what would be expected. Correspondingly, the emissivity has shifted up so that the delta between estimated and expected is about 0.5, as opposed to about 0.65.

As a final test, the initial estimate of the emissivity was fixed to the library values for slate. Temperature initialization was then performed to see what the algorithm would converge to. It estimated the temperature as about $281°$K, which is reasonable given the ambient temperature of about $277°$K.

While this performance is perhaps not as good as desired, there are several positives to take away. First, the algorithm did converge on real data. Second, the algorithm converges to an emissivity with the correct shape. Third, the expected library emissivity allows for temperature gradient ascent convergence to a temperature that is reasonable given the ambient air temperature.

All of this was possible with no tuning of the algorithm other than changing the downwelling covariance to the estimate based on the observations. No knowledge of the true noise or even the true downwelling mean was required. And this was all done with only 9 observations which may not even meet the pure pixel requirement.

With more access to MODTRAN or additional measured data and especially with ground truth reference data and assurances that the pixels are the material in question, it is likely that the parameters could be better estimated.

3.9 Conclusions

This section has:

- Tested traditional algorithms such as gradient ascent and Newton step as potential approaches for optimization.
- Explored their performance characteristics to identify significant features of the objective function, namely the ridge, that slow or prevent convergence.
- Identified a characteristic of this feature, namely an eigenvector of the Hessian, that can be used to overcome the performance impediment caused by the ridge through the eigenstep approach.
- Developed an algorithm that makes uses of traditional approaches for initialization and the new eigenstep approach to quickly converge to the maximum value.
- Verified that the model developed in Chapter 2 decreases the ambiguity of the estimator.
- Demonstrated that the algorithm will converge with minimal changes for real data, despite the lack of sufficient knowledge of said real data.

By testing several algorithms, the ridge shape of the function and its relative flatness were found to cause problems with optimization. The gradient ascent approach was affected in that convergence was incredibly slow. The Newton step algorithm moved more quickly, but was only able to find the ridge and proved unable to move along it.

By exploiting the shape of the function and making use of the eigenvectors of the Hessian, a new approach, the eigenstep algorithm, was developed which finds the maximum value with the same accuracy as the gradient ascent while quickly converging. The speed of the Newton step was

used for finding the ridge, while the gradient ascent approach was used to help solve problems with initial temperatures.

After an extensive search of the literature, making use of the eigenvalues of the Hessian while on a ridge as an optimization technique for moving along a ridge appears to be a new approach. No similar techniques were identified. While this algorithm is tailored to the specific problem of the TES setting, this approach has value for finding an optimal point on the ridge of any function and could easily be adapted to other objectives that display this behavior.

Also, adding variance to the downwelling radiance was shown to improve the quality of the model. This contrasts favorably to other approaches, which are often only accurate to within some unknown amount of scaling.

The chapter concluded by demonstrating that the algorithm will converge with real data, even when poorly initialized. Although the performance is not exceptional, it does display some of the characteristics expected of the algorithm, namely that the shape of the emissivity curve is comparable to library data.

CHAPTER 4
RESULTS

4.1 Introduction

Given the model developed in Chapter 2, Chapter 3 developed an algorithm that estimates the temperature and emissivity for a set of observations. The preliminary results from in Chapter 3 suggest good convergence speed and performance, although the result is not exact.

In general, an estimator should not be assumed to be unbiased. Although bias can be determined mathematically, demonstrating performance is often an easier way to show bias (or lack thereof). Also, accuracy and convergence speed may vary as parameters or observations change. Any potential bias or performance variance must be understood in order to truly judge the quality of an algorithm. In this chapter, the algorithm is tested under a variety of conditions in order to try and characterize these and other aspects of the eigenstep algorithm's performance.

4.2 Test Descriptions

In order to characterize the performance of the eigenstep algorithm, it must be tested in a variety of situations. Tests should be conducted with different temperatures and emissivities, as well as different variances for both the downwelling radiance and the observation noise. The number of wavelengths and observations should also be varied.

There is no intention to exhaustively test all combinations, as this is excessive. Rather, the goal is to try to appropriately characterize performance of the algorithm as it relates to changes in each parameter. For each combination, the algorithm will be run 100 times.

4.2.1 Parameter Values

Two materials will be considered, slate and alabaster. These materials were chosen because they cover a wide range of emissivity values. The values for emissivity that were used were taken

from a library of material values. For the five wavelength case, the emissivity vector for slate is $[0.8114, 0.7946, 0.7744, 0.7865, 0.7549]$ for the wavelengths 8.3μm to 8.7μm in 0.1μm steps. For the alabaster, the vector for the same wavelengths is $[0.5161, 0.4494, 0.3682, 0.3111, 0.3318]$. The wavelength will be tested at five wavelengths: 8.3μm to 8.7μm in 0.1μm steps.

Also, 25 wavelengths will be tested from 8.3μm to 10.7μm , again in 0.1μm steps. For the two materials, with 25 wavelengths, the vectors of emissivities are

$$\boldsymbol{\epsilon}_{slate} = \begin{bmatrix} 0.8114 \\ 0.7946 \\ 0.7744 \\ 0.7865 \\ 0.7549 \\ 0.7162 \\ 0.7005 \\ 0.6880 \\ 0.6770 \\ 0.6790 \\ 0.6945 \\ 0.7356 \\ 0.7464 \\ 0.7232 \\ 0.7407 \\ 0.7670 \\ 0.7787 \\ 0.7871 \\ 0.8004 \\ 0.8192 \\ 0.8416 \\ 0.8632 \\ 0.8788 \\ 0.8903 \\ 0.8973 \end{bmatrix} \quad \boldsymbol{\epsilon}_{alabaster} = \begin{bmatrix} 0.5160 \\ 0.4493 \\ 0.3682 \\ 0.3110 \\ 0.3317 \\ 0.4117 \\ 0.4952 \\ 0.5976 \\ 0.6779 \\ 0.7143 \\ 0.7342 \\ 0.7470 \\ 0.7562 \\ 0.7626 \\ 0.7677 \\ 0.7724 \\ 0.7772 \\ 0.7769 \\ 0.7806 \\ 0.7831 \\ 0.7852 \\ 0.7868 \\ 0.7885 \\ 0.7901 \\ 0.7916 \end{bmatrix} \tag{4.1}$$

Each material will be tested at three different temperatures, $280°$K, $290°$K, and $300°$K. Generally, the temperature is expected to be between $270°$K and $310°$K, so this will test across that range.

In order to see how the algorithm responds to different numbers of observations, 5, 10, and 20 observations will be tested.

For the downwelling variance, three cases will be explored. The first case will involve no correlation, with the diagonal set as 5.6 and all other elements set as 0. For the second case, the

diagonal will be 5.6, the off diagonal will be 0.6, and all other elements will be 0. For the third case, the diagonal 5.6 will be, the off diagonal 0.6, and all other elements 0.1. This should give a better appreciation for the impact that the inclusion of correlation in the covariance matrix has on the algorithm performance.

For all tests, the same downwelling mean will be used, which is the vector $[547.8675, 572.2968, 596.0602, 561.5744, 570.2581]$ for the five wavelength case and

$$\boldsymbol{\mu} = \begin{bmatrix} 547.8675 \\ 572.2968 \\ 596.0601 \\ 561.5744 \\ 570.2581 \\ 595.6379 \\ 577.4879 \\ 587.1237 \\ 588.9409 \\ 582.4457 \\ 595.4721 \\ 635.2826 \\ 657.5691 \\ 636.3335 \\ 637.2087 \\ 620.7805 \\ 599.3136 \\ 571.9482 \\ 567.4559 \\ 575.4198 \\ 567.0790 \\ 561.0350 \\ 572.5703 \\ 576.6303 \\ 565.5263 \end{bmatrix} \tag{4.2}$$

for the 25 wavelength case.

For the observation noise, the noise variance σ^2 will be set to 0.5, 1, and 2.

In order to keep the tests simple, only one variable will be changed at a time. As such, a baseline group of settings has been chosen from amongst the list. In any test where parameters are not specified more specifically, the baseline parameters will be those in use.

The baseline parameters are slate with five wavelengths and the five values specified for that case above. The temperature will be set to 290°K, 10 observations will be used, the observation

noise will be $\sigma^2 = 1$, and the downwelling variance will have a diagonal of 5.6, the off diagonal 0.6, and all other elements 0.1.

For initialization, all initial emissivities will be set to 0.5 and the initial temperature will be set to $295°$K. This temperature will not matter, since temperature gradient ascent algorithm will adjust the temperature to one that will work in the initialization of the algorithm. It will, however impact convergence of the algorithm, as it may take more or less steps to converge the temperature gradient ascent, depending on the true temperature.

4.3 Varying Emissivity

To begin, the two materials are compared. For the baseline (slate) case, Figures 4.1, 4.2, and 4.3 show the distributions of the converged likelihood, temperature, and emissivity, respectively. For the objective function, Figure 4.1 shows that there is a spread of about 25 in log likelihood. The majority of the weight is towards the center, but the distribution is rather wide, given that so much change is seen in temperature and emissivity for small likelihood changes in the last chapter.

Figure 4.2 shows that the temperature is typically close to about $288°$K, with most of the values being far below this. However, the total spread is almost $6°$K. This distribution is not particularly good from an accuracy standpoint, as most of the results are slightly low and consistent, but there is a reasonable chance to get outliers.

As expected, given this distribution, the emissivity, shown in Figure 4.3 has a large standard deviation of around 0.1. The mean is off the true value by about 0.06. Examining this figure more closely, the shape is slightly skewed, being somewhat flatter than the true emissivity. Interestingly, the lower bounds on standard deviation has a good shape, while the upper bound is even more flat. Examination of some of the individual samples shows that, for temperature/emissivity pairs that are closer to the true value, a good shape is obtained. However, as the values move farther away, flatter performance results, which is suggested by the upper standard deviation bound and the mean.

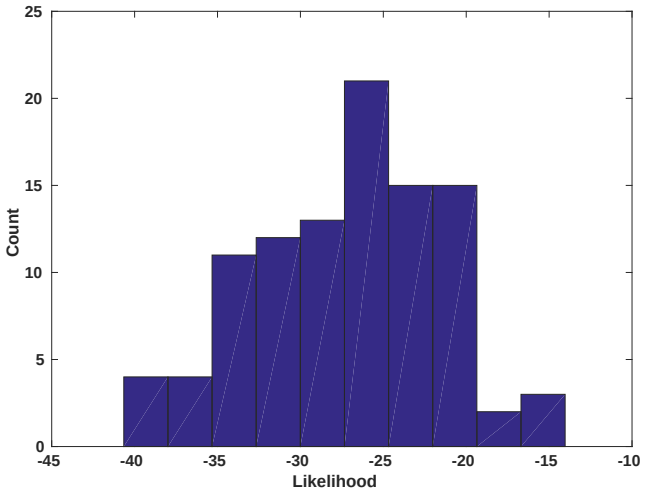

Fig. 4.1: Histogram of Maximum Likelihood for Baseline Parameters.

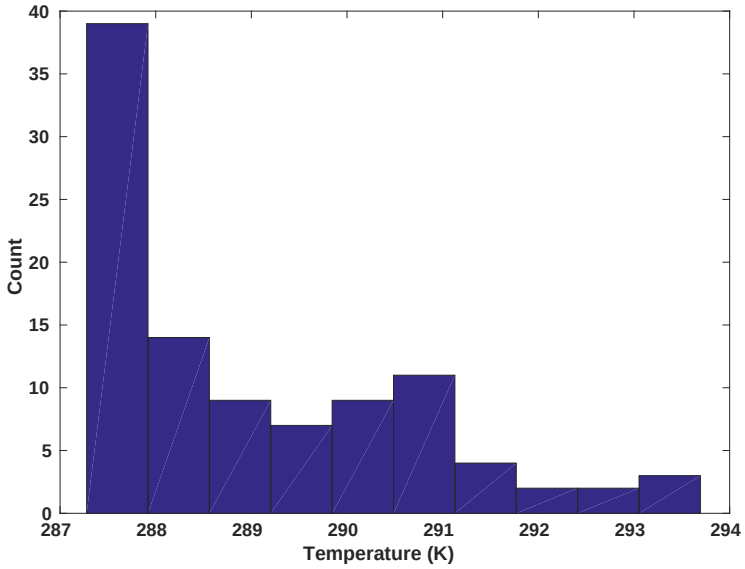

Fig. 4.2: Histogram of Estimated Temperature for Baseline Parameters. Mean $= 289.05°$K, Standard Deviation $= 1.629°$K.

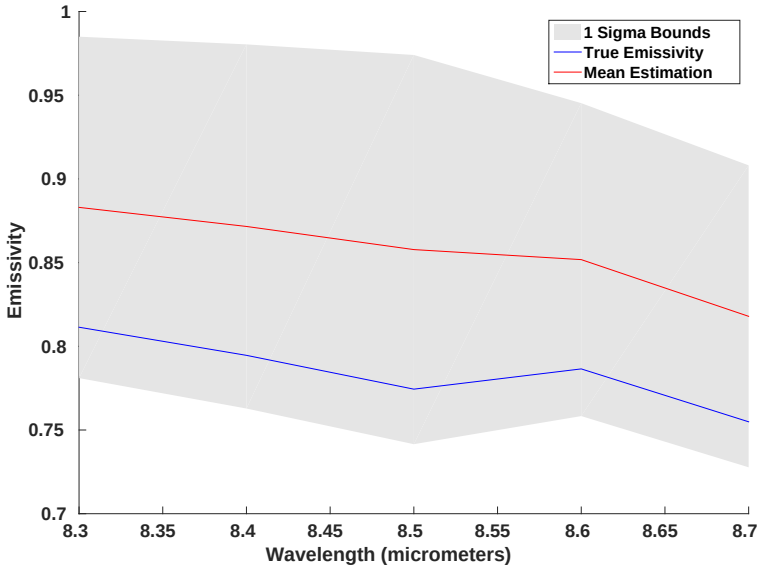

Fig. 4.3: Mean Estimated Emissivity with One-Sigma Bounds for Baseline Parameters.

In examining the convergence, the temperature convergence and ridge convergence in the initialization are both quick and consistent, with 9 steps being needed for every test for the temperature to converge and 2 steps being needed to find the ridge. After this point, the eigenstep algorithm takes over. The algorithm takes between 20 and 600 iterations to converge, with the average number of iterations being 141. The actual distribution is plotted in Figure 4.4. Most of the tests converged in less than 100 iterations, while there is a small group that took about 300 to 400 iterations. There are also some outlier results closer to 600.

For alabaster, slightly different performance is obtained. The likelihood values are actually similar in distribution and shape, as seen in Figure 4.5. For the temperature, Figure 4.6 shows that the temperature spread is much larger, almost 14°K. The majority of the results are still clumped at the low end, but in this case, this group is spread over almost 8°K. For the emissivity, seen in Figure 4.7, there is about the same difference in mean, with a slightly larger standard deviation of about .11. The shape is very good, following the true value much better than the slate case.

For convergence, the temperature converges in 5 steps repeatably, with the ridge convergence taking 3 to 4 steps, split about evenly. The remainder of the convergence takes between 10 and 59

steps, with an average of 26. The distribution is shown in Figure 4.8 and has a similar shape to the slate, although the outlying values are much smaller.

Between the two materials, differences are expected due to changes in the emissivity. Changes in performance due to higher or lower emissivity has been seen in other algorithms [50]. Smaller emissivity will result in larger variance in the observations, since it will scale up the downwelling variance. This is not strongly pronounced between these two materials, but slightly higher variance is seen in the alabaster for its lower emissivities.

Smaller emissivity will also reduce the effect of temperature error in the initial guess. For the specific examples with the alabaster, the algorithm converges in temperature faster and converges to a wider range of values, which both reflect this effect. The alabaster converges faster in the eigenstep portion, which would be expected, given that the initial emissivity of 0.5 is closer to the values that are in the true alabaster emissivity. For both materials, given the large standard deviation, it is hard to tell if there is a bias, but given the significant shift in mean, there does seem to be evidence to support that the solution is biased.

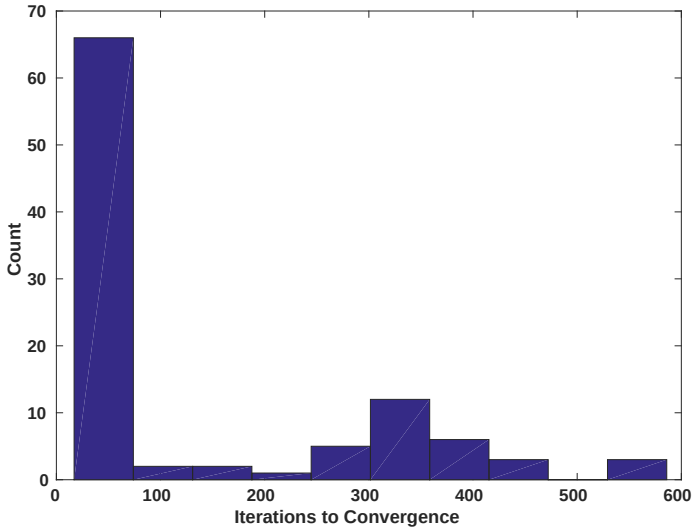

Fig. 4.4: Number of Iterations to Convergence After Initialization for Baseline Parameters.

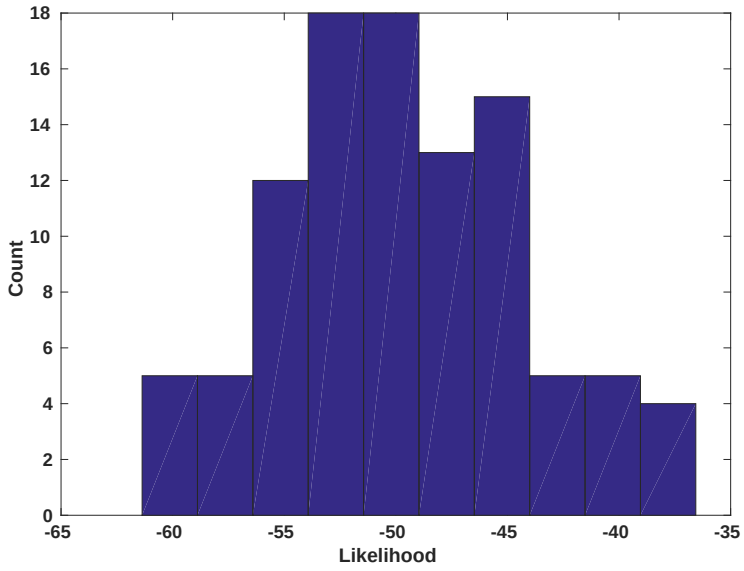

Fig. 4.5: Histogram of Maximum Likelihood for Baseline Parameters with Alabaster.

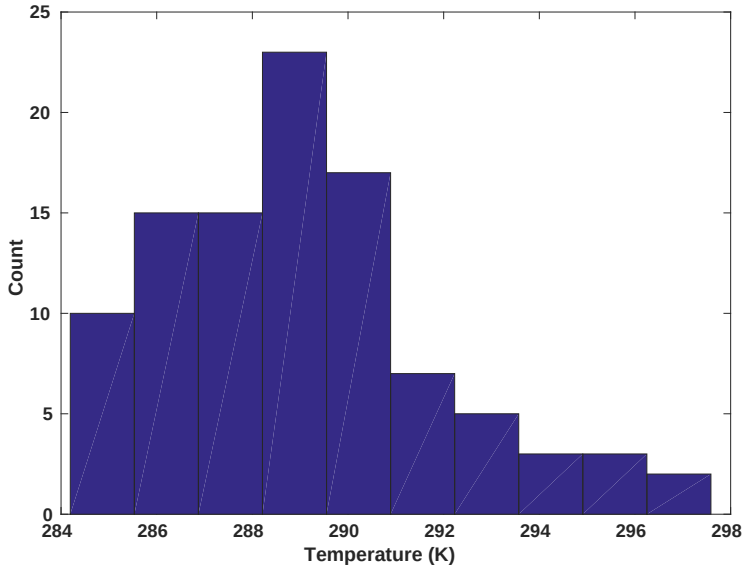

Fig. 4.6: Histogram of Estimated Temperature for Baseline Parameters with Alabaster. Mean = $288.99°$K, Standard Deviation = $2.875°$K.

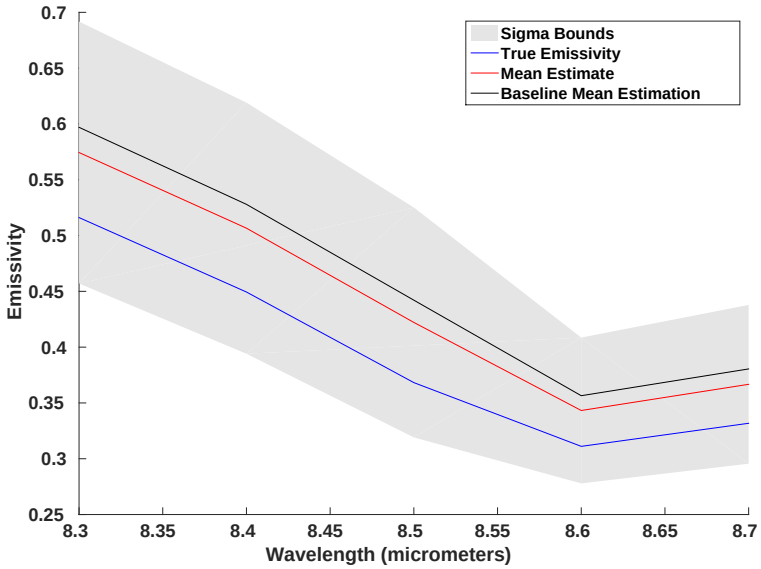

Fig. 4.7: Mean Estimated Emissivity with One-Sigma Bounds for Baseline Parameters with Alabaster.

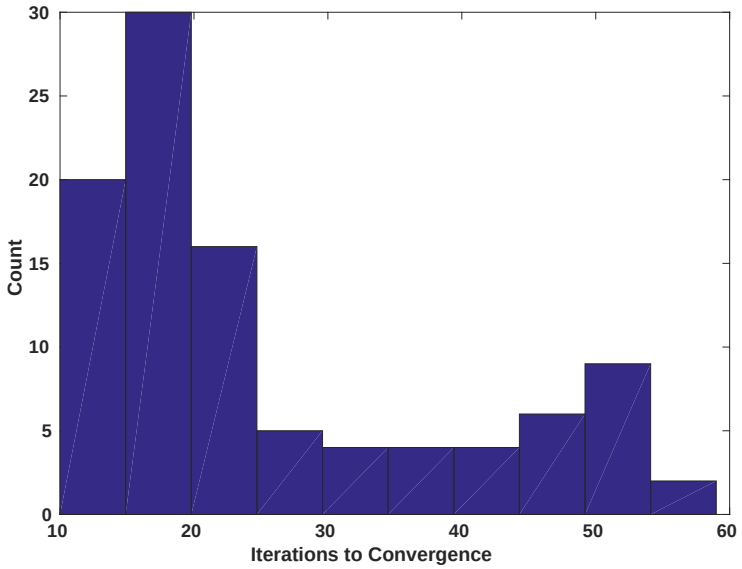

Fig. 4.8: Number of Iterations to Convergence After Initialization for Baseline Parameters with Alabaster.

4.4 Varying Temperature

Temperature is considered next. For the case where the true temperature is $280°$K, the likelihood distribution, shown in Figure 4.9 is almost identical to the $290°$K case. This is likewise the case for the likelihood with the true temperature being $300°$K, seen in Figure 4.10, as well as alabaster for both temperatures, seen in Figures 4.11 and 4.12.

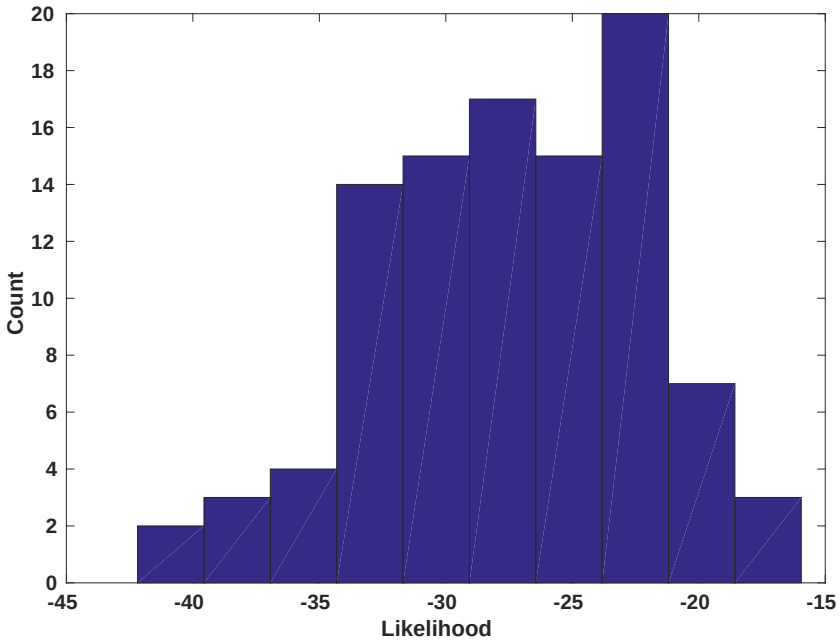

Fig. 4.9: Histogram of Maximum Likelihood for Baseline Parameters with Temperature $= 280°$K.

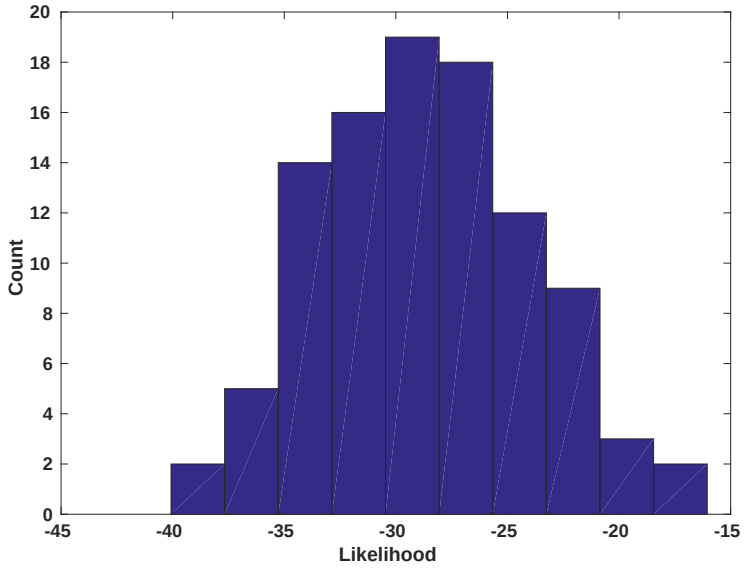

Fig. 4.10: Histogram of Maximum Likelihood for Baseline Parameters with Temperature $= 300°$K.

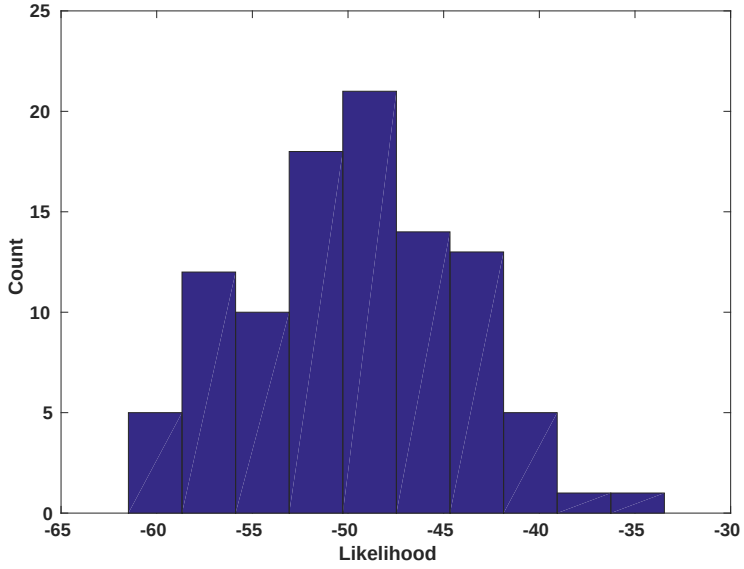

Fig. 4.11: Histogram of Maximum Likelihood for Baseline Parameters with Temperature $= 280°$K and Alabaster.

Interestingly, the temperature performance for the lower true temperature is much tighter for both materials, as seen in Figures 4.13 and 4.14, despite the likelihood spread not changing. The distribution shape for both is the same, despite the lower variance. For the higher temperature, shown in Figures 4.15 and 4.16, the distribution still has the same shape but is much closer to the overall spread for the 290°K case.

For the emissivity curves, for the lower temperature, Figures 4.17 and 4.18 show that the variance has reduced, which makes sense, given the reduction in temperature range. Of particular interest is the large bulge in the middle wavelengths for both cases. It is unclear what is causing this, but it seems to be consistent for both materials and may specifically be an artifact of running the algorithm with lower temperatures, since it is not present in any other tests. In the case of the slate, this bulge actually causes the standard deviation bounds to go above one. This does not actually mean that estimated emissivities went out of bounds. Rather, the estimates lower than the mean were low enough to increase the standard deviation beyond the bounds, in spite of no actual measurements attaining those values.

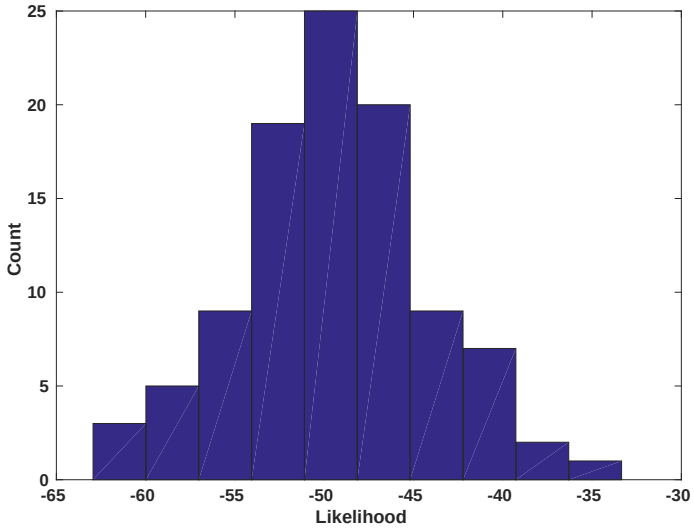

Fig. 4.12: Histogram of Maximum Likelihood for Baseline Parameters with Temperature = 300°K and Alabaster.

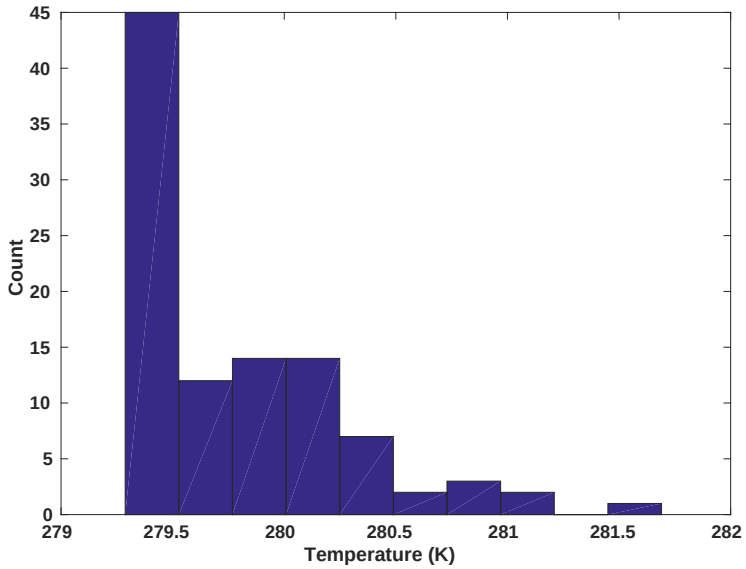

Fig. 4.13: Histogram of Estimated Temperature for Baseline Parameters with Temperature $=$ 280°K. Mean $=$ 279.78°K, Standard Deviation $=$ 0.486°K.

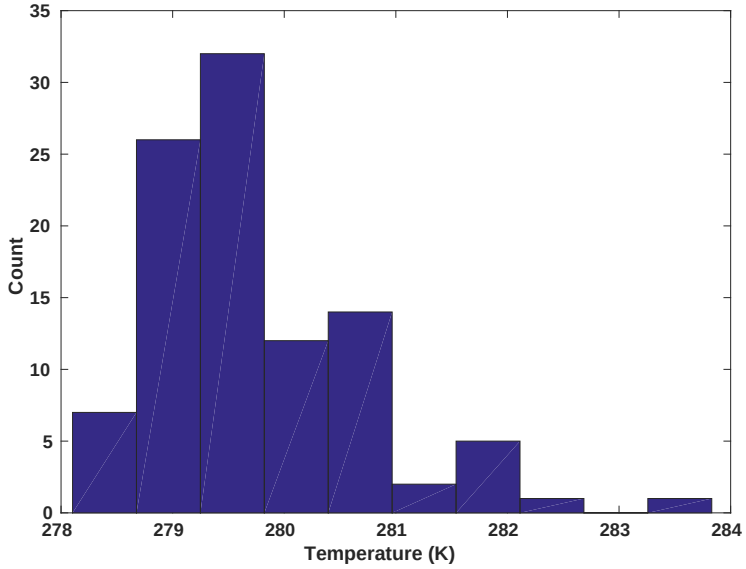

Fig. 4.14: Histogram of Estimated Temperature for Baseline Parameters with Temperature $=$ 280°K and Alabaster. Mean $=$ 279.70°K, Standard Deviation $=$ $.990^{\circ}$K.

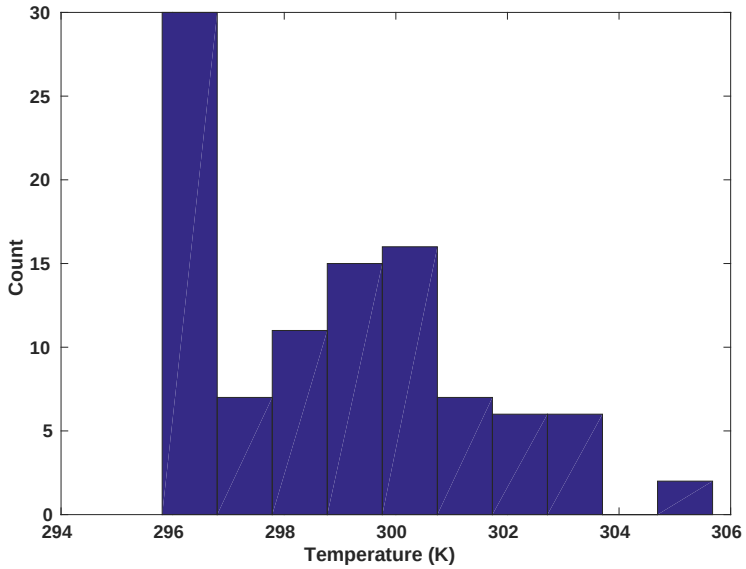

Fig. 4.15: Histogram of Estimated Temperature for Baseline Parameters with Temperature $=$ $300°$K. Mean $=$ $298.89°$K, Standard Deviation $=$ $2.390°$K.

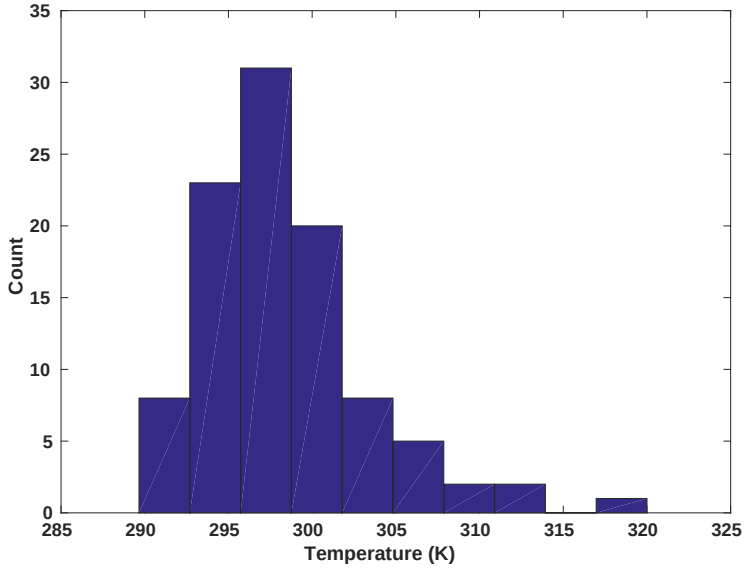

Fig. 4.16: Histogram of Estimated Temperature for Baseline Parameters with Temperature $=$ $300°$K and Alabaster. Mean $=$ $298.61°$K, Standard Deviation $=$ $4.948°$K.

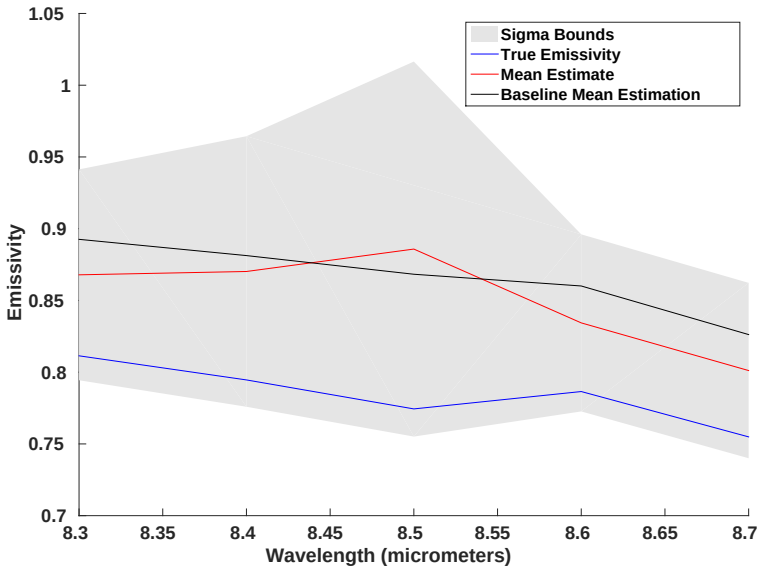

Fig. 4.17: Mean Estimated Emissivity with One-Sigma Bounds for Baseline Parameters with Temperature $= 280^{\circ}$K.

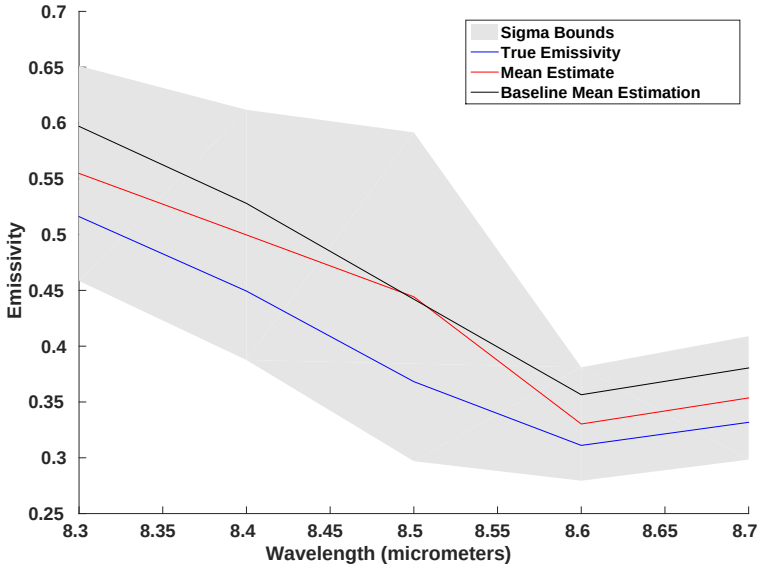

Fig. 4.18: Mean Estimated Emissivity with One-Sigma Bounds for Baseline Parameters with Temperature $= 280^{\circ}$K and Alabaster.

For the $300°$K case, the plots are almost identical to the $290°$K case, which is consistent with the temperature measurements. With regards to the convergence, the $300°$K performance was almost identical to the baseline cases, at least with regards to the eigenstep algorithm. The slate took 18 to 658 iterations with a mean of 171 and the alabaster took 9 to 80 with a mean of 28 and both having almost identical distributions. While the initialization of the alabaster was also similar, with 6 steps for temperature convergence and 4 to the ridge, the slate took an excessive number of steps to converge in temperature, taking between 154 and 163 steps, spread evenly across that range. The ridge convergence still took 2 steps. It is likely that the temperature convergence could be improved by tuning, but we simply note that the temperature being higher than the initial guess seemed to have a negative impact on this convergence.

For the $280°$ case, the initialization was similar The temperature performance was more in line with the baseline, with the slate taking 6 steps to converge and the alabaster taking 7. The ridge convergence was also better, with the slate taking 2 to 3 steps and the alabaster taking 3 to 4, with the lower number being significantly more likely in both cases.

For the overall convergence, the performance was actually quite different. The distributions are shown in Figures 4.21 and 4.22. For slate, the shape has changed, and the convergence has improved to between 23 and 72 steps with an average of 47 steps. The alabaster has similar overall performance, taking between 10 and 72 steps with an average of 33, but the distribution has actually changed shape.

Overall it appears that different temperatures can impact the performance by affecting the convergence in a variety of ways and potentially increasing the variance on the estimates.

4.5 Varying Number of Wavelengths

Varying the number of wavelengths had significant impact on the performance across the board. This performance impact is somewhat expected, as other algorithms get a similar accuracy improvement by increasing the number of wavelengths [30]. Testing was limited to 25 wavelengths for several significant reasons. First, there was a limit on the number of wavelengths available. The library data had only 47 wavelengths for many materials. Second, the Hessian and inverse Hessian become

extremely slow to compute as the number of wavelengths increases, so 25 was a compromise between a significant increase in wavelengths and an intractable increase, at least for the purposes of this research.

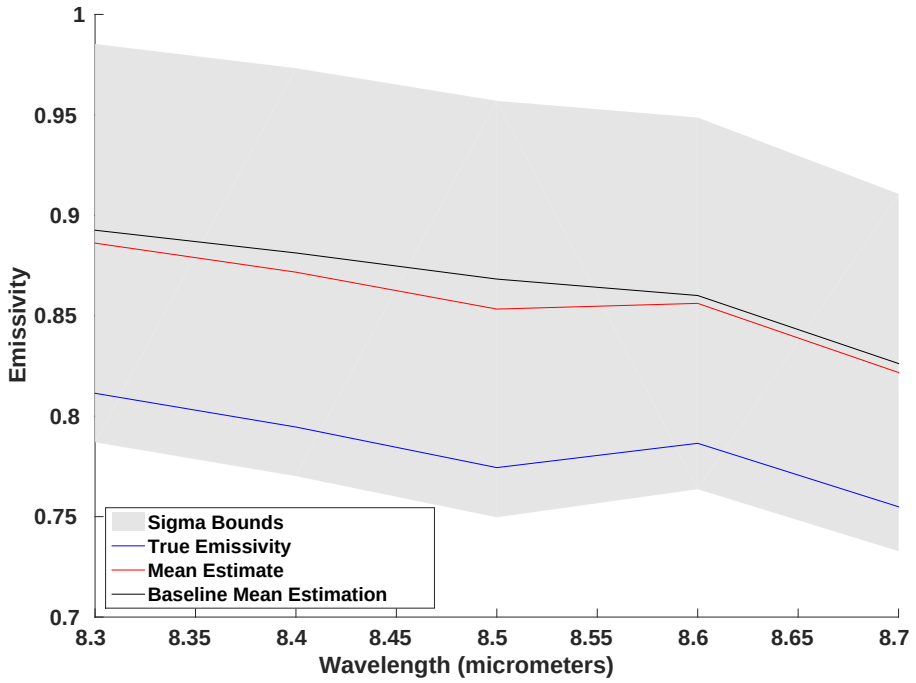

Fig. 4.19: Mean Estimated Emissivity with One-Sigma Bounds for Baseline Parameters with Temperature $= 300^{\circ}$K.

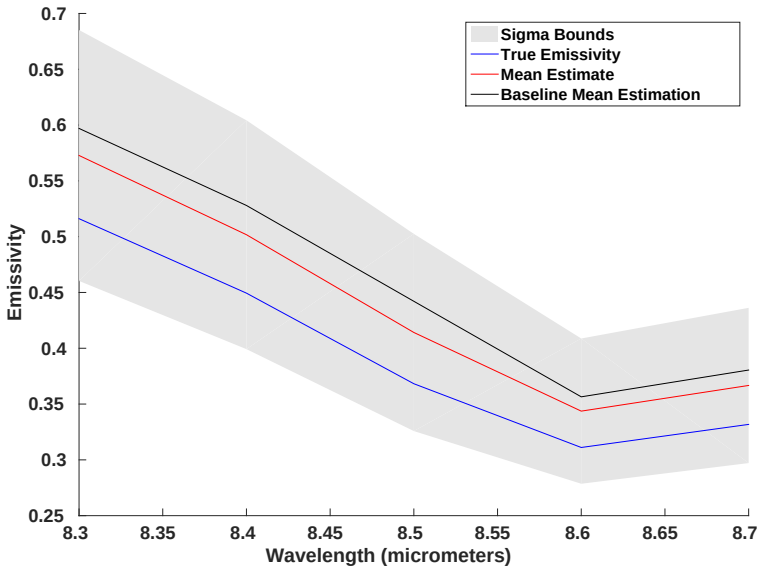

Fig. 4.20: Mean Estimated Emissivity with One-Sigma Bounds for Baseline Parameters with Temperature $= 300^{\circ}$K and Alabaster.

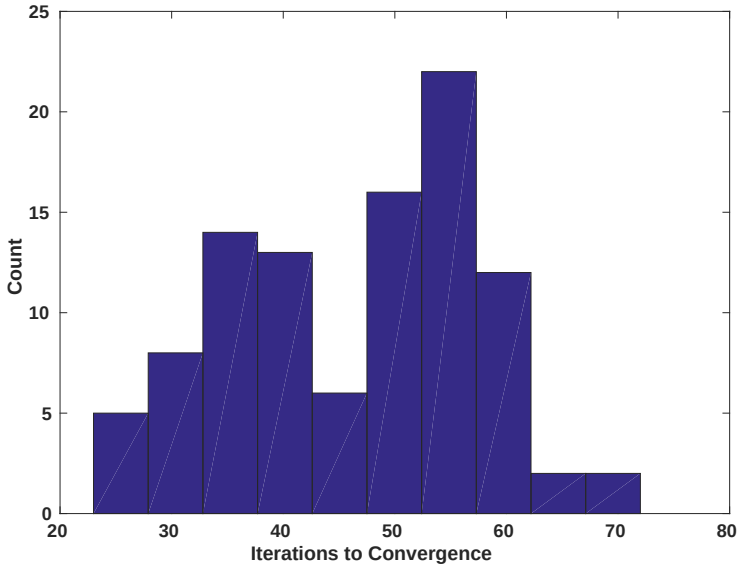

Fig. 4.21: Number of Iterations to Convergence After Initialization for Baseline Parameters with Temperature $= 280^{\circ}$K.

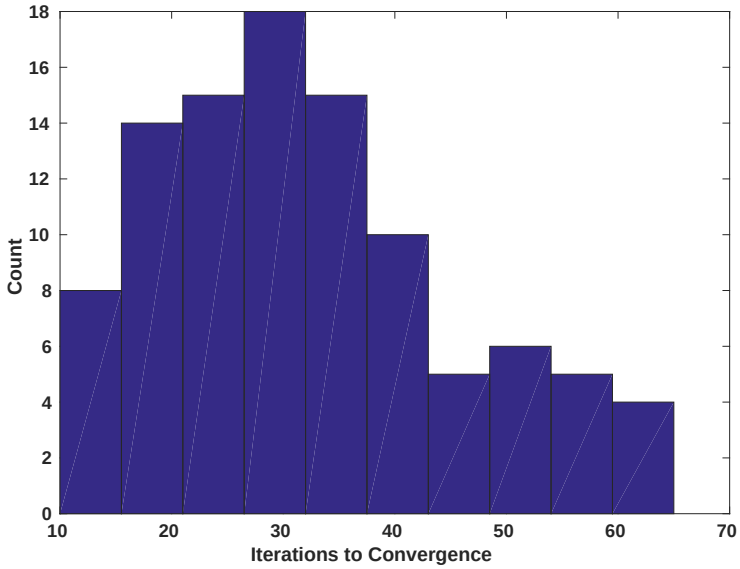

Fig. 4.22: Number of Iterations to Convergence After Initialization for Baseline Parameters with Alabaster and Temperature $= 280°$K.

Finally, tests with more wavelengths tended to result in poorly conditioned or singular Hessians, which breaks the eigenstep approach. It is not clear if this problem is a something that can be tuned in the algorithm or if the numerical flatness of the objective becomes an issue as we add wavelengths. Regardless, the performance was tested at 25 wavelengths.

For the slate, shown in Figure 4.23, the spread in likelihood is actually much larger, increasing to over 50. The likelihood values have also changed quite a bit.

For temperature, seen in Figure 4.24, the distribution is the same shape as the baseline, but is much tighter, spreading across only $3°$K. This is consistent with the reduction in emissivity standard deviation shown in 4.25. The standard deviation has been cut in half and the shape is much more accurate. The true value no longer falls within one standard deviation. The mean shift is still the about the same. The convergence initialization takes longer, requiring about 42 steps to converge in temperature and 4 to reach the ridge. The overall convergence has the same general distribution, but with fewer outliers. It takes between 29 and 336 eigenstep iterations to converge with a mean of 106.

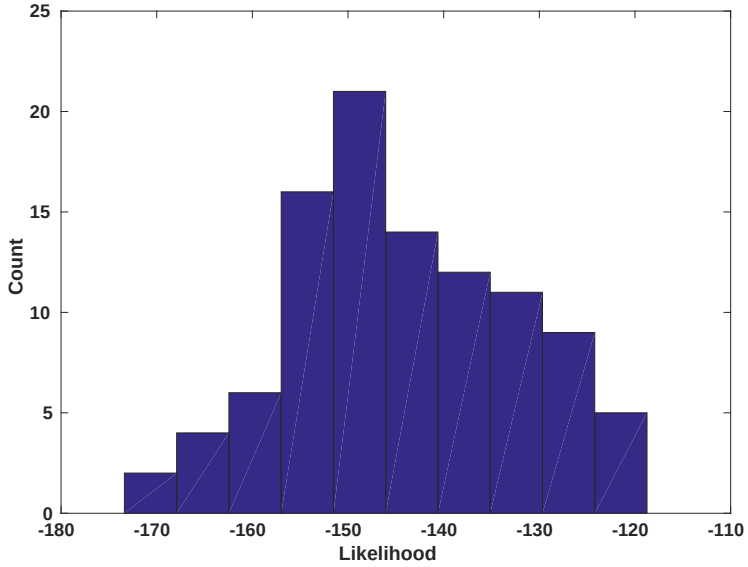

Fig. 4.23: Histogram of Maximum Likelihood for Baseline Parameters with 25 Wavelengths.

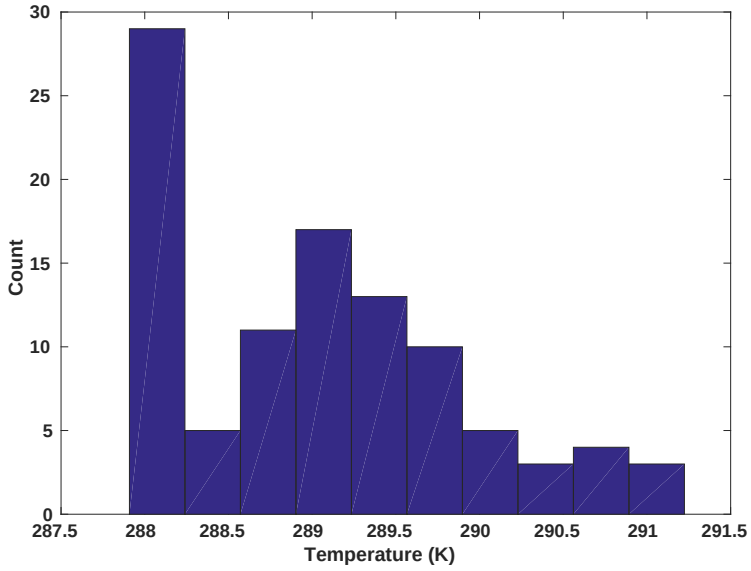

Fig. 4.24: Histogram of Estimated Temperature for Baseline Parameters with 25 Wavelengths. Mean $= 288.07°$K, Standard Deviation $= 0.851°$K.

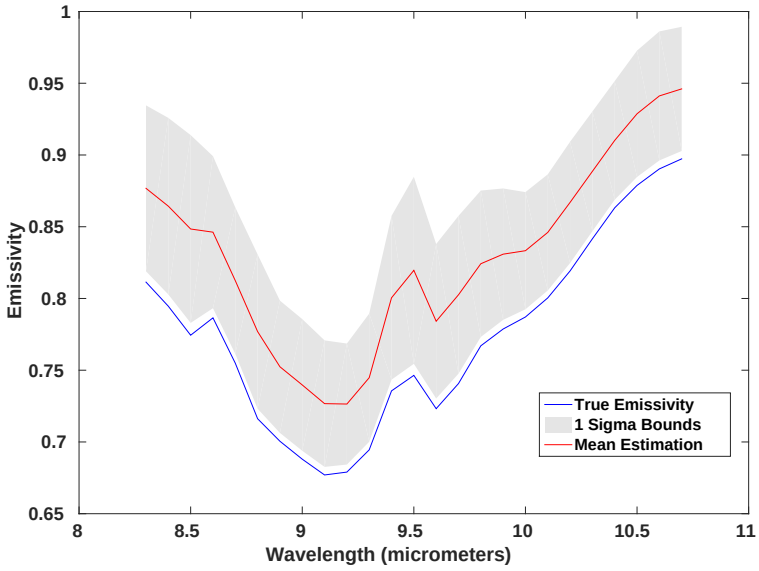

Fig. 4.25: Mean Estimated Emissivity with One-Sigma Bounds for Baseline Parameters with 25 Wavelengths.

For alabaster, the performance is likewise significantly impacted. The likelihood spread is almost 70. The distribution is similarly shaped, as we see in Figure 4.26. The temperature distribution, shown in Figure 4.27, is somewhat more uniform than in the baseline case. The spread is also reduced, down to about $4°$K in this case.

The emissivity is most interesting. Figure 4.28 shows the estimate. The resulting shape is good with some error in shape as the emissivity increases around 9.5μm. The delta between the true and estimated emissivity increases with emissivity, being around 0.05 for the emissivities below 0.7 and increasing to a maximum of 0.1 for the emissivities greater than 0.7. This strongly suggests that there is some correlation between emissivity value and the accuracy of the estimator.

The temperature convergence took 7 steps and the ridge 2, which is very similar. For eigenstep convergence, the alabaster is more similar in shape to the baseline slate, having a large peak on the low end with a secondary hump higher up. The actual distribution is shown in Figure 4.29. The number of iterations is between 25 and 94 with an average of 44.

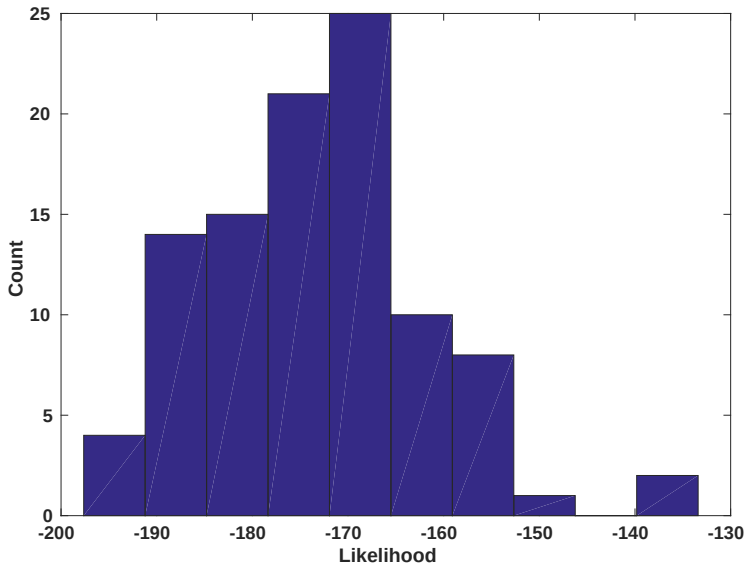

Fig. 4.26: Histogram of Maximum Likelihood for Baseline Parameters with Alabaster and 25 Wavelengths.

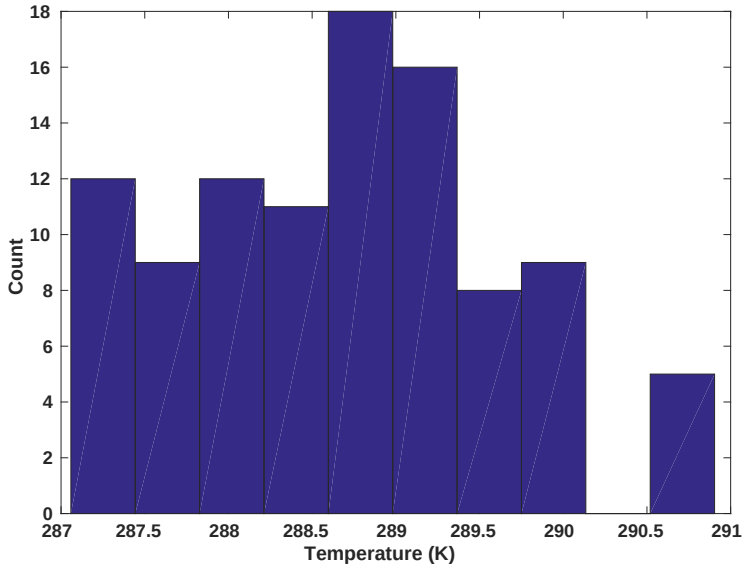

Fig. 4.27: Histogram of Estimated Temperature for Baseline Parameters with Alabaster and 25 Wavelengths. Mean $= 288.89°$K, Standard Deviation $= 0.833°$K.

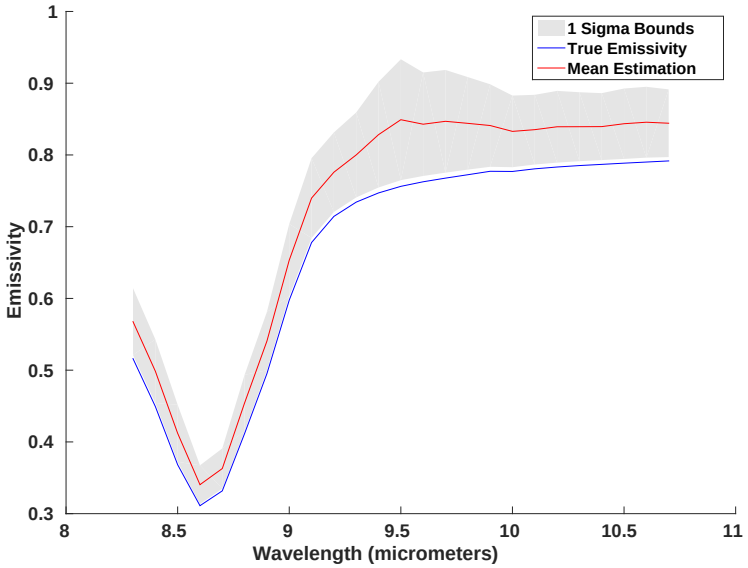

Fig. 4.28: Mean Estimated Emissivity with One-Sigma Bounds for Baseline Parameters with Alabaster and 25 Wavelengths.

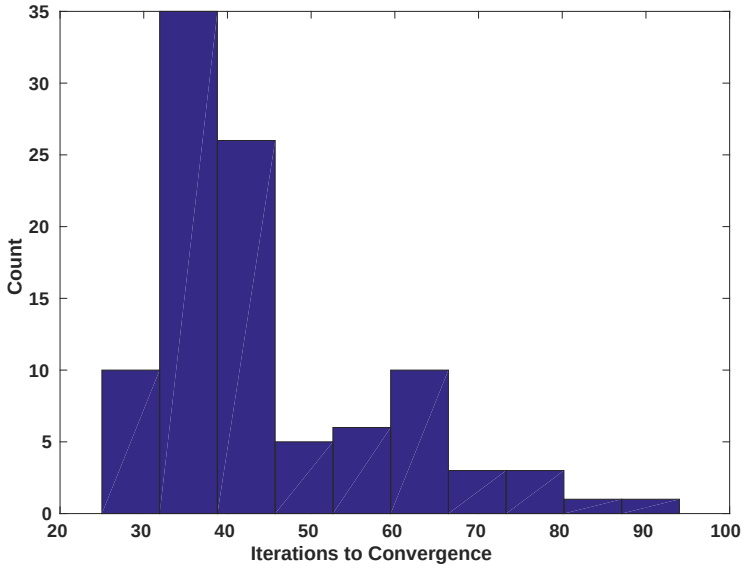

Fig. 4.29: Number of Iterations to Convergence After Initialization for Baseline Parameters with Alabaster and 25 Wavelengths.

These tests provide some clear indications that an increase in wavelengths improves the accuracy of the estimate, at least in a variance sense. Since the delta between the true and estimated values does not change, it supports the idea that the result is biased.

There is some impact in the convergence speed, but how exactly is unclear. The temperature convergence is slowed for both cases, but the eigenstep convergence is actually faster for the slate and slower for the alabaster.

4.6 Varying Number of Observations

For varying the number of observations, reducing to only 5 observations was considered first. The likelihood distributions for slate and alabaster are shown in Figures 4.30 and 4.31, respectively. The distributions are essentially the same, although the likelihood is slightly shifted. For temperature, the distributions are shown in Figures 4.32 and 4.33. In both cases, the distribution is unchanged in shape, but the spread has increased by a few degrees. For emissivity, shown in Figures 4.34 and 4.35, the shape is very similar to the baseline, but the delta between the mean and the true value has increased to over 0.1. The variance has increased slightly, especially for the higher values of emissivity in the alabaster curve.

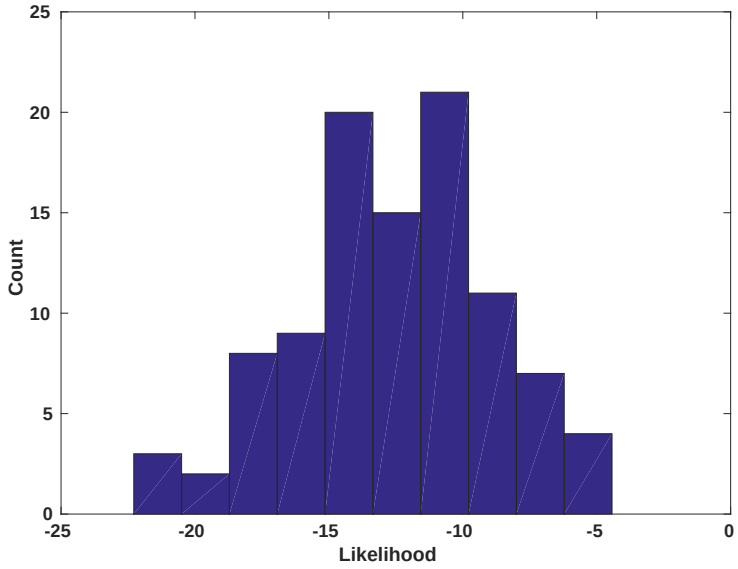

Fig. 4.30: Histogram of Maximum Likelihood for Baseline Parameters with 5 Observations.

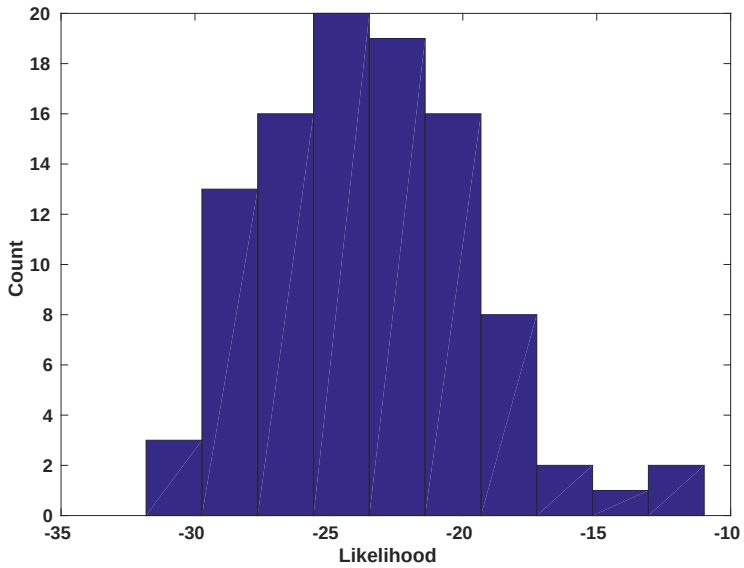

Fig. 4.31: Histogram of Maximum Likelihood for Baseline Parameters with Alabaster and 5 Observations.

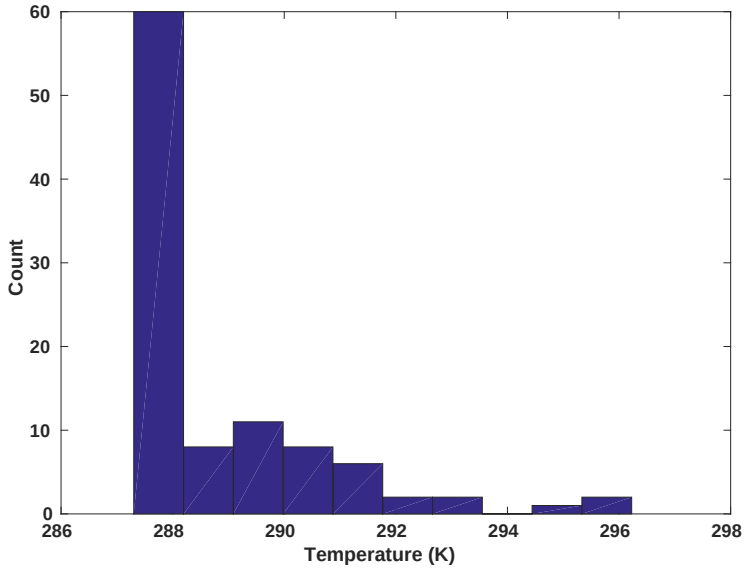

Fig. 4.32: Histogram of Estimated Temperature for Baseline Parameters with 5 Observations. Mean $= 289.02°$K, Standard Deviation $= 1.807°$K.

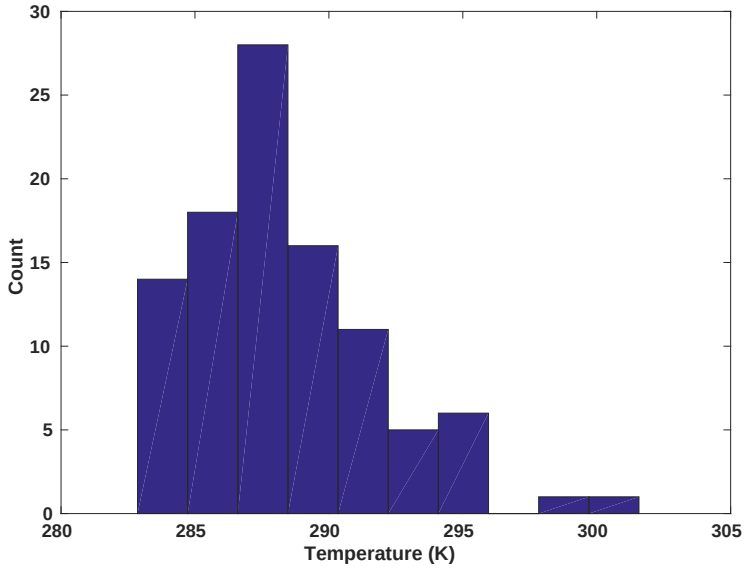

Fig. 4.33: Histogram of Estimated Temperature for Baseline Parameters with Alabaster and 5 Observations. Mean $= 288.55°$K, Standard Deviation $= 3.463°$K.

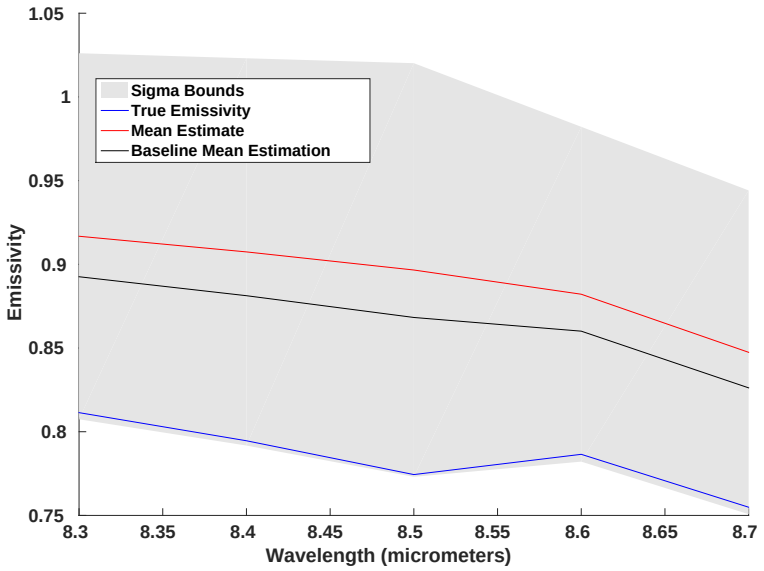

Fig. 4.34: Mean Estimated Emissivity with One-Sigma Bounds for Baseline Parameters with 5 Observations.

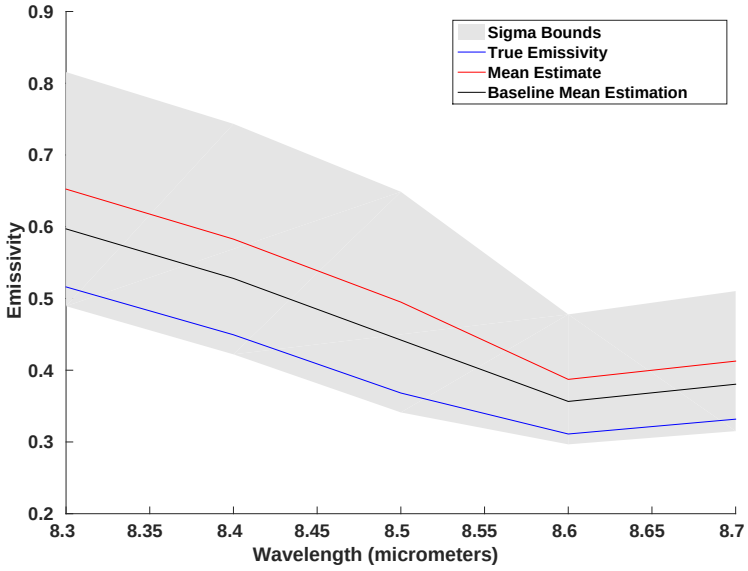

Fig. 4.35: Mean Estimated Emissivity with One-Sigma Bounds for Baseline Parameters with Alabaster and 5 Observations.

For convergence, the initial convergence is fairly typical, with the slate taking 7 iterations to reach temperature convergence and the alabaster taking 10. The ridge convergence for both is 2. The slate takes 17 to 607 iterations in eigenstep with a mean of 197. The shape is similar to the baseline, but the mean is shifted up due to an increase in the size of the group around 300. The alabaster is essentially identical to the baseline, except for a single outlier at 229. The spread is from 11 to 229 with a mean of 30.

For the case with 20 observations, the likelihood plots are shown in Figures 4.36 and 4.37. There is not much different in these distributions other than that the overall distribution seems shifted lower. For temperature, shown in Figures 4.38 and 4.39, there is also little change from the baseline, although there does seem to be a slightly tighter spread in the alabaster case, reducing to around $10°$K.

The emissivity distribution, shown in Figures 4.40 and 4.41 shows a decrease in both the mean and standard deviation of the estimate. For the slate, the mean delta is down to about 0.05 and the standard deviation is down to about 0.08. For the alabaster, the mean delta is down to about 0.025 with a standard deviation of about .06.

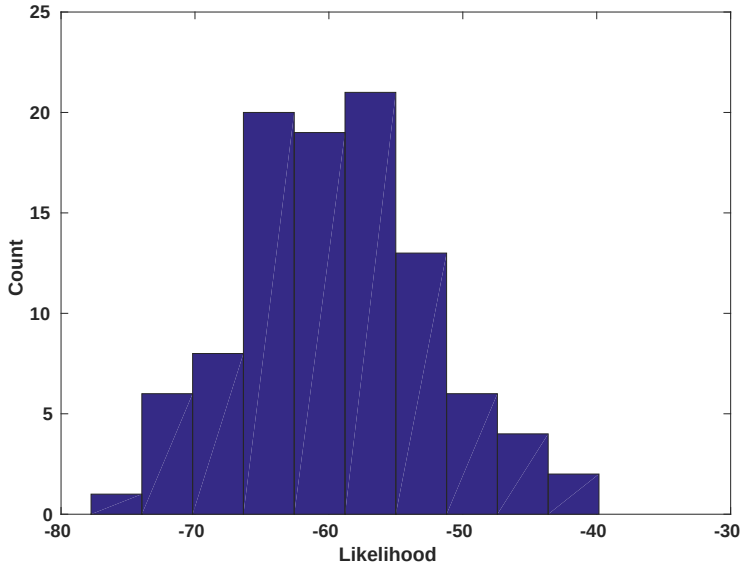

Fig. 4.36: Histogram of Maximum Likelihood for Baseline Parameters with 20 Observations.

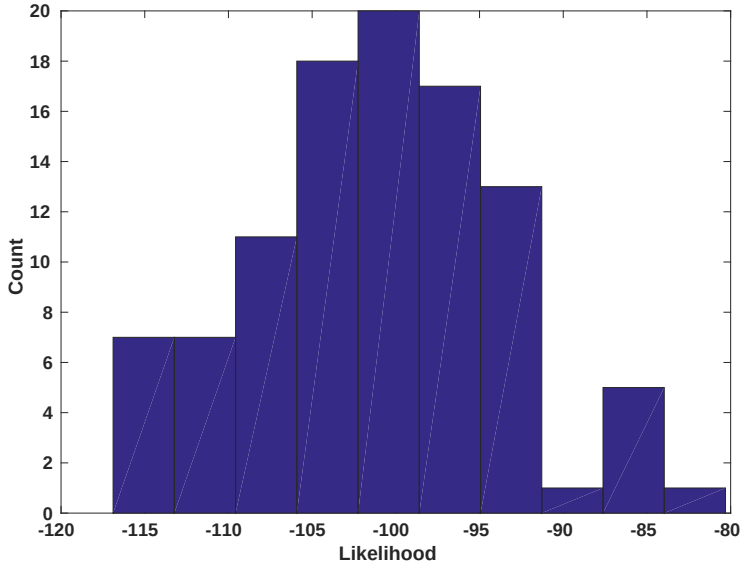

Fig. 4.37: Histogram of Maximum Likelihood for Baseline Parameters with Alabaster and 20 Observations.

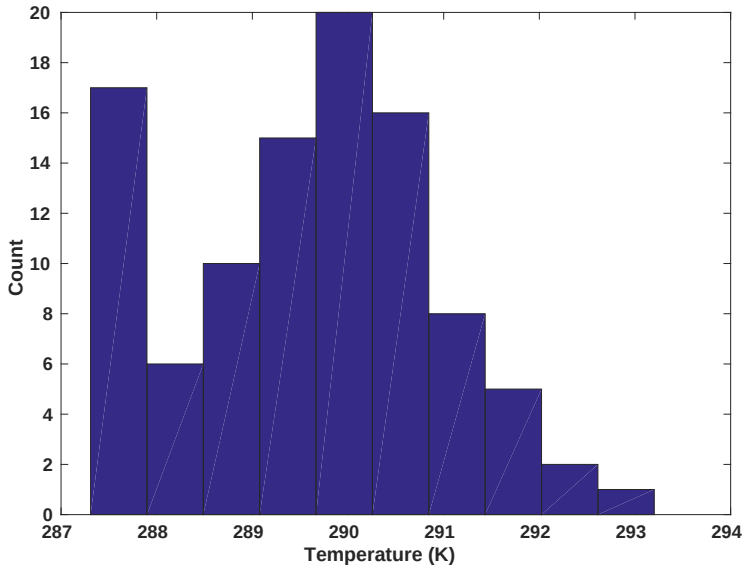

Fig. 4.38: Histogram of Estimated Temperature for Baseline Parameters with 20 Observations. Mean $= 289.67°$K, Standard Deviation $= 1.325°$K.

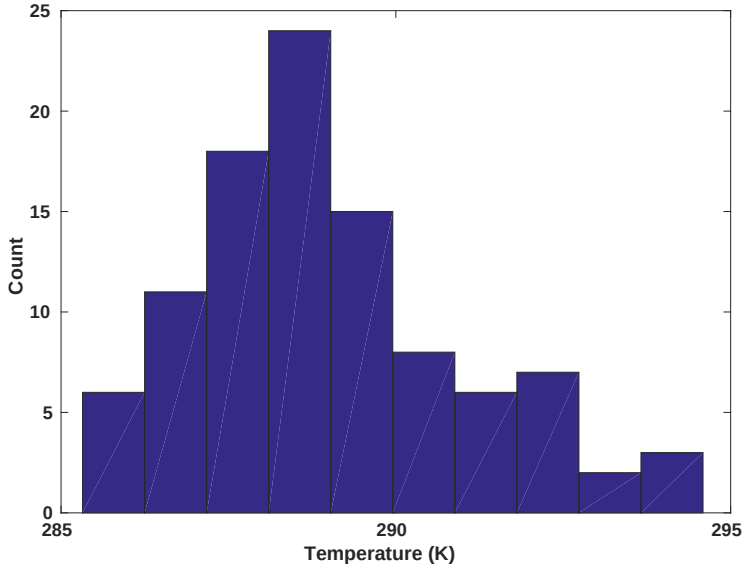

Fig. 4.39: Histogram of Estimated Temperature for Baseline Parameters with Alabaster and 20 Observations. Mean $= 289.15°$K, Standard Deviation $= 1.902°$K.

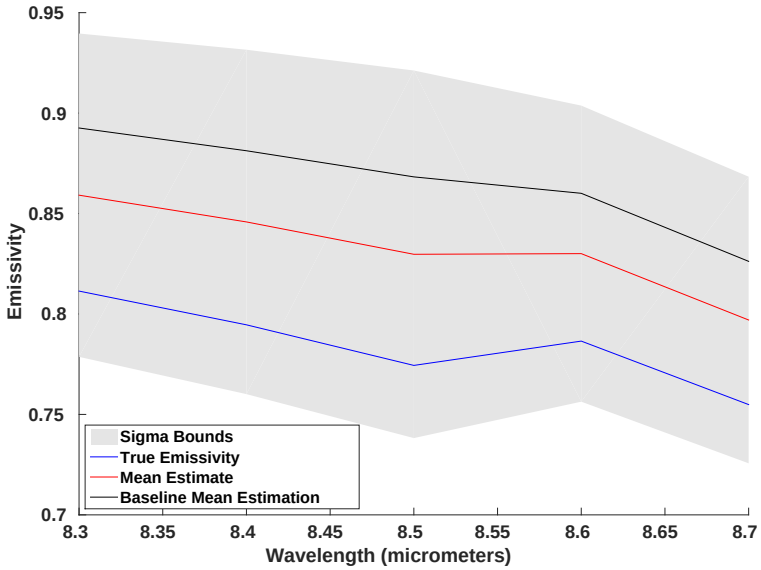

Fig. 4.40: Mean Estimated Emissivity with One-Sigma Bounds for Baseline Parameters with 20 Observations.

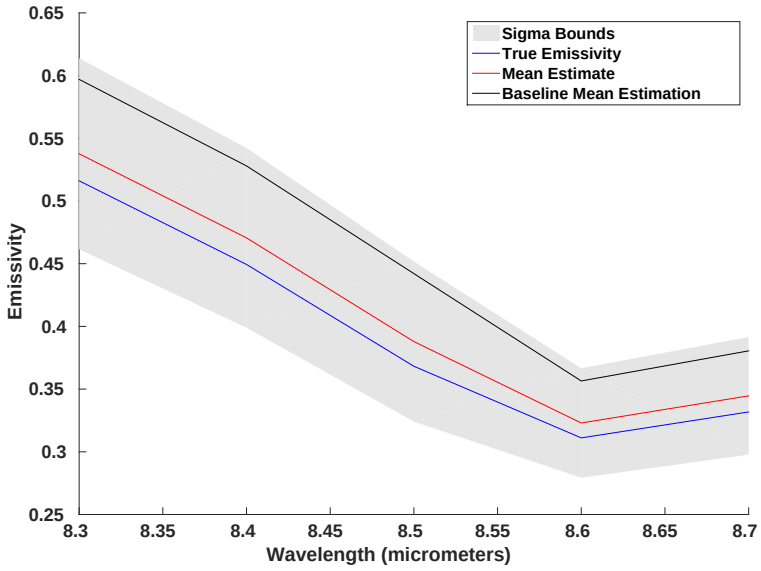

Fig. 4.41: Mean Estimated Emissivity with One-Sigma Bounds for Baseline Parameters with Alabaster and 20 Observations.

The convergence in temperature is slower for 20 observations, with slate taking 41 to 42 and alabaster taking 96 to 104. The ridge convergence is typical, taking 2 for slate and 4 for alabaster. The eigenstep convergence is actually somewhat improved, with the slate taking 17 to 562 iterations and an average of 103 and alabaster taking 10 to 55 with an average of 23.

Overall, there seems to be a relationship between the number of observations and the accuracy of the algorithm in an overall sense. Although it does not greatly impact the variance of the estimate, it does seem to improve the delta between the mean and the true value. There is some potential to reduce or remove any existing bias by increasing the number of observations. Generally speaking, increasing the number of observations improves accuracy as it averages out the noise variance on the observations, so it is reassuring that this algorithm performs this way.

The extra observations do seem to impact the initialization convergence, but the overall performance is the same or better, which is encouraging that more observations can be used to improve performance. Each iteration will run slower, due to the extra calculations required for more observations.

4.7 Varying Downwelling Radiance Variance

In varying the downwelling radiance, there is almost no appreciable change in the estimates or likelihood from the baseline. The only noticeable change was a slight increase in the standard deviation on the emissivity. This change is slightly more noticeable in the alabaster; see Figures 4.48 and 4.51. The figures are provided below, along with a summary of the convergence, which was slightly worse for slate and slightly better for alabaster.

For slate with the second variance, the results are shown in Figures 4.42, 4.43, and 4.44. The temperature convergence takes 9 steps, the ridge convergence takes 2, and the eigenstep convergence takes 16 to 729 iterations, with an average of 159. For alabaster with the second variance, the results are shown in Figures 4.48, 4.49, and 4.50. The temperature convergence takes 6 steps, the ridge convergence takes 3 to 4, and the eigenstep convergence takes 10 to 55 iterations, with an average of 24.

For slate with the third variance, the results are shown in Figures 4.45, 4.46, and 4.47. The

temperature convergence takes 13 steps, the ridge convergence takes 2, and the eigenstep convergence takes 16 to 789 iterations, with an average of 161. For alabaster with the third variance, the results are shown in Figures 4.51, 4.52, and 4.53. The temperature convergence takes 7 steps, the ridge convergence takes 3, and the eigenstep convergence takes 10 to 51 iterations, with an average of 26.

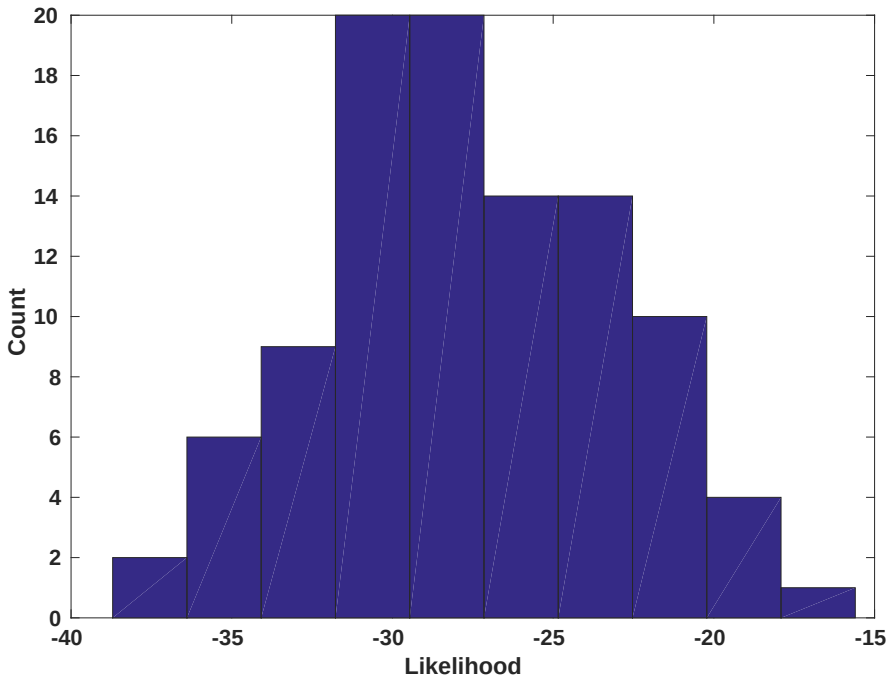

Fig. 4.42: Histogram of Maximum Likelihood for Baseline Parameters with Downwelling Variance Number 2.

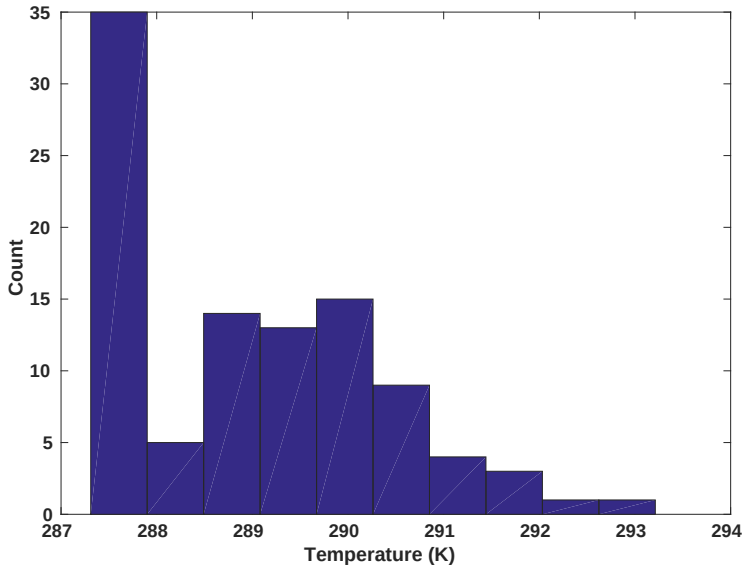

Fig. 4.43: Histogram of Estimated Temperature for Baseline Parameters with Downwelling Variance Number 2. Mean $= 289.02°$K, Standard Deviation $= 1.346°$K.

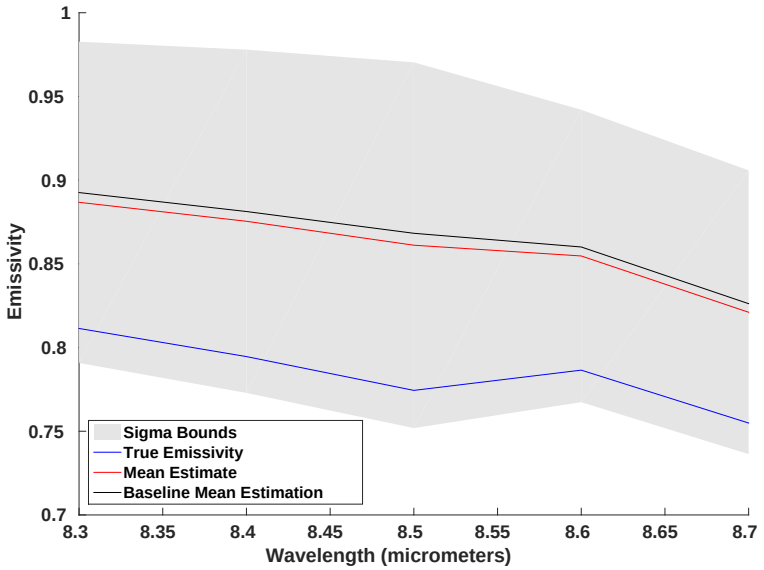

Fig. 4.44: Mean Estimated Emissivity with One-Sigma Bounds for Baseline Parameters with Downwelling Variance Number 2.

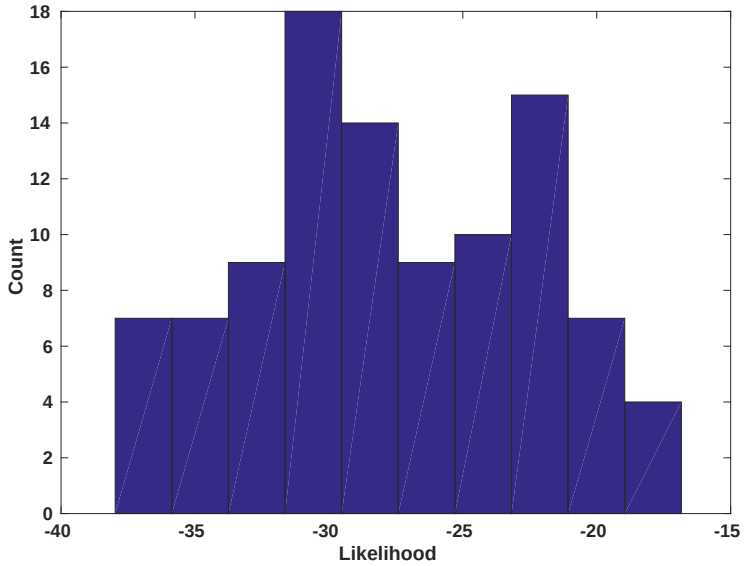

Fig. 4.45: Histogram of Maximum Likelihood for Baseline Parameters with Downwelling Variance Number 3.

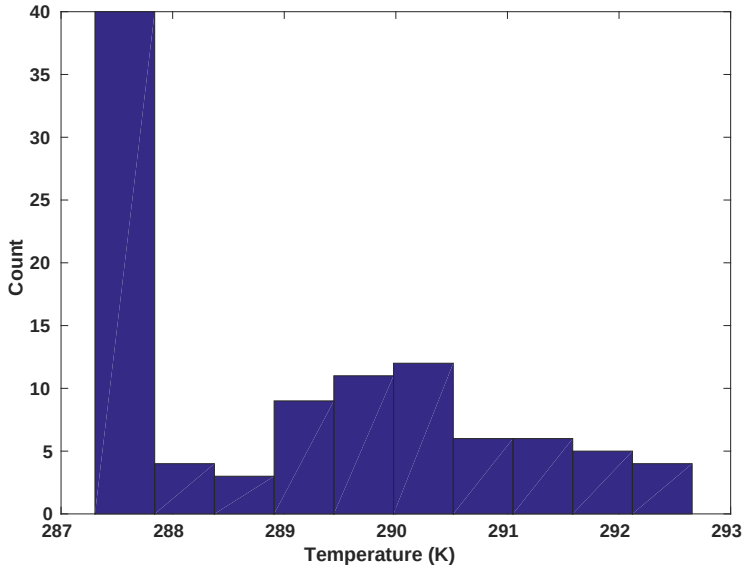

Fig. 4.46: Histogram of Estimated Temperature for Baseline Parameters with Downwelling Variance Number 3. Mean $= 289.16^{\circ}$K, Standard Deviation $= 1.602^{\circ}$K.

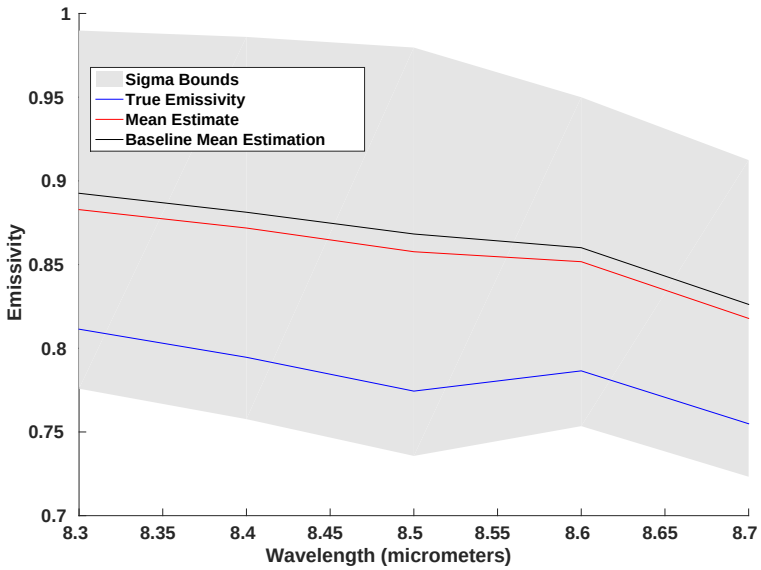

Fig. 4.47: Mean Estimated Emissivity with One-Sigma Bounds for Baseline Parameters with Downwelling Variance Number 3.

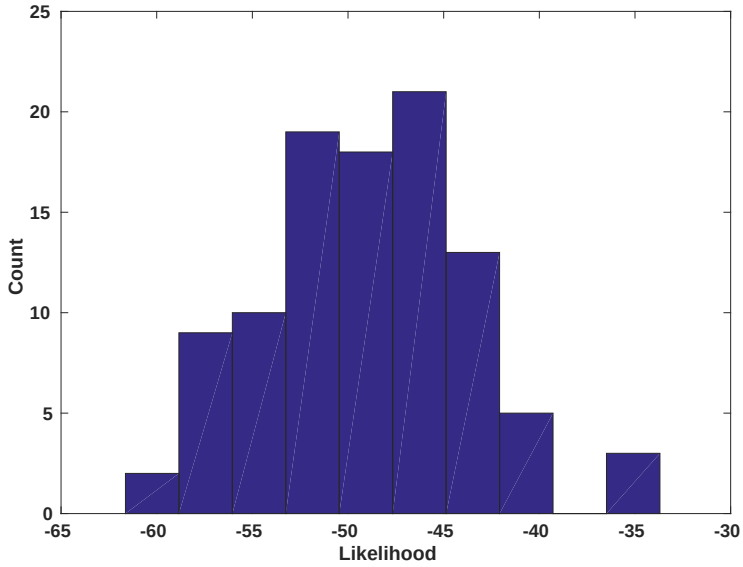

Fig. 4.48: Histogram of Maximum Likelihood for Baseline Parameters with Alabaster and Downwelling Variance Number 2.

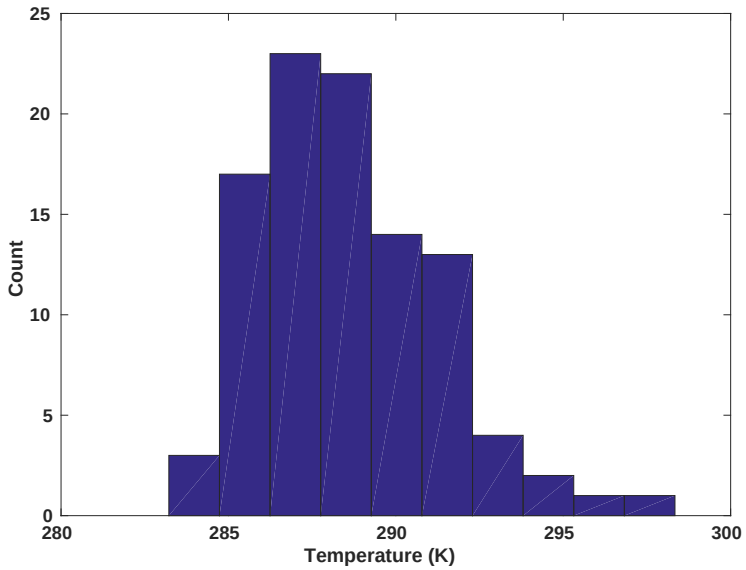

Fig. 4.49: Histogram of Estimated Temperature for Baseline Parameters with Alabaster and Downwelling Variance Number 2. Mean $= 288.91°$K, Standard Deviation $= 2.567°$K.

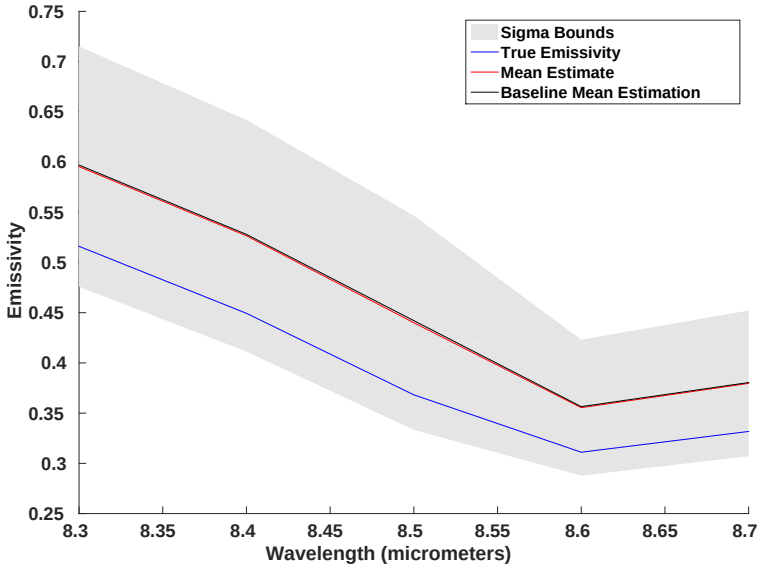

Fig. 4.50: Mean Estimated Emissivity with One-Sigma Bounds for Baseline Parameters with Alabaster and Downwelling Variance Number 2.

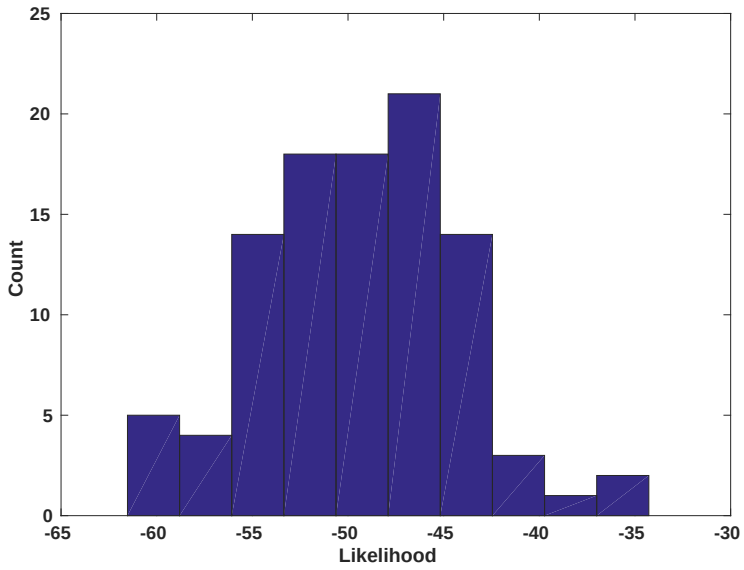

Fig. 4.51: Histogram of Maximum Likelihood for Baseline Parameters with Alabaster and Downwelling Variance Number 3.

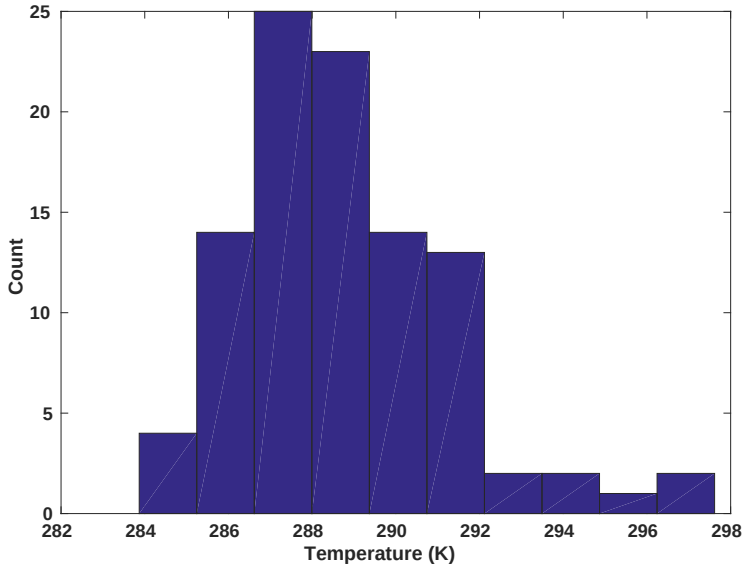

Fig. 4.52: Histogram of Estimated Temperature for Baseline Parameters with Alabaster and Downwelling Variance Number 3. Mean $= 288.85°$K, Standard Deviation $= 2.534°$K.

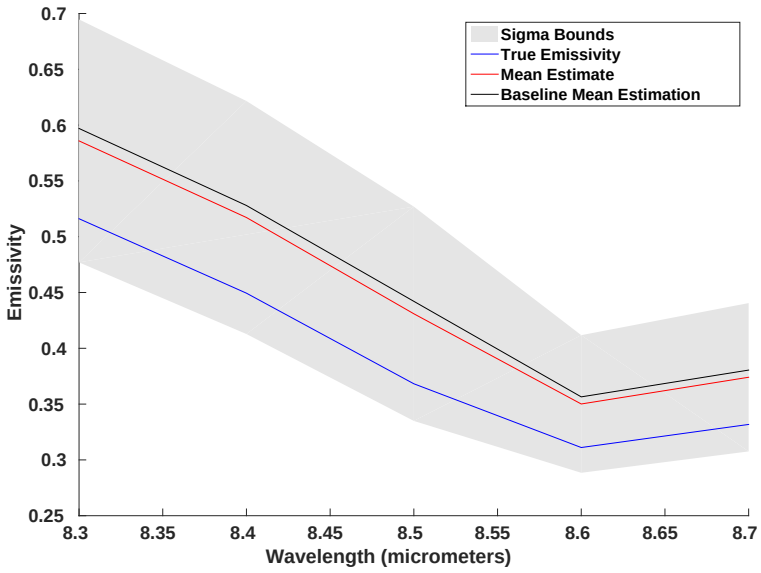

Fig. 4.53: Mean Estimated Emissivity with One-Sigma Bounds for Baseline Parameters with Alabaster and Downwelling Variance Number 3.

4.8 Varying Noise Variance

The final parameter to test is the noise variance on the observations. The lower noise variance of 0.5 was considered first. For this case, Figures 4.54 and 4.55 show that the likelihood is slightly shifted, but otherwise not significantly changed. The temperature distribution is shown in Figures 4.56 and 4.57. The reduction in noise variance seems to reduce the spread of the Temperature distributions slightly, with slate reducing from $6°$K to $5°$K and the alabaster reducing from $14°$K to $10°$K.

For the emissivity, shown in Figures 4.58 and 4.59, there is also some improvement in performance. In the slate, the improvement is slightly more pronounced, with a reduction in the mean error to about 0.05 and a reduction in the standard deviation to about 0.08. The alabaster does not show any significant change in mean delta, but the standard deviation does decrease to about 0.09.

For convergence, there is little change, although the average eigenstep convergence is slightly better than the baseline. The slate initialization convergence takes 21 steps in temperature and 2 steps to the ridge, while alabaster takes 9 and 4 to 5. The eigenstep convergence takes 15 to 705

iterations for slate, with an average of 88. The alabaster takes 10 to 59 with an average of 25.

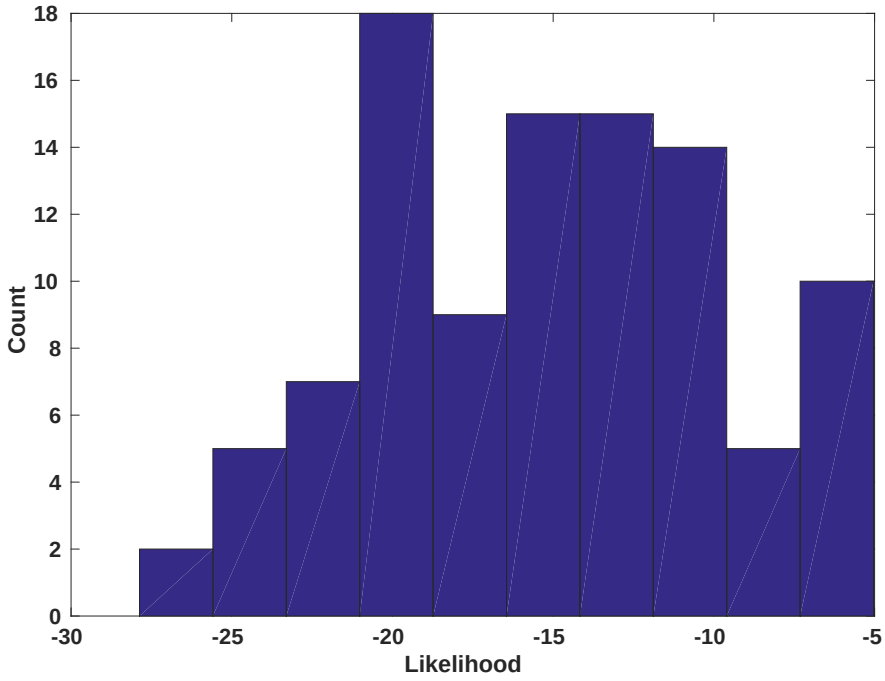

Fig. 4.54: Histogram of Maximum Likelihood for Baseline Parameters with Noise Variance $= 0.5$.

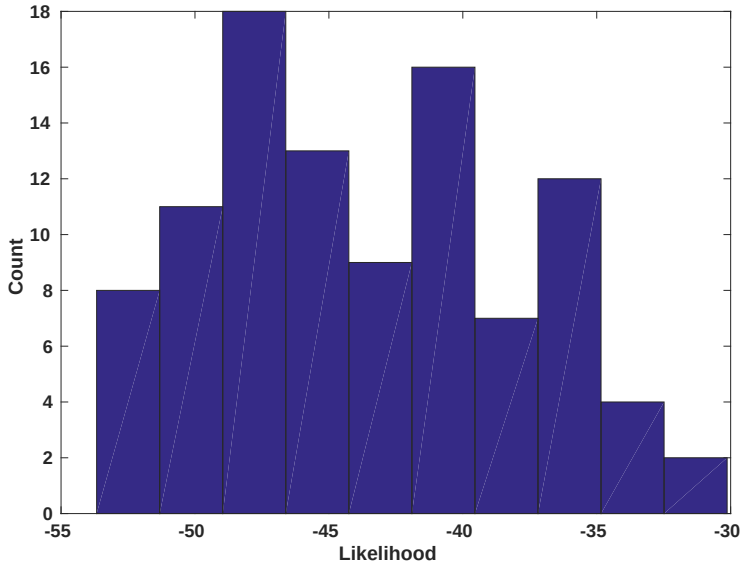

Fig. 4.55: Histogram of Maximum Likelihood for Baseline Parameters with Alabaster and Noise Variance $= 0.5$.

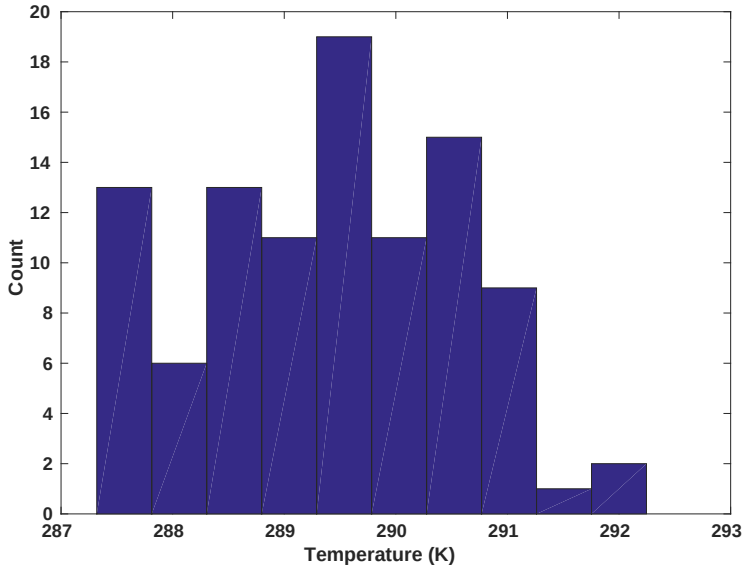

Fig. 4.56: Histogram of Estimated Temperature for Baseline Parameters with Noise Variance $= 0.5$. Mean $= 289.43°$K, Standard Deviation $= 1.164°$K.

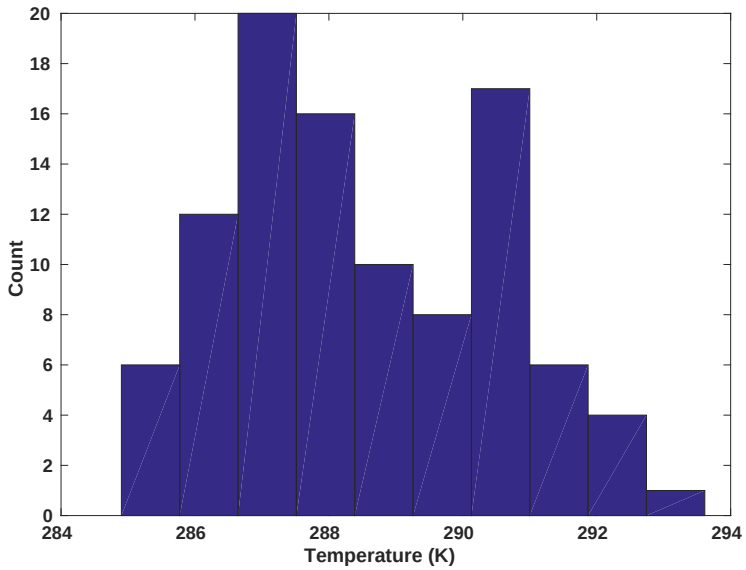

Fig. 4.57: Histogram of Estimated Temperature for Baseline Parameters with Alabaster and Noise Variance $= 0.5$. Mean $= 289.71°$K, Standard Deviation $= 1.433°$K.

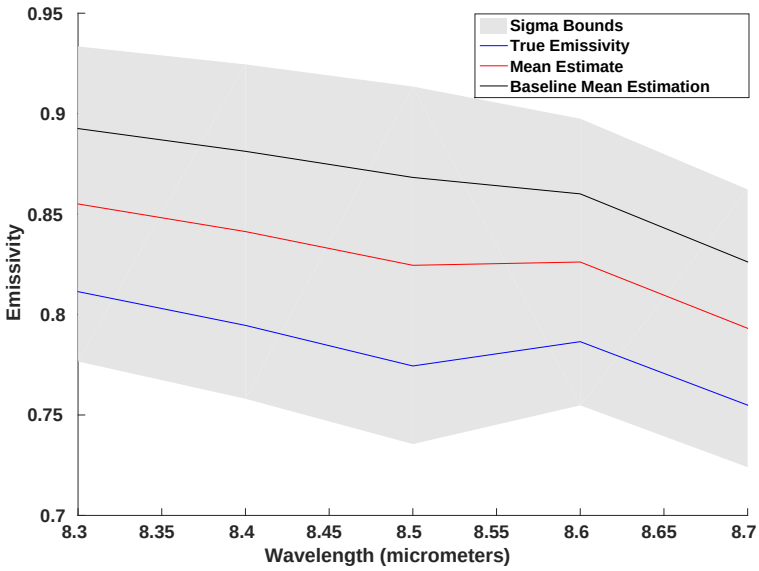

Fig. 4.58: Mean Estimated Emissivity with One-Sigma Bounds for Baseline Parameters with Noise Variance $= 0.5$.

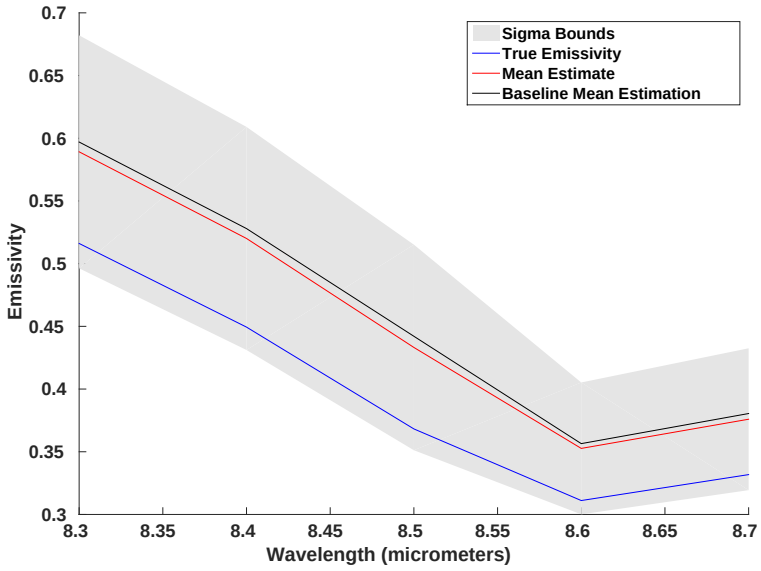

Fig. 4.59: Mean Estimated Emissivity with One-Sigma Bounds for Baseline Parameters with Alabaster and Noise Variance $= 0.5$.

Since the decrease in noise variance reduced the variance in the estimate, an increase in noise variance would be expected to do the opposite. The results bear this out.

Figures 4.60 and 4.61 show the likelihood is distributed the same, but shifted in the opposite direction. The temperatures, shown in Figure 4.62 and 4.63, show the same distribution, but with a wider spread of about $8°$K for slate and about $22°$K for alabaster. Corresponding to this loss of accuracy in temperature, the emissivity, which is shown in Figures 4.64 and 4.65, also is less accurate. The slate has a mean delta of about 0.07 and a standard deviation of 0.13, both higher than the baseline. Alabaster has similarly poor performance, with a mean delta of about 0.07 and standard deviation of 0.08.

The convergence is typical for initialization, with slate taking 4 steps to converge in temperature and 2 to find the ridge. Alabaster takes 6 and 3. The eigenstep convergence loses average performance, ranging from 12 to 715 (similar), but with an average of 197 iterations. Alabaster ranges from 10 to 189, which is a much larger range, and averages 32 iterations, which is slightly worse than baseline.

Overall, observation noise impacts the results as one would expect, increasing or decreasing estimation accuracy as noise decreases or increases, respectively.

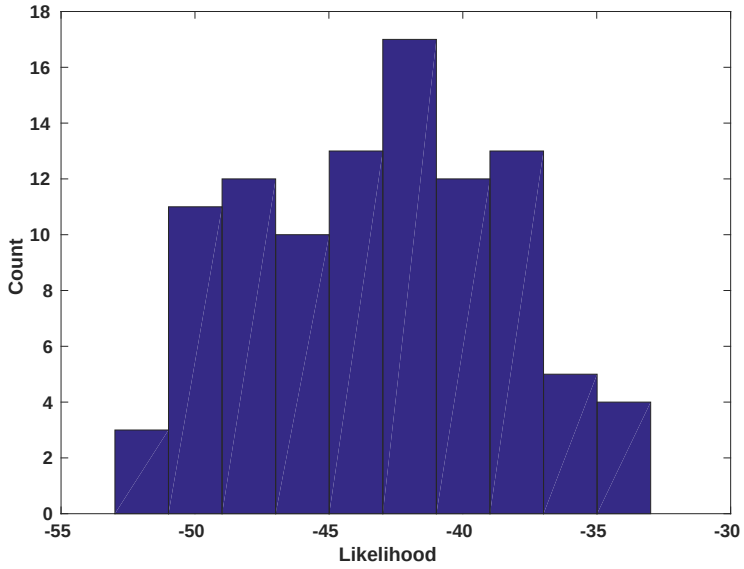

Fig. 4.60: Histogram of Maximum Likelihood for Baseline Parameters with Noise Variance $= 2$.

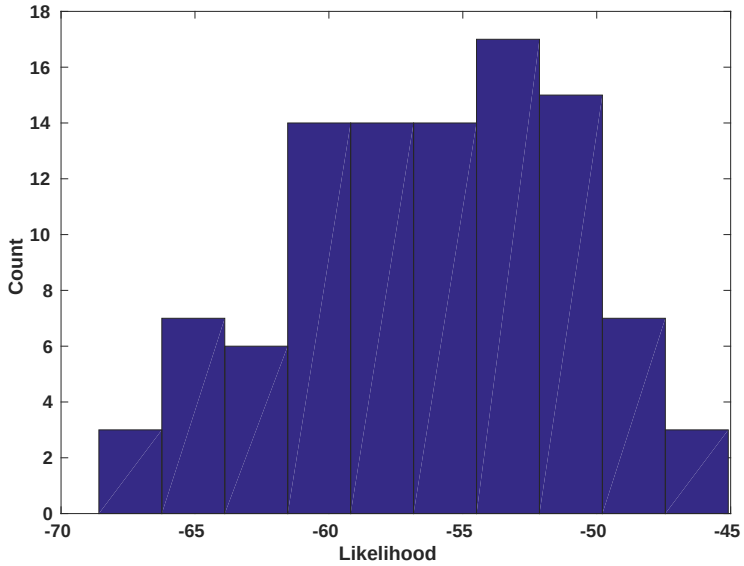

Fig. 4.61: Histogram of Maximum Likelihood for Baseline Parameters with Alabaster and Noise Variance $= 2$.

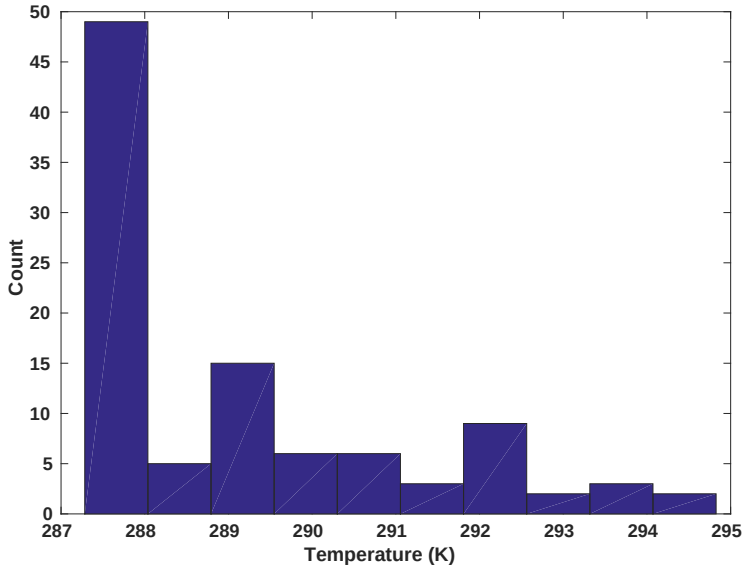

Fig. 4.62: Histogram of Estimated Temperature for Baseline Parameters with Noise Variance $= 2$. Mean $= 289.23°$K, Standard Deviation $= 1.948°$K.

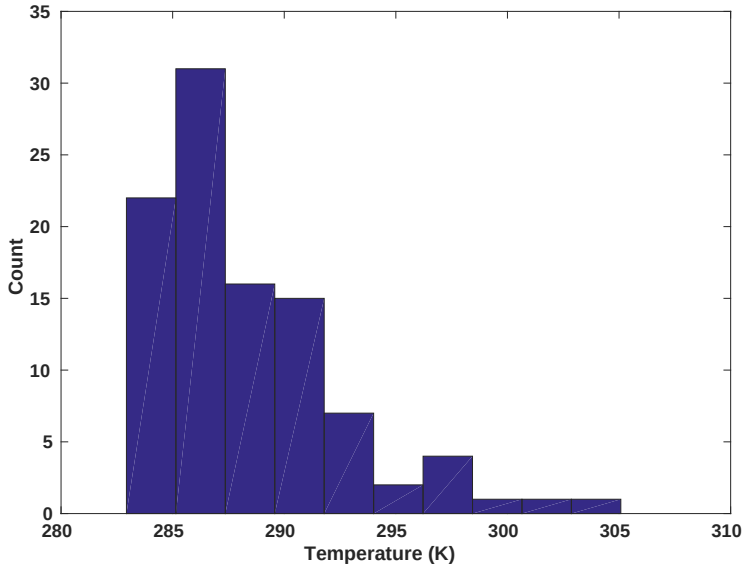

Fig. 4.63: Histogram of Estimated Temperature for Baseline Parameters with Alabaster and Noise Variance $= 2$. Mean $= 288.35°$K, Standard Deviation $= 4.242°$K.

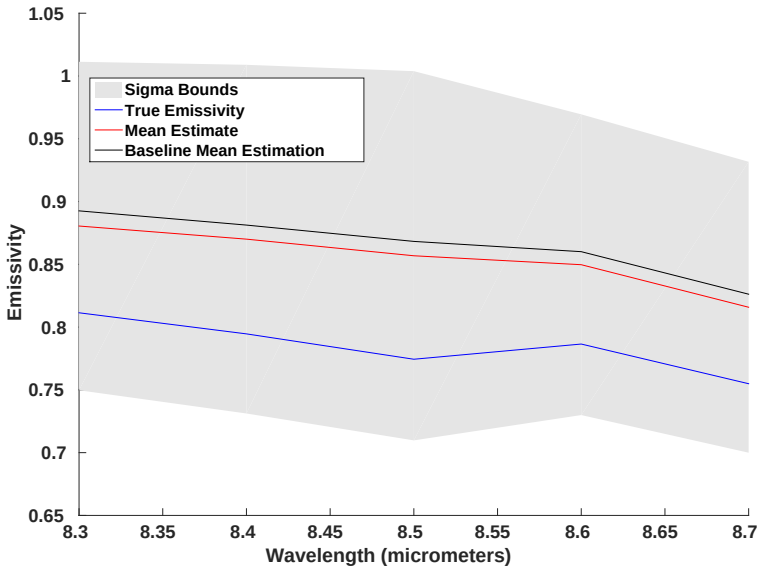

Fig. 4.64: Mean Estimated Emissivity with One-Sigma Bounds for Baseline Parameters with Noise Variance $= 2$.

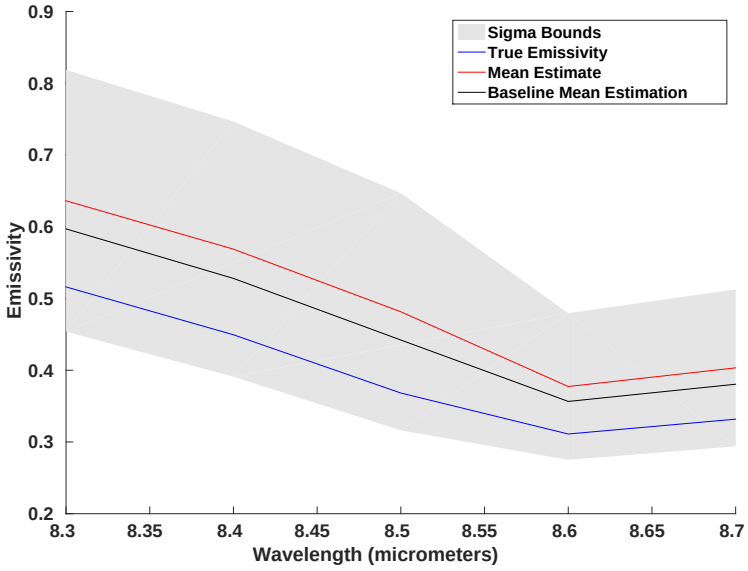

Fig. 4.65: Mean Estimated Emissivity with One-Sigma Bounds for Baseline Parameters with Alabaster and Noise Variance $= 2$.

4.9 Discussion

As a summary, the following characteristics were noted based on the results above:

- Different true emissivities and temperatures can impact both the accuracy and speed of the algorithm.
- As one would expect, the algorithm converges faster if the true emissivity is close to the initial guess.
- Convergence of the temperature initialization is affected by the number of parameters.
- Increasing the number of wavelengths decreases the estimate variance.
- Increasing the number of observations improves the accuracy of the mean estimate.
- We did not see significant change in the performance based on changing the number of correlated elements in the downwelling covariance.

- Higher noise variance reduced accuracy of the algorithm.

Based on these observations, there are a few points to make. With regards to faster convergence when the true emissivity is closer to the initial emissivity, our choice of initial emissivity was made to try and have an uninformed initialization. While maximum likelihood typically does not include prior knowledge, if we did have some information about the emissivity, it could be easily included by changing the initial values to speed convergence.

Regarding the number of parameters impacting temperature convergence, it is likely that the algorithm can be tuned in order to improve performance. It is also likely that choosing better initial temperature guesses would speed up performance. As discussed in the initialization setup for the algorithm in Chapter 3, the ambient temperature at the time of the measurements would probably be a good choice.

Next, while increasing the number of wavelengths and observations has the potential to increase accuracy, it also comes at a cost of speed. The more observations and wavelengths used, the more operations the algorithm needs. This is especially an issue with regards to the number of wavelengths, as the Hessian is $N + 1 \times N + 1$ (N being the number of wavelengths). The Hessian must be inverted and the eigen decomposition must be calculated. The matrix inversion can be especially expensive. This cost may still be worth paying, but it must be balanced against the desired performance.

Somewhat surprisingly, the impact of changing the downwelling variance was rather minimal. It is unclear if this is a general robustness in the algorithm or related to the fact that the downwelling was artificially chosen instead of estimated from MODTRAN or real data. The magnitude was not changed greatly. In the real data tests of Chapter 3, the estimated variance on the real data was very high and there seemed to be a larger amount of variance in the estimate than we saw in the other tests. It is not clear if the higher noise is due to observation noise or inaccuracy in the estimate of the downwelling radiance due to small sample size.

4.9.1 Improving Algorithm Performance

In varying each parameter separately, some insight has been gained into how changing that

parameter affects the estimate. For temperature and emissivity, the values in a scene cannot be changed, as they are fixed by the nature of the scene. Observation noise and downwelling variance are also beyond our control, although adjusting the estimates of them will likely affect performance. However, utilizing more wavelengths or observations can be tried in order to improve performance.

As an attempt to improve the performance beyond what as seen in the above tests, the algorithm will be run for a fixed true temperature, emissivity, noise variance, and downwelling variance, with adjustments to the number of observations in order to try to improve performance. The number of wavelengths will be set to 25, since this is the most that were able to be tested without modifying the algorithm. Some sample tests suggested that increasing the number of observations to 60 would provide significant improvement.

The results prove out the point. The likelihood spread, shown in Figure 4.66, is actually much larger, being over 100. Except for the increase in spread and the shift in value, it is much the same as the baseline.

The performance gain is easily seen in the temperature histogram, shown in Figure 4.67. The spread is down from 6°K to less than 2°K. Also, the distribution is more Gaussian, which would make sense, given the noise on the downwelling and observations.

In Figure 4.68, significant improvement is seen. The mean delta is down to about 0.005, more than a factor of 10 better than the baseline. The standard deviation is down to less than 0.02, about 5 times better than the baseline.

The convergence also provides some interesting results. The temperature convergence is again longer, taking 42 steps, which we expect, given the increase in number of wavelengths. The ridge convergence takes 4 steps. The most interesting part of the convergence is the eigenstep convergence, which takes from 27 to 76 iterations, with an average of 49. The convergence has actually improved with the increase in accuracy.

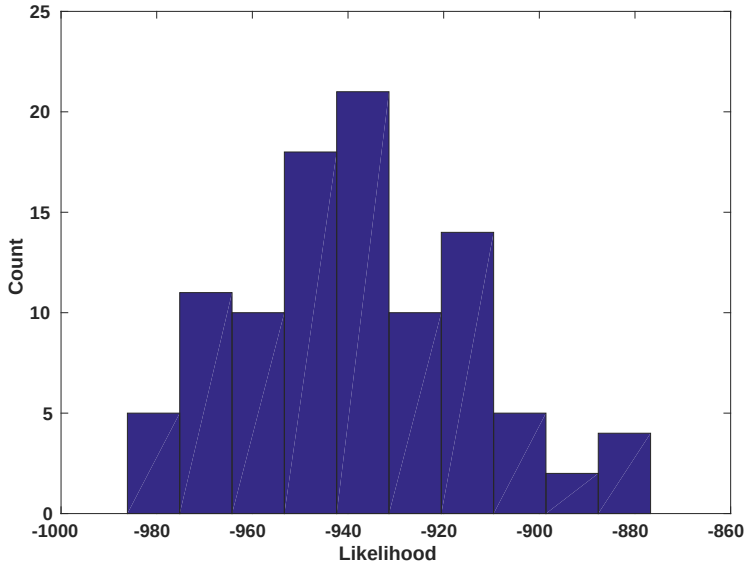

Fig. 4.66: Histogram of Maximum Likelihood for Baseline Parameters with 25 Wavelengths and 60 Observations.

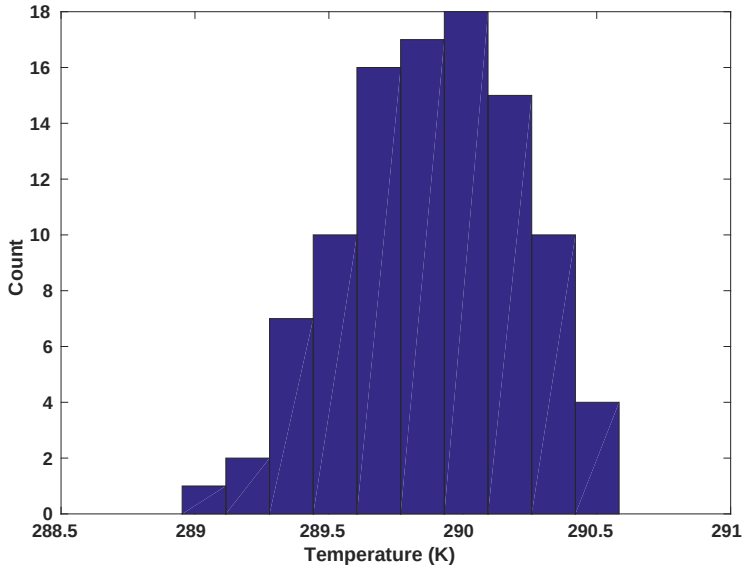

Fig. 4.67: Histogram of Estimated Temperature for Baseline Parameters with 25 Wavelengths and 60 Observations. Mean = 289.90°K, Standard Deviation = .330°K.

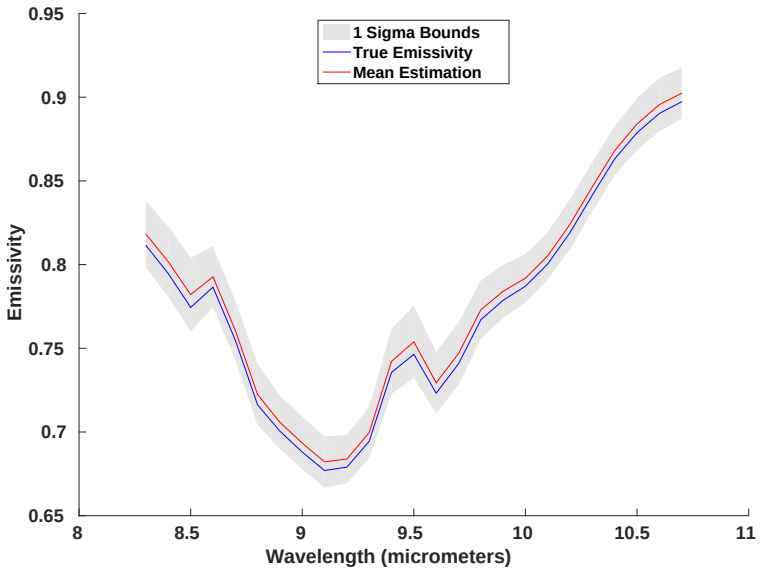

Fig. 4.68: Histogram of Estimated Emissivity for Baseline Parameters with 25 Wavelengths and 60 Observations.

For the alabaster, the likelihood plot, seen in Figure 4.69, also has a larger spread. The shape is consistent with other results. The temperature distribution, shown in Figure 4.70, is consistent in shape with previous results. However, it is within 0.2°K of the true value on average, which is much closer. Also, the variance is almost one-sixth of the baseline variance.

For the emissivity, which can be seen in Figure 4.71, there is also significant performance improvement. The performance is different for wavelengths less than about 9 μm than it is for higher wavelengths. For the lower wavelengths, the mean is about 0.005 with a standard deviation of about 0.015. For the higher wavelengths, the difference is about twice as much, coming in around 0.01, with a standard deviation of 0.02. Both of these are significantly better than the baseline performance.

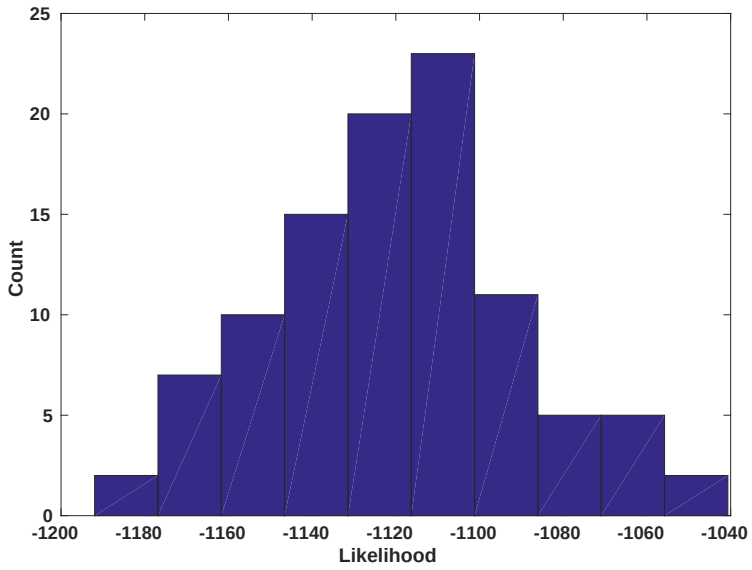

Fig. 4.69: Histogram of Maximum Likelihood for Baseline Parameters with Alabaster, 25 Wavelengths, and 60 Observations.

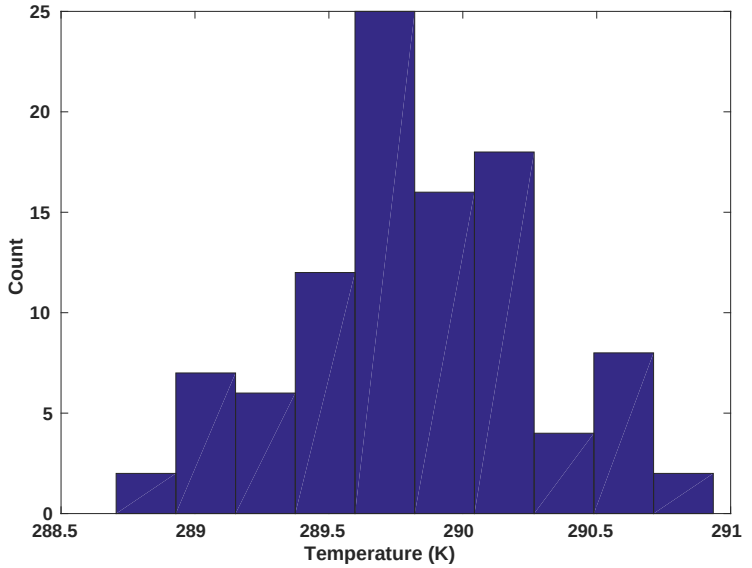

Fig. 4.70: Histogram of Estimated Temperature for Baseline Parameters with Alabaster, 25 Wavelengths, and 60 Observations. Mean $= 289.80^{\circ}$K, Standard Deviation $= .450^{\circ}$K.

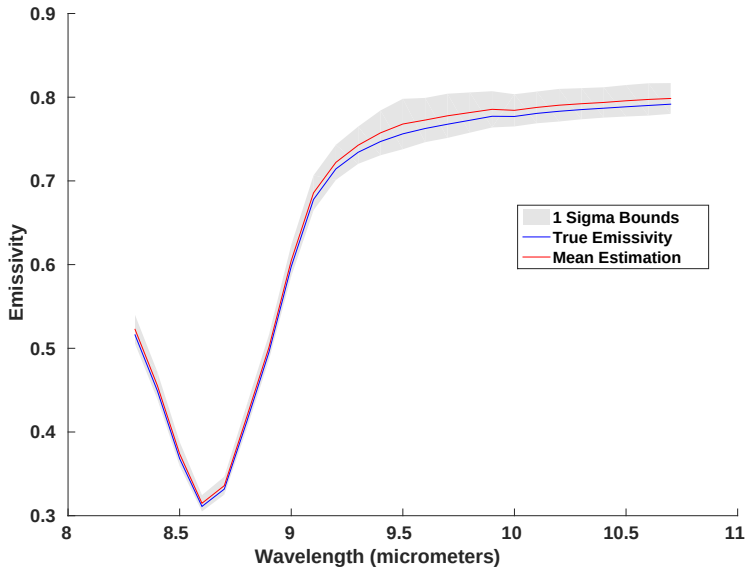

Fig. 4.71: Histogram of Estimated Emissivity for Baseline Parameters with Alabaster, 25 Wavelengths, and 60 Observations.

Convergence for the alabaster is typical, with slightly slower temperature initialization of 33 to 34 steps and ridge convergence of 5 steps. The eigenstep convergence takes between 24 and 78 iterations with an average of 41. The distribution is typical for the alabaster.

These results are consistent with the above tests. Combining an increase in the number of wavelengths and observations provides a significant improvement in performance.

4.10 Performance Comparison

The next step is to compare the performance of the algorithm to other approaches. As discussed in the introduction, this comparison will be against five algorithms that were selected from the literature.

As previously discussed, these algorithms contain a model that varies in a variety of ways. The most obvious of these, as was pointed out in the introduction, is that the downwelling radiance in the proposed model is considered to be Gaussian distributed.

Now that an algorithm has been developed to solve the ML problem posed earlier, the actual algorithms from the literature can be compared more directly. Along with the comparisons, the

performance of the algorithms is compared. Each author presents difference performance metrics for different materials and under different conditions. Between these differences, algorithmic differences, and model differences, there really is no expectation of a direct comparison. However, it is useful to see what kind of performance can be obtained with each algorithm.

4.10.1 Iterative Spectrally Smooth TES

As previously discussed, the ISSTES algorithm generates a family of emissivity curves that satisfy (1.5). A spectral smoothness measure is then used to select the curve that is considered the best emissivity estimate. The temperature that corresponds to this curve is considered the best temperature estimate. The algorithm makes use of MODTRAN to generate the unknown atmospheric parameters [12].

As discussed in the introduction, the optimality criterion for ISSTES is very different than in the eigenstep algorithm, so the estimates should be quite different. In addition, each algorithm has a number of different characteristics which affect the relative performance. It would be expected that the choice of temperatures for which the emissivity curves are estimated would introduce some error into the algorithm compared with the eigenstep algorithm. The use of MODTRAN in ISSTES compared with some other estimation technique to remove atmospheric attenuation and upwelling radiance provides some variation, but is probably roughly equivalent. The MODTRAN estimate of the downwelling radiance, however, will fix a certain amount of error in the ISSTES estimate. In contrast, the Gaussian model used for the eigenstep algorithm allows for an estimate that better uses the measured data. Additionally, the iterative nature of the eigenstep algorithm allows for more refinement of the estimates based on the observations, which should help improve the answer over ISSTES's direct estimation. The total number of wavelengths used in ISSTES is not defined, but appears to be large and closer to the 25 used in the increased performance tests for the eigenstep algorithm, rather than the 5 used in the baseline tests. The ISSTES algorithm does not make use of multiple observations, which should degrade the performance.

The actual performance of the ISSTES algorithm is reported as a function of altitude. Since the eigenstep algorithm assumes that the hyperspectral sensor is close to the target, only the lowest

altitude results for ISSTES are compared.

The worst case estimated emissivity bias is reported as 0.086, with a corresponding temperature bias of 3.43° K. Generally, however, performance was much better. An emissivity bias of 0.0008 was reported for the sensor near the surface. The corresponding temperature error for this case appears to be less than 0.05° K [12].

The worst case appears to be worse than the mean bias for the 25 wavelength and 5 wavelength case, which was about 0.05 for emissivity and about 2° K for temperature, which are seen in Figures 4.24 and 4.25. However, the typical performance for ISSTES at the surface appears to be better than the eigenstep algorithm.

A caveat is the difficulty in interpreting the results shown in the ISSTES paper versus the eigenstep results reported above. It is not clear how much noise would be present on the true data in the ISSTES measurements. It could be quite different from the noise added in the synthesized data for the eigenstep algorithm. Also, the actual results reported on ISSTES are not presented in a way that provides for an easy comparison. Additionally, the choice of material for ISSTES, which, as seen in the comparison between slate and alabaster above, results in some variation in performance.

Interestingly, the results for ISSTES show that both the mean bias and standard deviation of the bias for temperature and emissivity increase with altitude. This supports the assertion that use of MODTRAN data to estimate the atmospheric properties does impact the algorithm performance.

4.10.2 Linearized Iterative Convex Optimization

The linearized iterative convex optimization takes the detailed model used for ISSTES and removes the atmospheric attenuation and upwelling radiance, either through proximity of the sensor or some other estimation technique. The algorithm linearizes the Planck function and uses a Taylor series approximation to create a convex function. An initial point is chosen, then the convex function is iteratively solved for emissivity, downwelling radiance, and temperature, with the new estimates replacing the initial point until convergence [18].

Since this algorithm removes the atmospheric attenuation and upwelling radiance in the same way as the eigenstep algorithm, there is at least a starting point for similarity. However, the lin-

earization of the Planck function and the Taylor series approximation introduce error not present in the eigenstep approach. Additionally, the convex iteration provides a very different optimal condition, as the true function is not generally convex. The algorithm does use multiple observations, which provides an additional point of similarity. In the tests that were run, 120 wavelengths were tested with 10 observations [18]. This should provide some performance improvements over the eigenstep algorithm, as only 25 wavelengths were tested at most.

The actual performance was not evaluated in a quantitative way. Rather, curves are shown comparing the results to library values. Usefully, the algorithm was applied to both alabaster and slate. Only a single curve was reported, making direct comparison of average performance difficult. The plots for alabaster and slate are reproduced from the original paper in Figures 4.72 and 4.73, respectively [18]. While some elements of the true shape are shown, there is significant variance in the shape alone. Even the additional wavelengths do not seem to add much to the performance. Based on this limited comparison, it appears that the eigenstep algorithm drastically improves over the linearized iterative convex optimization approach.

4.10.3 TES Retrieval Using Linear Spectral Emissivity Constraint

In this approach, the model is the same as used in the eigenstep algorithm. Atmospheric correction has been applied accurately to remove upwelling radiance and atmospheric attenuation. If there are $N + 1$ wavelengths, the measurements are broken into N groups, where a pair of measurements at two adjacent wavelengths are considered to form a linear approximation of the value between them. A cost function is defined as the sum of the square of the residual error between the actual and estimated at-ground radiance. The downwelling radiance is estimated using MODTRAN [24].

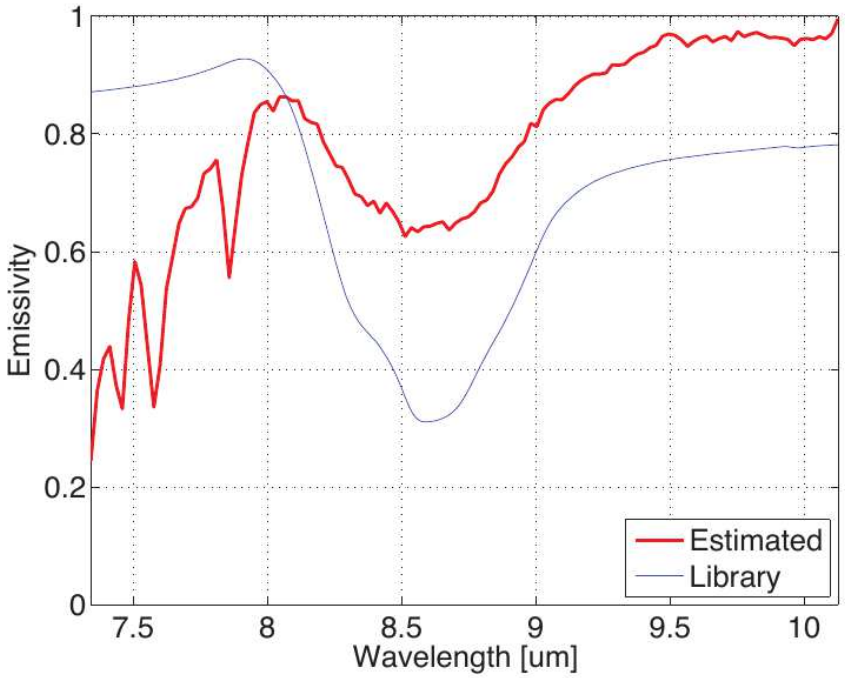

Fig. 4.72: Linearized Iterative Convex Optimization Performance for Alabaster.

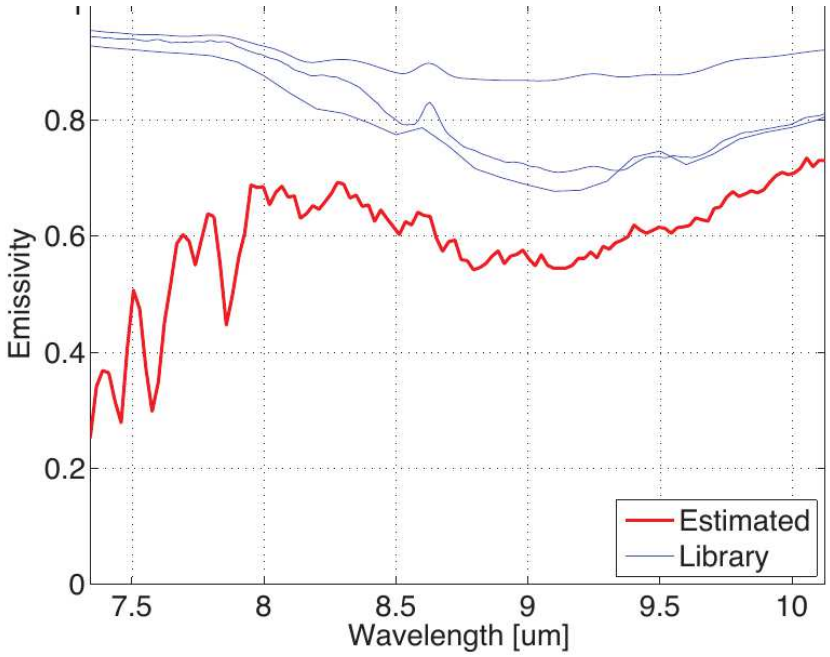

Fig. 4.73: Linearized Iterative Convex Optimization Performance for Slate.

An initial value for temperature, such as the surface brightness temperature, is found and used to estimate the emissivity spectrum, typically through a technique such as least squares. The cost function is then used to find an incremental step in the surface temperature. A gradient step approach is used. If the step is small enough, the algorithm terminates. Otherwise, the algorithm begins again, using the adjusted temperature to find the next emissivity spectrum [24].

This approach has some useful comparisons to the eigenstep function, as it uses the same general model and atmospheric compensation assumptions. As in other approaches, it does not model the downwelling radiance as a Gaussian random variable. Similar performance effects due to the use of MODTRAN will exist. The use of a cost function and an approach that seems similar to gradient ascent is also similar. However, the cost function is the mean-squared error, which will result in a different optimum from the ML approach used in the eigenstep algorithm. Also, the linear approximation between wavelengths will result in some differences, especially since it is not necessarily accurate. This problem will be more pronounced if the wavelengths in the spectrum are spaced far apart.

The actual performance of this algorithm is rather good, but, as noted in the tests of the eigenstep algorithm, varies with the material. A slate sample was tested and demonstrated performance similar to the eigenstep algorithm, at least for the given setup. The average mean error across the entire spectrum appears to be less than 0.001, with a standard deviation around 0.006. These values are better than the best performance obtained for slate in the 25 wavelength and 60 observation case. The approach used significantly more wavelengths, 100, and it is unclear how the noise might compare, but the performance was still very good. The specific performance for temperature of the slate is not clearly reported [24].

General results across all the materials tested varied more than in the slate. Some of the materials varied by almost 0.02 in mean and the worst case standard deviation was also about 0.02. Most of the estimated emissivity curves displayed similar characteristics to those in the eigenstep tests - the shape was generally correct, even if there was a mean shift. However, the temperature estimation seemed to vary slightly more. The mean accuracy was within 1° K for 80 percent of the samples [24].

Interestingly, materials that were less smooth, such as water and polystyrene, followed the emissivity curves less closely and had poorer performance [24]. One would expect that this is due to the linear approximation between wavelengths. Materials with emissivity curves that are not smooth would not fit this assumption well and would the estimates would expectedly be worse.

4.10.4 Stepwise Refining of TES

For stepwise refining of TES, the algorithm focuses on considering small groups of wavelengths that are close together. The foundation is the assumption that the true emissivity over a small change in wavelength should be linear. Groups of three wavelengths are chosen near strong atmospheric emission lines. The temperatures are estimated for these six locations. The average of these six temperatures is used as the temperature for the estimated surface temperature [34].

Top find the emissivity spectrum, emissivities are chosen with 0.1 intervals for each set of three wavenumbers . These values are arbitrarily chosen and are not expected to be initially accurate. The initial values are used to estimate the surface self-emission. The optimal estimate, by the linearity assumption, will be a straight line across the three wavelengths. The algorithm selects the estimate where the middle wavelength varies the least from this line. The algorithm continues by reducing the interval size to 0.01 and focuses around the selected estimate. This is repeated for step sizes of 0.001 and 0.0001. Within the limits of the assumptions and model, this should provide accuracy of within 0.0001 for the emissivity spectrum [34].

As with previous algorithms, the linearity assumption introduces some bias into the problem. Also, the use of specific features, such as the strong atmospheric emission lines, introduces error into the estimate. While the true value may be estimated with some accuracy at these points, noise can provide a shift, which is then introduced into all the estimates. The averaging likely reduces this, but six points only provides a limited reduction in noise effects. Essentially, there are quite a few factors that indicate that the results may vary significantly from the eigenstep approach. These factors also include the typical difference from all the other algorithms that the downwelling radiance is not modeled as a Gaussian random variable.

The actual performance shows that the temperature estimate is generally good. The majority

of estimates fall within about 0.3 degrees of the true value. However, a number of results fall much farther away and, in some cases, can be biased up to 5 degrees [34].

The emissivity estimate has a root mean square error of less than about 0.025, although this gets worse near the band edges (about 700 and 1300 cm^{-1}). This is likely due to changes in the model accuracy near the band edges. The model does not appear to be particularly sensitive to the accuracy of the temperature, which is very different from what happens with most of the other algorithms [34]. While this is good if the emissivity is the main focus, it does present problems in identifying the overall quality of the temperature-emissivity estimate. One estimate may be good, while the other very incorrect. This suggests that there is some sort of disconnect between the actual model and the assumptions in the algorithm. It is not that this will actually degrade performance of the algorithm.

The algorithm is specifically compared to ISSTES, which was reviewed above. In every case, this algorithm appears to perform better than ISSTES [34]. At the very least, this suggests that this algorithm does provide good performance. In comparison with the eigenstep algorithm, this algorithm appears to provide some good and bad advantages. The temperature estimate, on average, appears to have similar performance. However, the outliers for this algorithm are much worse than the eigenstep algorithm which, for 25 wavelengths and 60 observations, does not exhibit any tests with a bias greater than about 1 degree. The emissivity estimate seems to be slightly better than the eigenstep algorithm, although there is no root mean square error for direct comparison. The eigenstep algorithm does perform better in that its error does not seem to vary at the band edges, although performance is impacted by the magnitude of the true emissivity.

4.10.5 MODTES

In MODTES, atmospheric correction is performed by estimating the upwelling radiance and atmospheric attenuation using MODTRAN. The algorithm then performs TES on the same model as in the eigenstep algorithm. The MODTRAN estimates are only applied to graybody pixels, which are identified by their normalized difference vegetation index. Water vapor profiles are used to estimate the brightness temperature of these pixels and this estimate of temperature is used to

refine the MODTRAN-estimated upwelling radiance and atmospheric attenuation. These refined estimates are used to correct for atmosphere for all pixels. For graybody pixels, the values are applied directly; for non-graybody pixels, the surrounding graybody pixels are used to interpolate atmospheric correction values [38].

Next, the algorithm sets the emissivity for all graybody pixels to 0.985 and for non-graybody pixels to 0.980. These values are used with the downwelling radiance calculated in MODTRAN to provide an initial estimate of the temperature. The maximum value of those temperatures is selected as a temperature for all pixels and a new emissivity estimate for each pixels is calculated [38].

As part of the algorithm, three specific wavelengths were selected as a reference point and are used to calculate a ratio of three times the emissivity of each wavelength to the sum of the emissivities at each of the three specific wavelengths. The difference between the maximum and minimum values of this ratio across the image is used to estimate the minimum emissivity in the image. The specific relationship is empirical and the coefficients were estimated by regression techniques. The true emissivities are estimated by a relationship between the emissivity ratios and the minimum emissivity. Finally, the temperature is estimated from the estimated emissivity spectrum [38].

As with the other algorithms, this approach has a number of features that will result in variations from the eigenstep algorithm. The relationships between the different values calculated are unique to this algorithm. Also, as in other algorithms, this approach does not model the downwelling radiance as a Gaussian random variable. Instead, this approach uses MODTRAN and will have similar performance differences as in previously discussed for other algorithms. While these differences exist, it is difficult to quantify how they will impact the results, as the relationships between the emissivity ratio and the estimated minimum emissivity have an unclear impact on the performance. Further, the selection of three specific wavelengths will impact the performance depending. No specific justification is given for this assumption. As such, the results themselves speak for the accuracy of this algorithm.

The performance of the results are reported with a variety of variations to the algorithm and different types of error calculations. The mean temperature bias for the base algorithm is 0.5 degrees with a standard deviation of 0.4 degrees [38]. This compares favorably with the performance of the

eigenstep algorithm, falling about halfway between the performance in the baseline cases and the improved algorithm (25 wavelengths and 60 observations case).

The emissivity had a mean bias of 0.016 with a standard deviation of 0.007 [38]. The mean is again somewhere between the baseline and improved eigenstep performance. However, the standard deviation seems somewhat better than the performance for the eigenstep algorithm.

4.10.6 Performance Comparison Conclusions

In reviewing all of these other TES algorithms, the main and most obvious conclusion is that it is difficult to compare performance on even footing. The assumptions, formulations, and optimality criteria vary for each algorithm. Even directly comparing results can be difficult due to the differences in materials tested or simply how the results are reported.

In spite of this, it is clear that the performance of the eigenstep algorithm is in family with the body of the literature. Some algorithms have similar performance; some are less accurate. While some have greater accuracy, for any of the algorithms, it is not entirely clear if the improved performance is inherent in the algorithm or due to variations in noise, initial values, or other assumptions. For example, the eigenstep algorithm starts from a constant initial condition that is essentially unbiased. Only the data informs the estimates. Almost every other algorithm requires some specific initialization. Due to the difference in the algorithm formulations, it may not be possible to even put them on even footing. As such, it can be said that the eigenstep algorithm is in family and an improvement on some of the approaches addressed herein.

Further, the algorithm has yet to be optimized. The correlation between wavelengths could be more accurately modeled. The algorithm could also possibly be modified to run even faster than it already runs. And, as seen above, increasing the number of wavelengths and observations improves the performance. Additional testing could show that this provides significant performance improvement over what has been tested here. In short, this brand-new algorithm is already comparable to other approaches and there are obvious avenues for improving the performance.

4.11 Conclusions

Based on the results in this chapter, changes in the parameters that affect the performance of the

algorithm have been identified. The algorithm converges consistently, with some fluctuations due to parameter changes. Highly accurate performance can be achieved by sacrificing computational efficiency. The performance is similar to other approaches reviewed, even in the early stages of this algorithm's development.

Based on the performance change with regards to observations, it does appear that some level of bias exists which is a function of the number of observations. This is consistent with expected performance of ML estimators, which typically approach the true value in mean as the number of observations approaches infinity.

CHAPTER 5
BIAS ESTIMATION AND CORRECTION

5.1 Introduction

As identified in Chapter 4, for cases where there are few observations, a significant bias may be present in the mean estimated emissivity and temperature. This bias may be removed by obtaining more observations, but this is not always possible. Methods also exist for estimating a bias and removing it from the estimator.

Early on in the algorithm development, a number of different approaches were tested. These were derived from the standard approaches described in Chapter 3. In most cases, these algorithms contained significant bias. Further, it was not clear at this point that increasing the number of observations or any other potential changes would actually improve the performance of those algorithms.

Based on those estimators, a general approach for estimating the bias and removing it was developed. This was originally presented in [51]. The results are presented here, which begin from the model developed in Chapter 2.

At the time this bias correction was developed, a direct solution was being calculated for each emissivity, holding all other wavelengths constant. The temperature could not be directly estimated, so gradient ascent was to be used to estimate the temperature based on the current emissivity estimate. This was in turn fed back in a recursive approach until convergence.

The bias present in this approach caused it to converge with high bias, so the bias correction below was implemented. Some results, such as those below, seemed promising. There were some complications created by the use of completely artificial numbers for the temperature and downwelling radiance. Regardless, the recursive algorithm itself failed to do much better than converging to the closest ridge peak and was often far from the true answer. This was the original motivation for examining bias correction as a method for improved accuracy.

While, due to poor performance, the algorithm was not used in the previous chapters, this bias

estimation approach could still be used to estimate the bias in the eigenstep estimations, or at least provide a point from which to start estimating the bias. The bias correction was ultimately unused, since using significant numbers of observations reduces or removes the bias, but this could be useful where sufficient observations are not available.

5.2 Statistical Model

This statistical model essentially starts from Chapter 3. In this development, only a single wavelength is employed. The analysis developed below can be applied to each wavelength. The measurement at the ith pixel is

$$y_i = b_i\epsilon + (1-\epsilon)l_i, \qquad i = 1, 2, \ldots, n.$$

For generality in this initial model, the temperature, represented using b_i, was considered to vary from measurement to measurement. Pixels are chosen, however, in such a way that the temperature variation is small, and in some analysis is assumed to be constant. Statistically, then $y_i|b_i\epsilon \sim \mathcal{N}(b\epsilon + 1 - \epsilon)\mu, (1-\epsilon)^2\sigma_l^2)$ so that the likelihood function is

$$\begin{aligned} &f(y_1, \ldots, y_n|b_1, \ldots, b_n, \epsilon) = \\ &\frac{1}{(2\pi)^{n/2}[(1-\epsilon)^2\sigma_l^2]^{n/2}} \times \\ &\qquad \exp[-\frac{1}{2(1-\epsilon)^2\sigma_l^2}\sum_{i=1}^{n}(y_i - b_i\epsilon - (1-\epsilon)\mu)^2] \end{aligned} \tag{5.1}$$

Assuming the $\{b_i\}$ are known, equating the derivative of the log-likelihood with respect to ϵ to 0 leads to a quadratic equation likelihood equation $a_l\epsilon^2 + b_l\epsilon + c_l$ where $a_l = n\sigma_l^2$,

$$\begin{aligned} b_l &= -2n\sigma_l^2 + \sum_{i=1}^{n}(b_i - y_i)(\mu - b_i) \\ &= -2n\sigma_l^2 + \sum_{i=1}^{n}(1-\epsilon)(l - b_i)(b_i - \mu) \end{aligned}$$

$$c_l = \quad = n\sigma_l^2 + \sum_{i}(y_i - b_i)(\mu - y_i)$$

The maximum likelihood estimate for emissivity is thus

$$\hat{\epsilon}_{\text{ML}} = \frac{-b_l \pm \sqrt{b_l^2 - 4a_l c_l}}{2a_l} \stackrel{\triangle}{=} \frac{-b_l \pm \sqrt{\Delta}}{2a_l}. \tag{5.2}$$

The root is selected which is in the range $[0, 1]$. The bias of this estimator is of interest.

$$E[\hat{\epsilon}_{\text{ML}}] = \frac{1}{2a_l}(-E[b_l] \pm E[\sqrt{\Delta}]).$$

It straightforward to show that

$$E[b_l] = -2n\sigma_l^2 - (1-\epsilon)\sum_{i=1}^{n}(\mu - b_i)^2.$$

The expectation of the square root of the discriminant is more challenging. It can be shown that

$$\begin{aligned}\Delta = 4n\sigma_l^2 \sum_i (b_i - y_i)^2 + \\ \sum_i \sum_j (\mu - b_i)(\mu - b_j)(b_i - y_i)(b_j - y_j)\end{aligned}$$

which can be written as

$$\Delta = (\mathbf{y} - \mathbf{b})^T M (\mathbf{y} - \mathbf{b}) = (1-\epsilon)^2 (\boldsymbol{\ell} - \mathbf{b})^T M (\boldsymbol{\ell} - \mathbf{b})$$

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

and

$$M = \begin{bmatrix} 4n\sigma_l^2+(\mu-b_1)^2 & (\mu-b_1)(\mu-b_2) & \cdots & (\mu-b_1)(\mu-b_n) \\ (\mu-b_1)(\mu-b_2) & 4n\sigma_l^2+(\mu-b_2)^2 & \cdots & (\mu-b_2)(\mu-b_n) \\ \vdots & & & \\ (\mu-b_1)(\mu-b_n) & (\mu-b_2)(\mu-b_n) & \cdots & 4n\sigma_l^2+(\mu-b_n)^2 \end{bmatrix}$$
$$= 4n\sigma_l^2 + (\boldsymbol{\mu}-\mathbf{b})(\boldsymbol{\mu}-\mathbf{b})^T \tag{5.3}$$

Thus

$$E[\sqrt{\Delta}] = (1-\epsilon)E\left[\sqrt{(\boldsymbol{\ell}-\mathbf{b})^T M(\boldsymbol{\ell}-\mathbf{b})}\right] \stackrel{\triangle}{=} (1-\epsilon)s.$$

To compute this expectation, some change of variables is needed. Write the Cholesky factorization of M as $M = LL^T$ and let $\mathbf{z} = L^T(\boldsymbol{\ell}-\mathbf{b})$, so that $\Delta = (1-\epsilon)^t \mathbf{z}^T\mathbf{z}$. The mean of $\mathbf{z}$ is $E[\mathbf{z}] = L^T(\boldsymbol{\mu}-\mathbf{b})$ and the covariance is

$$\operatorname{cov}(\mathbf{z}) = L^T \operatorname{cov}(\mathbf{y}) L = \sigma_l^2 L^T L \stackrel{\triangle}{=} Q.$$

Thus the components of $\mathbf{z}$ are correlated. They can be de-correlated by writing $\mathbf{w} = V^T\mathbf{z}$, where V is the matrix of normalized eigenvectors of Q such that $Q = V\Lambda V^T$. Then

$$(1-\epsilon)^2\mathbf{w}^T\mathbf{w} = (1-\epsilon)^2\mathbf{z}^T V V^T\mathbf{z} = (1-\epsilon)^2\mathbf{z}^T\mathbf{z} = \Delta.$$

The mean and covariance of $\mathbf{w}$ are $E[\mathbf{w}] = V^T L^T(\boldsymbol{\mu}-\mathbf{b})$ and

$$\operatorname{cov}(\mathbf{w}) = V^T \operatorname{cov}(\mathbf{z}) V = V^T V \Lambda V^T V = \Lambda,$$

so the components of $\mathbf{w}$ are uncorrelated. Combining the transformations, $\mathbf{w} = V^T L^T(\boldsymbol{\ell}-\mathbf{b})$, and

$$E[\sqrt{\Delta}] = (1-\epsilon)E[\sqrt{\mathbf{w}^T\mathbf{w}}] \stackrel{\triangle}{=} (1-\epsilon)s.$$

In terms of this notation,

$$\begin{aligned} E[\hat{\epsilon}_{\text{ML}}] &= \frac{1}{2n\sigma_l^2}\left[2n\sigma_l^2 + (1-\epsilon)\sum_{i=1}^{n}(\mu - b_i)^2 \pm (1-\epsilon)s\right] \\ &= \epsilon\left[\frac{-\sum_{i=1}^{n}(\mu-b_i)^2 \mp s}{2n\sigma_l^2}\right] + \\ &\qquad \frac{2n\sigma_l^2 + \sum_{i=1}^{n}(\mu-b_i)^2 \pm s}{2n\sigma_l^2} \\ &\triangleq b_\epsilon \epsilon + a_\epsilon \end{aligned}$$

Depending on the parameters b_ϵ and a_ϵ, the maximum likelihood estimate may be unbiased. However, assuming that s can be computed (as discussed below), an unbiased estimate is

$$\hat{\hat{\epsilon}}_{\text{ML}} \triangleq \frac{\hat{\epsilon}_{ML} - a_\epsilon}{b_\epsilon} \tag{5.4}$$

5.3 Computing s

Computing the unbiased estimate (5.4) requires knowledge of s. Since $\mathbf{w}$ is just a Gaussian vector with known mean and covariance, $E[\sqrt{\mathbf{w}^T\mathbf{w}}]$ could be computed using simulation. However, since there are n dimensions in the random vector, reliably simulating the expectation runs into difficulty with the curse of dimensionality, if even a moderate number (such as 5) samples are needed for each dimension, resulting in 5^n samples for an accurate estimate. A method which combines approximation, analysis, and simulation is used to determine the simulation. First, assume (or approximate) that the elements in $\mathbf{b}$ are all the same. As mentioned above, this is a reasonable approximation under the sampling methods intended. If $\mathbf{b}$ is a constant vector, by (5.3), M is a circulant matrix. The eigenvalues of M are the DFT of the elements in a row of M. The DFT of the first row of M is

$$\begin{aligned} & DFT\left[4n\sigma_l^2 \quad 0 \quad 0\cdots \quad 0\right] \\ &\qquad + DFT\left[(\mu-b)^2 \quad (\mu-b)^2 \quad \cdots \quad (\mu-b)^2\right] \\ &= \left[4n\sigma_l^2 \quad 4n\sigma_l^2 \quad \cdots \quad 4n\sigma_l^2\right] \\ &\qquad + \left[n(\mu-b)^2 \quad 0 \quad \cdots \quad 0\right] \end{aligned}$$

Thus, there are two distinct values of eigenvalues, $4n\sigma_l^2$ which is repeated $n-1$ times, and $4n\sigma_l^2 + n(\mu - b)^2$.

It is not the eigenvalues of M that are needed, however, but the eigenvalues of Q. It is straightforward to show that the eigenvalues of Q are equal to the eigenvalues of M multiplied by σ_l^2. Q thus has the same pattern of $n-1$ repeated eigenvalues. Ordering the components so that the largest eigenvalue is last, we denote these eigenvalues, the variances of the components of $\mathbf{w}$ as $\sigma_1^2 = \sigma_l^2(4n\sigma_l^2)$ and $\sigma_n^2 = \sigma_l^2(4n\sigma_l^2 + n(\mu - b)^2$. We can write

$$\begin{aligned}\mathbf{w}^T\mathbf{w} = \sum_{i=1}^{n} w_i^2 = \sigma_1^2 \sum_{i=1}^{n} \left(\frac{w_i}{\sigma_i}\right)^2 &\triangleq \sigma_1^2 \left(\sum_{i=1}^{n-1} x_i^2 + x_n^2\right) \\ &= \sigma_1^2(\chi'^2_{n-1}(\lambda) + x_n^2) \end{aligned} \tag{5.5}$$

where [52, Chapter 29] $\chi'^2_{n-1}(\lambda)$ is a noncentral χ^2 random variable with $n-1$ degrees of freedom and noncentrality parameter

$$\lambda = \sum_{i=1}^{n-1} \left(\frac{\eta_i}{\sigma_1}\right)^2$$

and

$$x_n \sim \mathcal{N}(\eta_4/\sigma_1, \sigma_n^2/\sigma_1^2)$$

A $\chi'^2_{n-1}(\lambda)$ random variable can be efficiently simulated; for example, the MATLAB function `ncxrnd` simulates a non-central χ^2 with a single Poisson simulation and two gamma simulations.) Since $\mathbf{w}^T\mathbf{w}$ has now been expressed as the sum of two random variables, the expectation can now be simulated efficiently. Let N be a number of iterations. Then $E[\sqrt{\Delta}]$ can be estimated as follows.

---

1 $\hat{s} = 0$
2 **for** $i = 1$ *to* n **do**
3 | draw $y_1 \sim \chi'^2_{n-1}(\lambda)$
4 | draw $y_2 \sim \mathcal{N}(\eta_n/\sigma_1, \sigma_n^2/\sigma_1^2)$
5 | $\hat{s}$ += $\sqrt{y_1 + y_2^2}$
6 $\hat{s} = \hat{s}/N$

As an example of the computation, we estimate ϵ in the case that $b = 1.2$, $\sigma_l^2 = 0.5$ and $\mu = 0.2$ with $n = 5$ observations. The true ϵ is 0.8. Figure 5.1(a) shows the likelihood function, with the red dashed line indicating the true ϵ. Figure 5.1(b) shows the histogram of the maximum likelihood estimate $\hat{\epsilon}_{\text{ML}}$ and the adjusted estimate $\hat{\hat{\epsilon}}_{\text{ML}}$, obtained from running the estimator on 10,000 iterations with observations generated independently each time. From the figure, both estimates are close to the true value. In fact, the parameter a_ϵ is small and b_ϵ is near 1, with values such as $\alpha_\epsilon = 0.06$, $b_\epsilon = 0.93$ being in a typical range. Thus, while there is a bias, it is small.

5.4 Estimating b and TES

Assuming that all of the b_i are the same (same temperature in each measured pixel), maximizing the likelihood function (5.1) with respect to b gives the straightforward estimate

$$\hat{b}_{\text{ML}} = \frac{\sum_{i=1}^{n} y_i - n(1-\epsilon)}{n\epsilon} \tag{5.6}$$

This can be used in an iterative fashion to estimate both temperature and emissivity. Starting from an initial guess of the temperature $\hat{b}$, the emissivity is estimated using (5.2) (or the corrected version in (5.4). This estimate is used for ϵ in (5.6), to produce an update for $\hat{b}$, which can then be used in the emissivity estimate, etc.

Figure 5.2 shows the result of this iterative algorithm for a particular example. The dashed lines show the true emissivity and temperature, using $n = 5$ observations. Within fewer than five iterations the algorithm has converged to good values. The initial $\hat{b} = 0.8$.

5.5 Conclusions

For the statistical model developed in Chapter 3, one potential way the bias can be estimated is to directly solve for the emissivity. A quadratic equation exists for an emissivity at a specific wavelength if the temperature and all other wavelengths are held constant. This direct solution can be used to estimate the bias in the estimate. Bias correction was tested and shown to work well for the toy problem presented above. Due to the differences between the final eigenstep algorithm and the above approach, further development will be required in order to apply this approach.

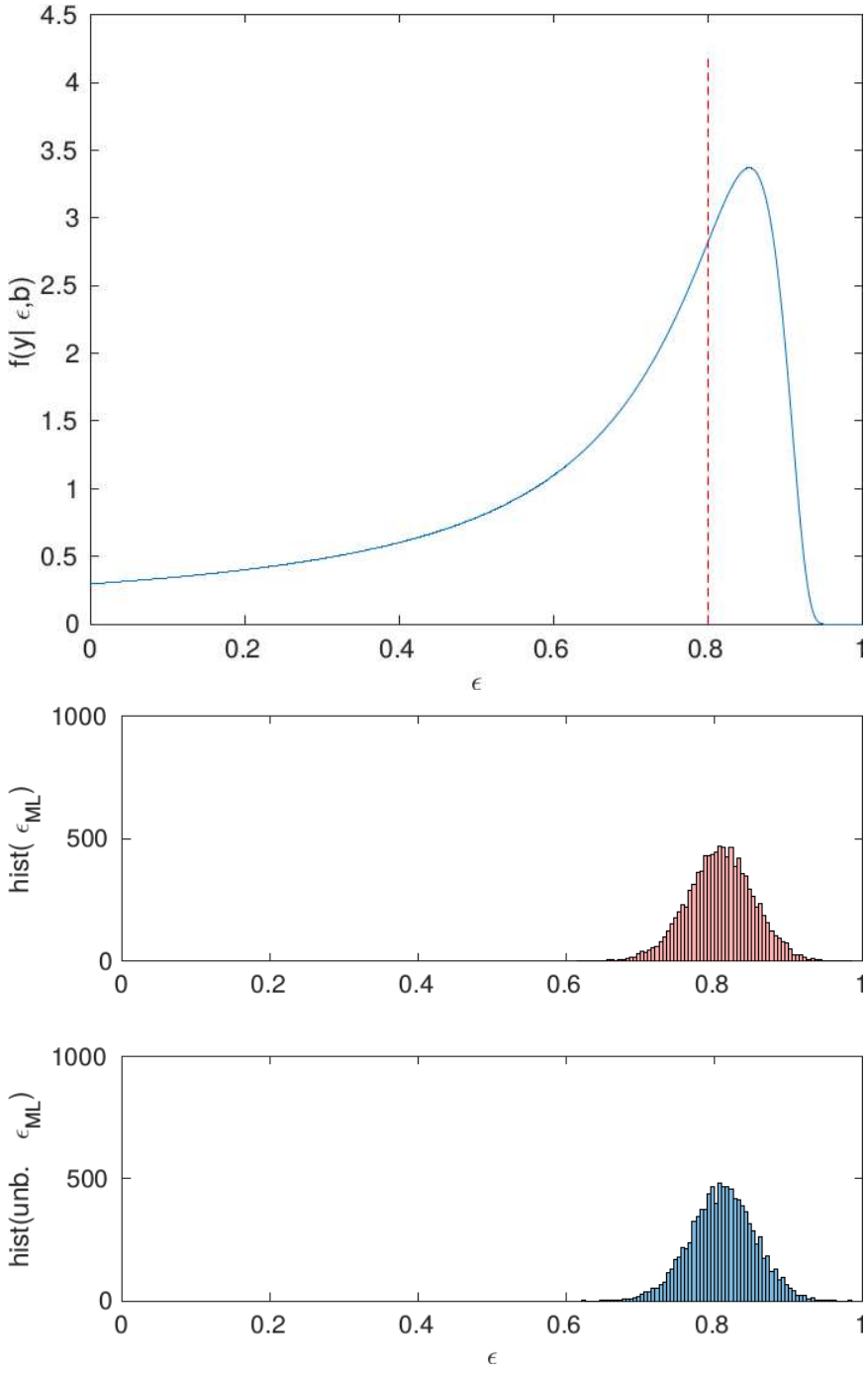

Fig. 5.1: (a) Likelihood Function; (b) ML Estimate; (c) and Adjusted ML Estimate

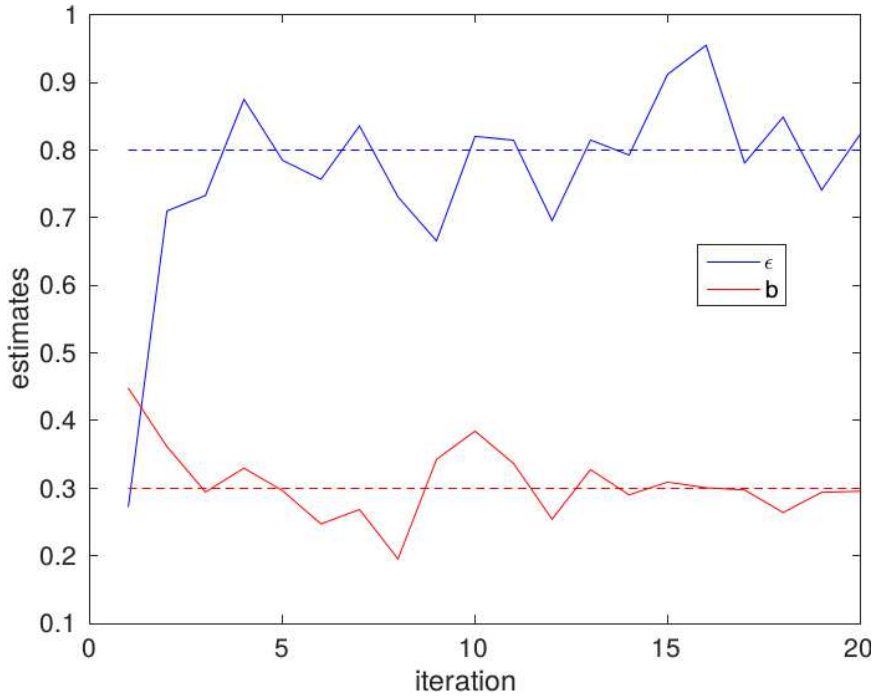

Fig. 5.2: Joint Temperature and Emissivity Separation, $n = 5$ Observations, $\mu = 0.2$, $\sigma_l^2 = 0.1$.

CHAPTER 6
CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

In this dissertation, the problem of ambiguity in solutions to the TES problem was addressed. A model was outlined for the approach and was shown to reduce ambiguity in the hyperspectral image model.

After testing traditional approaches in order to find the maximum likelihood solution, it was determined that the ridge-shape of the objective function was not well suited to these approaches. A new approach making use of the eigen decomposition of the Hessian was applied. An exhaustive search of the literature showed this approach to be new. This algorithm is generally applicable to optimizing along any function with an optimal value located on a ridge.

This new approach was developed into an algorithm tailored to our model. This approach performed with similar accuracy to gradient ascent while converging orders of magnitude faster. Preliminary tests with real data showed some promise in spite of the lack of necessary information for initialization.

Extensive testing of the algorithm was performed to characterize its convergence speed and accuracy. The impact of variation in various parameters on performance was identified. While initial results appeared biased, it was found that increasing the amount of wavelengths considered and the number of observations taken greatly improved the performance while not significantly increasing the number of iterations necessary for convergence. The algorithm was also compared to several approaches in the existing literature. Although exact comparison between algorithms with such different assumptions and optimality criteria does not necessarily yield conclusion results, the eigenstep algorithm did provide comparable performance to existing algorithms. Potential improvements to the algorithm and advantages in initialization suggest that it is superior to some algorithms and would be a good choice for performing TES.

Finally, a potential method for estimating the bias in the algorithm was discussed. This could be of use when insufficient observations are available to achieve the desired accuracy.

The main contributions of this work are the:

1. Development and characterization of a new statistical model for hyperspectral image pixels that reduces ambiguity for TES;

2. Development of a novel optimization technique for finding the maximum value along a long, relatively flat ridge (or minimum in a valley);

3. And development and characterization of an algorithm for finding the maximum likelihood solution to the new statistical model based on this new optimization technique.

In characterizing the model, it was shown that the eigenstep algorithm converged quickly, at least with regards to the number of iterations. The complexity of the algorithm is rather high, given that a matrix inverse is required for the Newton step portions and the eigenstep requires calculation of an eigen decomposition. Further, the algorithm suffers from the curse of dimensionality. The size of the Hessian increases with the number of wavelengths, which both increases the complexity of the matrix inverse and eigen decomposition.

From a performance standpoint, the algorithm is able to find a single maximum with decent accuracy. Increasing the number of observations and wavelengths allows improvement of performance at the cost of increased complexity. The number of iterations remains approximately constant even as complexity increases.

The algorithm can be made unbiased, or close to unbiased, by increasing the number of observations. Unbiasing will only be useful inasmuch as it is less computationally complex than increasing the number of observations or wavelengths. It could be essential if a limited number of observations were available.

The algorithm is sensitive to noise. There was a direct increase in the estimator variance as the observation noise increased. In real cases, the performance will then be limited by the actual observation noise.

While the majority of the performance tests were for artificial data, some initial tests with real data were performed that show promise. A well-defined and controlled set of data will be required to truly characterize the algorithm for real observations.

6.2 Future Work

Given that that this work proposes a new model and algorithm for performing TES, it is expected that a large amount of work could be based on this dissertation. Following are some potential areas for future research.

Some of the most obvious changes would be in the assumptions. It was assumed that the temperature for all the pixels was identical. Due to the lack of mixing between temperatures, it would be relatively simple to develop the algorithm for sets of pixels with the same emissivity, but different temperatures.

It was also assumed that the pixels were pure pixels. Adding mixed pixels would also be relatively straightforward if a linear mixing model (LMM) was applied [3, 7, 53, 54] . This would provide sets of linearly independent equations with different temperatures and emissivities, but the same downwelling. Care would have to be taken with this, as the data would most likely be reduced to single observations, since real pixels seldom will have identical mixing. Non-linear models could also be used [6, 55], although this might significantly complicate the gradients and Hessian.

The downwelling variance was arbitrarily chosen. It was used in generating the observations, but this downwelling was not based on any real data. Future work could explore what the actual downwelling variance looks like. This could be developed either from real data or MODTRAN simulations.

Additional work could be performed with real observations. This would especially useful if ground truth data were available and the downwelling distribution and noise variance could be better defined (e.g., from real data).

For smaller numbers of observations, there appears to be a bias in the estimate. Additional work building on that outlined in Chapter 5 could be performed. New methods of estimating and remove the bias could also be developed. Additionally, the impact of the number of observations on

performance could be more accurately defined in order to identify when bias correction might be needed.

The algorithm becomes less tractable as the number of wavelengths increases due to having to invert the Hessian and calculate its eigen decomposition. Alternative approaches to the Newton step or even the eigenstep approach could be developed that might be more efficient in calculations or convergence speed. In some cases, approximations are also effective substitutes for these computationally costly calculations.

One approach to this might be to exploit the actual relationship between elements at different wavelengths. A method of estimating the downwelling covariance, either through use of MODTRAN, historical data, or actual observations would be needed to identify the actual covariance structure. If the covariance of the downwelling radiance does not contain correlation between certain wavelengths, smaller groupings of wavelengths could be used for speed. When the correlation is absent, there should be no gain for adding additional wavelengths, so this could naturally provide an upper limit on the number of wavelengths to use.

More interestingly, if a given wavelength was only correlated with a few adjacent wavelengths, but this was true across the entire set of wavelengths, a sub group could be identified that included all the correlations for a specific wavelength or set of wavelengths. After estimating this, the algorithm could shift to a new group that overlapped, similar to taking a spectrogram. The previous estimate could be used for the overlapping values.

This should increase the convergence speed, since the initial condition would be closer to the right answer. It might not give much in performance gain, but it might approach the performance gain of including all the wavelengths at once while being significantly faster computationally.

Finally, more detailed comparison with other algorithms could be performed. Those algorithms could be implemented. Common data could be used for both so, at least in the data, the comparison could be more fair. The actual impact of the variations between the model assumptions and optimality criteria could be compared more accurately. This would provide a better understanding of how well the eigenstep algorithm really performs compared to existing TES techniques.

REFERENCES

[1] J. R. Schott, *Remote Sensing : The Image Chain Approach: The Image Chain Approach*, 2nd ed. New York, NY: Oxford University Press, 2007.

[2] D. Manolakis, E. Truslow, M. Pieper, T. Cooley, and M. Brueggeman, "Detection algorithms in hyperspectral imaging systems: An overview of practical algorithms," *Signal Processing Magazine, IEEE*, vol. 31, no. 1, pp. 24–33, Jan 2014.

[3] J. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, vol. 5, no. 2, pp. 354–379, April 2012.

[4] F. A. Kruse, J. W. Boardman, and J. F. Huntington, "Comparison of airborne hyperspectral data and eo-1 hyperion for mineral mapping," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 6, pp. 1388–1400, June 2003.

[5] S. J. Hook, A. R. Gabell, A. A. Green, and P. S. Kealy, "A comparison of techniques for extracting emissivity information from thermal infrared data for geologic studies," *Remote Sensing of Environment*, vol. 42, no. 2, pp. 123–135, 1992.

[6] W.-K. Ma, J. Bioucas-Dias, T.-H. Chan, N. Gillis, P. Gader, A. Plaza, A. Ambikapathi, and C.-Y. Chi, "A signal processing perspective on hyperspectral unmixing: Insights from remote sensing," *Signal Processing Magazine, IEEE*, vol. 31, no. 1, pp. 67–81, Jan 2014.

[7] A. Zare and K. Ho, "Endmember variability in hyperspectral analysis: Addressing spectral variability during spectral unmixing," *Signal Processing Magazine, IEEE*, vol. 31, no. 1, pp. 95–104, Jan 2014.

[8] G. Camps-Valls, D. Tuia, L. Bruzzone, and J. Atli Benediktsson, "Advances in hyperspectral image classification: Earth monitoring with statistical learning methods," *Signal Processing Magazine, IEEE*, vol. 31, no. 1, pp. 45–54, Jan 2014.

[9] N. Nasrabadi, "Hyperspectral target detection : An overview of current and future challenges," *Signal Processing Magazine, IEEE*, vol. 31, no. 1, pp. 34–44, Jan 2014.

[10] S. J. Young, B. R. Johnson, and J. A. Hackwell, "An in-scene method for atmospheric compensation of thermal hyperspectral data," *Journal of Geophysical Research: Atmospheres*, vol. 107, no. D24, pp. ACH 14–1–ACH 14–20, 2002. [Online]. Available: http://dx.doi.org/10.1029/2001JD001266

[11] D. Gu, A. Gillespie, A. Kahle, and F. Palluconi, "Autonomous atmospheric compensation (AAC) of high resolution hyperspectral thermal infrared remote-sensing imagery," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 38, no. 6, pp. 2557–2570, Nov 2000.

[12] P. Ingram and A. Muse, "Sensitivity of iterative spectrally smooth temperature/emissivity separation to algorithmic assumptions and measurement noise," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 39, no. 10, pp. 2158–2167, Oct 2001.

[13] C. Borel and R. Tuttle, "Recent advances in temperature-emissivity separation algorithms," in *Aerospace Conference, 2011 IEEE*, March 2011, pp. 1–14.

[14] J. Jimenez-Muoz, J. Sobrino, and A. Gillespie, "Surface emissivity retrieval from airborne hyperspectral scanner data: Insights on atmospheric correction and noise removal," *Geoscience and Remote Sensing Letters, IEEE*, vol. 9, no. 2, pp. 180–184, March 2012.

[15] Y. Rong, H. Su, R. Zhang, and Y. Yang, "Emissivity measurement for low emissivity objects by two blackbody tube methods," in *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*, July 2012, pp. 6122–6125.

[16] G. Arce, D. Brady, L. Carin, H. Arguello, and D. Kittle, "Compressive coded aperture spectral imaging: An introduction," *Signal Processing Magazine, IEEE*, vol. 31, no. 1, pp. 105–115, Jan 2014.

[17] D. Lunga, S. Prasad, M. Crawford, and O. Ersoy, "Manifold-learning-based feature extraction for classification of hyperspectral data: A review of advances in manifold learning," *Signal Processing Magazine, IEEE*, vol. 31, no. 1, pp. 55–66, Jan 2014.

[18] J. Gunther, T. Moon, M. Stites, and G. Williams, "Separating temperature, emissivity and downwelling radiance in thermal infrared pure-pixel hyperspectral images," in *Signals, Systems and Computers, 2013 Asilomar Conference on*, Nov 2013, pp. 574–578.

[19] M. R. Stites, "Utilization of a Probabilistic Model for Improved Hyperspectral Unmixing," Ph.D. dissertation, Utah State University, 2011.

[20] K. Kanani, L. Poutier, F. Nerry, and M.-P. Stoll, "Directional effects consideration to improve out-doors emissivity retrieval in the 3-13 μm domain," *Opt. Express*, vol. 15, no. 19, pp. 12 464–12 482, Sep 2007. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-15-19-12464

[21] C. Borel, "Surface emissivity and temperature retrieval for a hyperspectral sensor," in *Geoscience and Remote Sensing Symposium Proceedings, 1998. IGARSS '98. 1998 IEEE International*, vol. 1, Jul 1998, pp. 546–549 vol.1.

[22] J. Jimenez-Munoz, J. Sobrino, C. Mattar, G. Hulley, and F.-M. Gottsche, "Temperature and emissivity separation from msg/seviri data," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 52, no. 9, pp. 5937–5951, Sept 2014.

[23] J. Jimenez-Muoz, J. Sobrino, J. El-Kharraz, M. Gomez, M. Romaguera, and G. Soria, "Synergistic use of dais bands to retrieve land surface emissivity and temperature," in *Geoscience and Remote Sensing Symposium, 2003. IGARSS '03. Proceedings. 2003 IEEE International*, vol. 2, July 2003, pp. 1062–1064 vol.2.

[24] N. Wang, H. Wu, F. Nerry, C. Li, and Z.-L. Li, "Temperature and emissivity retrievals from hyperspectral thermal infrared data using linear spectral emissivity constraint," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 49, no. 4, pp. 1291–1303, April 2011.

[25] M. Matsumoto-Moriyama and K. Arai, "Some ideas for iterative estimation of atmospheric condition, surface temperature and emissivity," in *Geoscience and Remote Sensing Symposium, 1993. IGARSS '93. Better Understanding of Earth Environment., International*, Aug 1993, pp. 1291–1293 vol.3.

[26] H. Wang, Q. Xiao, H. Li, and B. Zhong, "Temperature and emissivity separation algorithm for tasi airborne thermal hyperspectral data," in *Electronics, Communications and Control (ICECC), 2011 International Conference on*, Sept 2011, pp. 1075–1078.

[27] J. Morgan, "A bayesian estimator for linear calibration error effects in thermal remote sensing," *Geoscience and Remote Sensing Letters, IEEE*, vol. 3, no. 1, pp. 117–119, Jan 2006.

[28] ——, "Bayesian estimation for land surface temperature retrieval: the nuisance of emissivities," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 43, no. 6, pp. 1279–1288, June 2005.

[29] D. A. Faysash and E. A. Smith, "Simultaneous land surface temperatureemissivity retrieval in the infrared split window," *Journal of Atmospheric and Oceanic Technology*, vol. 16, no. 11, pp. 1673–1689, 1999. [Online]. Available: http://dx.doi.org/10.1175/1520-0426(1999)016⟨1673:SLSTER⟩2.0.CO;2

[30] A. Barducci, D. Guzzi, C. Lastri, P. Marcoionni, V. Nardino, and I. Pippi, "Emissivity and temperature assessment using a maximum entropy estimator: Structure and performance of the maxentes algorithm," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 2, pp. 738–751, Feb 2015.

[31] C. Borel, "Error analysis for a temperature and emissivity retrieval algorithm for hyperspectral imaging data," in *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XIII*, ser. , vol. 6565, Apr. 2007, p. 65651Q.

[32] L. Peres and C. DaCamara, "Inverse problems theory and application: analysis of the two-temperature method for land-surface temperature and emissivity estimation," *Geoscience and Remote Sensing Letters, IEEE*, vol. 1, no. 3, pp. 206–210, July 2004.

[33] X. Wang, X. OuYang, B. Tang, Z. L. Li, and R. Zhang, "A new method for temperature/emissivity separation from hyperspectral thermal infrared data," in *IGARSS 2008 - 2008 IEEE International Geoscience and Remote Sensing Symposium*, vol. 3, July 2008, pp. III – 286–III – 289.

[34] J. Cheng, S. Liang, J. Wang, and X. Li, "A stepwise refining algorithm of temperature and emissivity separation for hyperspectral thermal infrared data," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 48, no. 3, pp. 1588–1597, March 2010.

[35] Y. Hang, H. Zhaoqiang, Z. Lifu, and T. Qingxi, "Spatial scale issue in temperature and emissivity separation from thermal hyperspectral imager," in *Geoscience and Remote Sensing Symposium (IGARSS), 2013 IEEE International*, July 2013, pp. 485–488.

[36] K. Watson, "Two-temperature method for measuring emissivity," *Remote Sensing of Environment*, vol. 42, no. 2, pp. 117–121, 1992.

[37] M. Shimoni, R. Haelterman, and P. Lodewyckx, "Data fusion for improving thermal emissivity separation from hyperspectral data," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2015, pp. 2955–2958.

[38] C. Coll, V. Garca-Santos, R. Nicls, and V. Caselles, "Test of the modis land surface temperature and emissivity separation algorithm with ground measurements over a rice paddy," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 5, pp. 3061–3069, May 2016.

[39] L. Guanter, R. Richter, and J. Moreno, "Spectral calibration of hyperspectral imagery using atmospheric absorption features," *Appl. Opt.*, vol. 45, no. 10, pp. 2360–2370, Apr 2006. [Online]. Available: http://ao.osa.org/abstract.cfm?URI=ao-45-10-2360

[40] H. He, B. Zhang, B. Liu, W. Zhang, and R. Li, "Temperature and emissivity separation from aster data based on the urban land cover classification," in *Urban Remote Sensing Event, 2009 Joint*, May 2009, pp. 1–5.

[41] Y.-G. Qian, N. Wang, C. Gao, Y.-Y. Jia, L. Ma, H. Wu, Z.-L. Li, and L. Tang, "Preliminary evaluation of linear spectral emissivity constraint temperature and emissivity separation method for contrast samples from hyperspectral thermal infrared data," in *Geoscience and Remote Sensing Symposium (IGARSS), 2013 IEEE International*, July 2013, pp. 465–468.

[42] X. OuYang, X. Wang, B. H. Tang, and Z. L. Li, "Sensitive analysis of various measurement errors on tempearture and emissivity separation method with hyperspectral data," in *2009 IEEE International Geoscience and Remote Sensing Symposium*, vol. 2, July 2009, pp. II–682–II–685.

[43] A. Berk, G. P. Anderson, P. K. Acharya, L. S. Bernstein, L. Muratov, J. Lee, M. Fox, S. M. Adler-Golden, J. H. Chetwynd, M. L. Hoke, R. B. Lockwood, J. A. Gardner, T. W. Cooley, C. C. Borel, P. E. Lewis, and E. P. Shettle, "Modtran5: 2006 update," *Proceedings SPIE*, vol. 6233, no. 62331F, 2006.

[44] H. H. Rosenbrock, "An automatic method for finding the greatest or least value of a function," *The Computer Journal*, vol. 3, no. 3, pp. 175–184, 1960. [Online]. Available: http://comjnl.oxfordjournals.org/content/3/3/175.abstract

[45] D. A. Neal, T. K. Moon, J. H. Gunther, and G. Williams, "Correlated maximum likelihood temperature/emissivity separation of hyperspectral images," in *2015 49th Asilomar Conference on Signals, Systems and Computers*, Nov 2015, pp. 536–540.

[46] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY: Cambridge University Press, 2004.

[47] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipies: The Art of Scientific Computing*, 3rd ed. New York, NY: Cambridge University Press, 2007.

[48] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*. Upper Saddle River, NJ: Prentice Hall, 2000, ch. 13, pp. 591–620.

[49] J. Lawson, J. Conger, and L. Balick, "Investigation of temporal variation of lwir solid signatures using denali," Lawrence Livermore National Laboratory, Los Alamos National Laboratory, Tech. Rep. LLNL-TR-403504, May 2008.

[50] X. Zhong, Z. L. Li, H. Wu, B. H. Tang, and R. Tang, "Comparison of two hyperspectral temperature and emissivity separation methods: Cbtes and isstes," in *2014 IEEE Geoscience and Remote Sensing Symposium*, July 2014, pp. 2977–2980.

[51] T. K. Moon, D. Neal, J. H. Gunther, and G. Williams, "Temperature emissivity separation: Estimation with a parameter affecting both the mean and variance of the observation," in *Signal Processing and Signal Processing Education Workshop (SP/SPE), 2015 IEEE*, Aug 2015, pp. 380–384.

[52] N. Johnson, S. Kotz, and N. Balakrishnan, *Continuous univariate distributions*, 2nd ed., ser. Distributions in statistics. New York, NY [u.a.]: Wiley.

[53] N. Keshava and J. F. Mustard, "Spectral unmixing," *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 44–57, Jan 2002.

[54] T. K. Moon, J. H. Gunther, C. Berrett, and G. P. Williams, "Hierarchical bayesian sparse source separation of hyperspectral signals," in *Signals, Systems and Computers, 2013 Asilomar Conference on*, Nov 2013, pp. 888–892.

[55] N. Dobigeon, J.-Y. Tourneret, C. Richard, J. Bermudez, S. McLaughlin, and A. Hero, "Nonlinear unmixing of hyperspectral images: Models and algorithms," *Signal Processing Magazine, IEEE*, vol. 31, no. 1, pp. 82–94, Jan 2014.

APPENDICES

Appendix A

Gradient and Hessian Derivations

A.1 Noise-Free Gradients

The log-likelihood function for noise-free hyperspectral observations is

$$L(\{\mathbf{y}_i\}|\{\mathbf{B}(T)\},\boldsymbol{\epsilon}) = -\frac{Nn}{2}\log(2\pi) - \frac{N}{2}\log(|\Lambda_{1-\boldsymbol{\epsilon}}R\Lambda_{1-\boldsymbol{\epsilon}}|) \\ -\frac{1}{2}\sum_{i=1}^{N}(\mathbf{y}_i - \Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon} - \Lambda_{1-\boldsymbol{\epsilon}}\boldsymbol{\mu})^T(\Lambda_{1-\boldsymbol{\epsilon}}R\Lambda_{1-\boldsymbol{\epsilon}})^{-1}(\mathbf{y}_i - \Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon} - \Lambda_{1-\boldsymbol{\epsilon}}\boldsymbol{\mu}). \tag{A.1}$$

Here, N is the number of observations, n is the number of wavelengths, $\boldsymbol{\epsilon}$ is the emissivity, $\mathbf{y}_i$ is the ith observation, $\mathbf{B}(T)$ is the black body function, and $\boldsymbol{\mu}$ and R are the mean and covariance of the downwelling radiance, respectively.

First, consider the derivative with respect to ϵ_ℓ, the ℓth component of $\boldsymbol{\epsilon}$. The first term of (A.1) is constant, so it is ignored. The second and third terms are much more complex.

For the second term, the partial derivative is

$$\frac{\partial}{\partial\epsilon_\ell}(-\frac{N}{2}\log(|\Lambda_{1-\boldsymbol{\epsilon}}R\Lambda_{1-\boldsymbol{\epsilon}}|)). \tag{A.2}$$

Using determinant rules, the determinant of the product is the product of the determinants. The log of the product is broken into the sum of the logs. The two of the resulting terms are identical and combine into

$$\frac{\partial}{\partial\epsilon_\ell}(-N\log(|\Lambda_{1-\boldsymbol{\epsilon}}|)), \tag{A.3}$$

This partial derivative is

$$\frac{N}{1-\epsilon_\ell}. \tag{A.4}$$

The third term is

$$-\frac{1}{2}\sum_{i=1}^{N}(\mathbf{y}_i-\Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon}-\Lambda_{1-\boldsymbol{\epsilon}}\boldsymbol{\mu})^T(\Lambda_{1-\boldsymbol{\epsilon}}R\Lambda_{1-\boldsymbol{\epsilon}})^{-1}(\mathbf{y}_i-\Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon}-\Lambda_{1-\boldsymbol{\epsilon}}\boldsymbol{\mu}). \tag{A.5}$$

As before, factor out the $\Lambda_{1-\boldsymbol{\epsilon}}$, yielding

$$-\frac{1}{2}\sum_{i=1}^{N}(\mathbf{y}_i-\Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon}-\Lambda_{1-\boldsymbol{\epsilon}}\boldsymbol{\mu})^T\Lambda_{1/(1-\boldsymbol{\epsilon})}R^{-1}\Lambda_{1/(1-\boldsymbol{\epsilon})}(\mathbf{y}_i-\Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon}-\Lambda_{1-\boldsymbol{\epsilon}}\boldsymbol{\mu}). \tag{A.6}$$

Then, for convenience in notation, define $\mathbf{w}_i^T=(\mathbf{y}_i-\Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon}-\Lambda_{1-\boldsymbol{\epsilon}}\boldsymbol{\mu})^T\Lambda_{1/(1-\boldsymbol{\epsilon})}$ and rewrite (A.6) as

$$-\frac{1}{2}\sum_{i=1}^{N}\mathbf{w}_i^TR^{-1}\mathbf{w}_i. \tag{A.7}$$

From this, find

$$-\frac{1}{2}\sum_{i=1}^{N}\frac{\partial}{\partial\epsilon_\ell}\mathbf{w}_i^TR^{-1}\mathbf{w}_i, \tag{A.8}$$

where $\mathbf{w}_i$ is a function of ϵ_ℓ. Apply the chain rule and, recognizing the symmetry, combine the derivatives of the two factors, which can be written as

$$-\sum_{i=1}^{N}(\frac{\partial\mathbf{w}_i^T}{\partial\epsilon_\ell})R^{-1}\mathbf{w}_i. \tag{A.9}$$

Next, consider the factor

$$\frac{\partial\mathbf{w}_i^T}{\partial\epsilon_\ell}. \tag{A.10}$$

Only the ℓth component is a function of ϵ_ℓ. The non-zero part of the derivative is

$$\frac{(\mu_\ell-B(T)_\ell)(1-\epsilon_\ell)+(y_{i,\ell}-B(T)_\ell\epsilon_\ell-(1-\epsilon_\ell)\mu_\ell)}{(1-\epsilon_\ell)^2}. \tag{A.11}$$

This can be reduced to

$$\frac{(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^2}. \tag{A.12}$$

Stacking into vectors or matrices, the derivative of the third term can be written as

$$-\sum_{i=1}^{N}\frac{(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^2}\mathbf{e}_\ell^T R^{-1}\mathbf{w}_i, \tag{A.13}$$

where $\mathbf{e}_\ell$ is an elemental vector.

The overall derivative of the log-likelihood with respect to ϵ_ℓ is the sum of (A.4) and (A.13). There will be a set of n scalars that can be stacked into a vector representing the emissivity derivative at each wavelength.

For the derivative with respect to T, only the third term in (A.1) needs to be considered, as it is the only portion of the log likelihood that is a function of temperature. Consider only the term

$$-\frac{1}{2}\sum_{i=1}^{N}(\mathbf{y}_i-\Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon}-\Lambda_{1-\boldsymbol{\epsilon}}\boldsymbol{\mu})^T(\Lambda_{1-\boldsymbol{\epsilon}}R\Lambda_{1-\boldsymbol{\epsilon}})^{-1}(\mathbf{y}_i-\Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon}-\Lambda_{1-\epsilon}\mu). \tag{A.14}$$

Using the same notation as before, (A.14) can be rewritten as

$$-\frac{1}{2}\sum_{i=1}^{N}\mathbf{w}_i^T R_{obs}^{-1}\mathbf{w}_i. \tag{A.15}$$

Take the partial derivative with respect to T. Since $\mathbf{w}_i$ is a function of T, the chain rule can be used and the partial derivative, adding the symmetric terms, can be written as

$$-\sum_{i=1}^{N}(\frac{\partial\mathbf{w}_i}{\partial T})(R^{-1}\mathbf{w}_i. \tag{A.16}$$

The only term in $\mathbf{w}$ that is a function of T is $\Lambda(\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon}))\mathbf{B}(T)$. Factor the portion that is a function of $\boldsymbol{\epsilon}$, leaving the Planck function, which is

$$B(T)_k = C_1\lambda_k^{-5}(e^{C_2/(\lambda_k T)}-1)^{-1}, \tag{A.17}$$

where $C_1 = 1.191\times10^{10}Wcm^{-2}sr^{-1}\mu m^4$, λ_k is the kth wavelength in μm, $C_2 = 1.4388\times10^4\mu mK$, and T is the temperature in degrees Kelvin.

Taking the derivative of this with respect to T yields

$$\frac{\partial B(T)_k}{\partial T} = \frac{C_1 C_2 e^{C_2/(\lambda_k T)}}{\lambda_k^6 T^2 (e^{C_2/(\lambda_k T)} - 1)^2}. \tag{A.18}$$

Writing the derivatives out, the partial of (A.14) with respect to T is

$$\sum_{i=1}^{N} \mathbf{w}_i'^T \Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})} R^{-1} \mathbf{w}_i, \tag{A.19}$$

where $w'_{i,k} = C_1 C_2 e^{C_2/(\lambda_k T)}/(\lambda_k^6 T^2 (e^{C_2/(\lambda_k T)} - 1)^2)$.

A.2 Noisy Gradients

The log-likelihood function for noisy hyperspectral observations with zero-mean white Gaussian noise is

$$\begin{aligned} L(\{\mathbf{y}_i\}|\{\mathbf{B}(T)\}, \boldsymbol{\epsilon}) &= -\frac{Nn}{2}\log(2\pi) - \frac{N}{2}\log(|\Lambda_{1-\boldsymbol{\epsilon}} R \Lambda_{1-\boldsymbol{\epsilon}} + \sigma^2 I|) \\ -\frac{1}{2}\sum_{i=1}^{N} (\mathbf{y}_i - \Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon} - \Lambda_{1-\boldsymbol{\epsilon}}\boldsymbol{\mu})^T (\Lambda_{1-\boldsymbol{\epsilon}} R \Lambda_{1-\boldsymbol{\epsilon}} + \sigma^2 I)^{-1} (\mathbf{y}_i - \Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon} - \Lambda_{1-\boldsymbol{\epsilon}}\boldsymbol{\mu}). \end{aligned} \tag{A.20}$$

Here, N is the number of observations, n is the number of wavelengths, $\boldsymbol{\epsilon}$ is the emissivity, $\mathbf{y}_i$ is the ith observation, $\mathbf{B}(T)$ is the black body function, σ^2 is the variance of the noise, and $\boldsymbol{\mu}$ and R are the mean and covariance of the downwelling radiance, respectively.

First consider the derivative with respect to ϵ_ℓ, the ℓth component of $\boldsymbol{\epsilon}$. The first term of (A.20) is constant, so it is ignored. The second and third terms are much more complex.

For the second term, the partial derivative is

$$\frac{\partial}{\partial \epsilon_\ell}(-\frac{N}{2}\log(|\Lambda_{1-\boldsymbol{\epsilon}} R \Lambda_{1-\boldsymbol{\epsilon}} + \sigma^2 I|)). \tag{A.21}$$

The $\Lambda_{1-\boldsymbol{\epsilon}}$ can be factored out, giving

$$\frac{\partial}{\partial \epsilon_\ell}(-\frac{N}{2}\log(|\Lambda_{1-\boldsymbol{\epsilon}}(R + \sigma^2 \Lambda_{1/(1-\boldsymbol{\epsilon})}\Lambda_{1/(1-\boldsymbol{\epsilon})})\Lambda_{1-\boldsymbol{\epsilon}}|)). \tag{A.22}$$

Using determinant rules, the determinant of the product is the product of the determinant. Then break the log of the product into the sum of the logs. Two of the resulting terms are identical and combine into

$$\frac{\partial}{\partial \epsilon_\ell}(-N\log(|\Lambda_{1-\boldsymbol{\epsilon}}|)), \tag{A.23}$$

This was worked through before above and was found to be

$$\frac{N}{1-\epsilon_\ell}. \tag{A.24}$$

The remaining term is

$$\frac{\partial}{\partial \epsilon_\ell}(-\frac{N}{2}\log(|R+\sigma^2\Lambda_{1/(1-\boldsymbol{\epsilon})}\Lambda_{1/(1-\boldsymbol{\epsilon})}|)). \tag{A.25}$$

In order to find this derivative, the chain rule will have to be applied.

First, for convenience, define

$$X = R+\sigma^2\Lambda_{1/(1-\boldsymbol{\epsilon})}\Lambda_{1/(1-\boldsymbol{\epsilon})}. \tag{A.26}$$

Then rewrite (A.25) as

$$\frac{\partial}{\partial \epsilon_\ell}(-\frac{N}{2}\log(|X|)). \tag{A.27}$$

Now use the chain rule, as X is a function of ϵ_ℓ. This rewrites as

$$-\frac{N}{2}\sum_{j=1}^{n}\sum_{k=1}^{n}(\frac{\partial X}{\partial \epsilon_\ell})_{j,k}(\frac{\partial \log(|X|)}{\partial X})_{j,k}. \tag{A.28}$$

Each element of the first factor is the derivative of $X_{j,k}$ with respect to ϵ_ℓ. Note that elements of $\boldsymbol{\epsilon}$ only appear in the diagonal of X, so the derivative is only non-zero in the diagonal elements and the double sum collapses to a single sum, since j always equals k. Further, the jth diagonal element is only a function of one element of $\boldsymbol{\epsilon}$, so the remaining single sum collapses to the ℓth element. This results in

$$-\frac{N}{2}(\frac{\partial X}{\partial \epsilon_\ell})_{\ell,\ell}(\frac{\partial \log(|X|)}{\partial X})_{\ell,\ell}. \tag{A.29}$$

For the first factor, this is $2\sigma^2/(1-\epsilon_\ell)^3$. For the second factor, the partial of the log determinant

with respect to the argument is $2X^{-1} - \mathrm{diag}(X^{-1})$. Take the ℓ, ℓth component of this. Note that both elements are pulled from the diagonal, so this is essentially $2X^{-1}_{\ell,\ell} - X^{-1}_{\ell,\ell} = X^{-1}_{\ell,\ell}$. The overall result is then

$$- N\frac{\sigma^2 X^{-1}_{\ell,\ell}}{(1-\epsilon_\ell)^3}. \tag{A.30}$$

Now move to the third term, which is

$$-\frac{1}{2}\sum_{i=1}^{N}(\mathbf{y}_i - \Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon} - \Lambda_{1-\boldsymbol{\epsilon}}\boldsymbol{\mu})^T(\Lambda_{1-\boldsymbol{\epsilon}}R\Lambda_{1-\boldsymbol{\epsilon}} + \sigma^2 I)^{-1}(\mathbf{y}_i - \Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon} - \Lambda_{1-\boldsymbol{\epsilon}}\boldsymbol{\mu}). \tag{A.31}$$

As before, factor out the $\Lambda_{1-\boldsymbol{\epsilon}}$, yielding

$$\begin{aligned}-\frac{1}{2}\sum_{i=1}^{N}(\mathbf{y}_i - \Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon} - \Lambda_{1-\boldsymbol{\epsilon}}\boldsymbol{\mu})^T\Lambda_{1/(1-\boldsymbol{\epsilon})}(R + \Lambda_{1/(1-\boldsymbol{\epsilon})}\sigma^2\Lambda_{1/(1-\boldsymbol{\epsilon})})^{-1} \\ \times\Lambda_{1/(1-\boldsymbol{\epsilon})}(\mathbf{y}_i - \Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon} - \Lambda_{1-\boldsymbol{\epsilon}}\boldsymbol{\mu}).\end{aligned} \tag{A.32}$$

Then define $\mathbf{w}_i^T = (\mathbf{y}_i - \Lambda_{\mathbf{B}(T)}\boldsymbol{\epsilon} - \Lambda_{1-\boldsymbol{\epsilon}}\boldsymbol{\mu})^T\Lambda_{1/(1-\boldsymbol{\epsilon})}$ and use X from before, rewriting now as

$$-\frac{1}{2}\sum_{i=1}^{N}\mathbf{w}_i^T X^{-1}\mathbf{w}_i. \tag{A.33}$$

From this, the desired partial derivative can be written as

$$-\frac{1}{2}\sum_{i=1}^{N}\frac{\partial}{\partial\epsilon_\ell}\mathbf{w}_i^T X^{-1}\mathbf{w}_i, \tag{A.34}$$

where now both X and $\mathbf{w}_i$ are functions of ϵ_ℓ. Apply the chain rule and, recognizing the symmetry, combine the derivatives of the first and third factor. The result can be written as

$$-\sum_{i=1}^{N}(\frac{\partial\mathbf{w}_i^T}{\partial\epsilon_\ell})X^{-1}\mathbf{w}_i + \frac{1}{2}\mathbf{w}_i^T(\frac{\partial X^{-1}}{\partial\epsilon_\ell})\mathbf{w}_i. \tag{A.35}$$

First consider

$$\frac{\partial\mathbf{w}_i^T}{\partial\epsilon_\ell}. \tag{A.36}$$

Only the ℓth component is a function of ϵ_ℓ. The non-zero part of the derivative is

$$\frac{(\mu_\ell - B(T)_\ell)(1-\epsilon_\ell) + (y_{i,\ell} - B(T)_\ell\epsilon_\ell - (1-\epsilon_\ell)\mu_\ell)}{(1-\epsilon_\ell)^2}. \tag{A.37}$$

This can be reduced to

$$\frac{(y_{i,\ell} - B(T)_\ell)}{(1-\epsilon_\ell)^2}. \tag{A.38}$$

Next consider

$$\frac{\partial X^{-1}}{\partial \epsilon_\ell} \tag{A.39}$$

A direct solution is unclear. As a solution around this problem, write

$$X^{-1}X = I. \tag{A.40}$$

Taking the partial of both sides yields

$$\frac{\partial X^{-1}}{\partial \epsilon_\ell}X + X^{-1}\frac{\partial X}{\partial \epsilon_\ell} = 0. \tag{A.41}$$

The desired quantity can be solved for as

$$\frac{\partial X^{-1}}{\partial \epsilon_\ell} = -X^{-1}\frac{\partial X}{\partial \epsilon_\ell}X^{-1}. \tag{A.42}$$

From before, the derivative of the ℓth diagonal element is $2\sigma^2/(1-\epsilon_\ell)^3$ and all other derivatives are zero. Stack this into a matrix, giving

$$-2\sigma^2/(1-\epsilon_\ell)^3 X^{-1}E_{\ell,\ell}X^{-1}, \tag{A.43}$$

where $E_{\ell,\ell}$ is the elemental matrix. Stacking into vectors or matrices, the solution can be written for the third term as

$$-\sum_{i=1}^{N}\frac{(y_{i,\ell} - B(T)_\ell)}{(1-\epsilon_\ell)^2}\mathbf{e}_\ell^T X^{-1}\mathbf{w}_i - \sigma^2/(1-\epsilon_\ell)^3\mathbf{w}_i^T X^{-1}E_{\ell,\ell}X^{-1}\mathbf{w}_i, \tag{A.44}$$

where $\mathbf{e}_\ell$ is an elemental vector.

The overall derivative of the log-likelihood with respect to ϵ_ℓ is the sum of (A.24), (A.30), and (A.44). There will be a set of n scalars that can be stacked into a vector representing the emissivity derivative at each wavelength.

For the temperature case, simply replace R in the previous derivatives with X. As this is not a function of temperature, no other changes are made.

A.3 Noise-Free Hessian

When calculating the Hessian, two terms must be considered. The first is the gradient with respect to ϵ_ℓ, which is the sum of (A.4) and (A.13). The second term is the gradient with respect to temperature, which is

$$\sum_{i=1}^{N} \mathbf{w}_i'^T \Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})} R^{-1} \mathbf{w}_i, \tag{A.45}$$

where $w'_{i,k} = C_1 C_2 e^{C_2/(\lambda_k T)} / (\lambda_k^6 T^2 (e^{C_2/(\lambda_k T)} - 1)^2)$.

For the emissivity gradient, the derivatives with respect to ϵ_ℓ (the same emissivity element), ϵ_m (a different emissivity element), and T (temperature) must be found. For the temperature gradient, only the derivatives with respect to ϵ_ℓ and T need to be considered.

First consider the emissivity gradient. As it is broken into three parts, consider each part individually. The results may then be combined in to an answer.

For (A.4), there are no factors with T or ϵ_m present, so those derivatives are zero. The derivative with respect to ϵ_ℓ is

$$\frac{N}{(1-\epsilon_\ell)^2}. \tag{A.46}$$

For the term, (A.13), all three partial derivatives must be considered. For the derivative with respect to ϵ_ℓ, use the product rule, and write the term as

$$-\sum_{i=1}^{N} \partial\left(\frac{(y_{i,\ell} - B(T)_\ell)}{(1-\epsilon_\ell)^2} / \partial\epsilon_\ell\right) \mathbf{e}_\ell^T R^{-1} \mathbf{w}_i + \frac{(y_{i,\ell} - B(T)_\ell)}{(1-\epsilon_\ell)^2} \mathbf{e}_\ell^T R^{-1} \left(\frac{\partial \mathbf{w}_i}{\partial \epsilon_\ell}\right). \tag{A.47}$$

It is relatively straight forward to find the derivative, as it is similar to those that we have calculated when finding the gradient. The result for (A.47) is

$$-\sum_{i=1}^{N}\frac{2(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^3}\mathbf{e}_\ell^T R^{-1}\mathbf{w}_i+\frac{(y_{i,\ell}-B(T)_\ell)^2 R^{-1}_{\ell,\ell}}{(1-\epsilon_\ell)^4}. \tag{A.48}$$

For the derivative with respect to ϵ_m, there is slightly different partial derivatives than in (A.47). Write the term as

$$-\sum_{i=1}^{N}+\frac{(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^2}\mathbf{e}_\ell^T R^{-1}(\frac{\partial \mathbf{w}_i}{\partial \epsilon_m}). \tag{A.49}$$

Again the derivatives are straightforward and the result can be written as

$$-\sum_{i=1}^{N}\frac{(y_{i,\ell}-B(T)_\ell)(y_{i,m}-B(T)_m)R^{-1}_{\ell,m}}{(1-\epsilon_\ell)^2(1-\epsilon_m)^2}. \tag{A.50}$$

Finally, consider the derivative with respect to T. Again using the product rule, write

$$-\sum_{i=1}^{N}(\partial\frac{(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^2}/\partial T)\mathbf{e}_\ell^T R^{-1}\mathbf{w}_i+\frac{(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^2}\mathbf{e}_\ell^T R^{-1}\frac{\partial \mathbf{w}_i}{\partial T}. \tag{A.51}$$

The derivative is

$$\sum_{i=1}^{N}\frac{w'_{i,\ell}}{(1-\epsilon_\ell)^2}\mathbf{e}_\ell^T R^{-1}\mathbf{w}_i+\frac{(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^2}\mathbf{e}_\ell^T R^{-1}\Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})}\mathbf{w}'_i. \tag{A.52}$$

For the temperature portion of the Hessian, begin by considering the derivative of the gradient with respect to T. This is written as

$$\sum_{i=1}^{N}\mathbf{w}_i''^T\Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})}R^{-1}\mathbf{w}_i+\mathbf{w}_i'^T\Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})}R^{-1}\Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})}\mathbf{w}'_i, \tag{A.53}$$

where $\mathbf{w}_i''$ is the sum of

$$\frac{-C_1C_2^2e^{\frac{C_2}{\lambda_k T}}}{\lambda_k^7T^4(e^{C_2/(\lambda_k T)}-1)^2}, \tag{A.54}$$

$$\frac{-2C_1C_2e^{\frac{C_2}{\lambda_k T}}}{\lambda_k^6T^3(e^{C_2/(\lambda_k T)}-1)^2}, \tag{A.55}$$

and

$$\frac{C_1 C_2^2 e^{\frac{2C_2}{\lambda_k T}}}{\lambda_k^7 T^4 (e^{C_2/(\lambda_k T)} - 1)^3}. \tag{A.56}$$

For the partial with respect to ϵ_ℓ, apply the product rule, with the derivative terms being written as

$$\sum_{i=1}^{N} \mathbf{w}_i'^T (\frac{\partial \Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})}}{\partial \epsilon_\ell}) R^{-1} \mathbf{w}_i + \mathbf{w}_i'^T \Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})} R^{-1} \frac{\partial \mathbf{w}_i}{\partial \epsilon_\ell}. \tag{A.57}$$

All of these terms are straightforward and the result is

$$\sum_{i=1}^{N} \frac{1}{(1-\epsilon_\ell)^2} \mathbf{w}_i'^T E_{\ell,\ell} R^{-1} \mathbf{w}_i + \frac{y_{i,\ell} - B(T)_\ell}{(1-\epsilon_\ell)^2} \mathbf{w}_i'^T \Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})} R^{-1} \mathbf{e}_\ell. \tag{A.58}$$

The Hessian matrix is then formed by placing the derivatives in the appropriate indices corresponding to the partials taken for the first and second derivatives. Index (j,k) corresponds to ϵ_j for the first derivative and ϵ_k for the second for j and k less than $n+1$ and index $n+1$ corresponding to T.

A.4 Noisy Hessian

When calculating the Hessian, two terms must be considered. The first is the gradient with respect to ϵ_ℓ, which is the sum of (A.24), (A.30), and (A.44). The second term is the gradient with respect to temperature, which is

$$\sum_{i=1}^{N} \mathbf{w}_i'^T \Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})} X^{-1} \mathbf{w}_i, \tag{A.59}$$

where $w'_{i,k} = C_1 C_2 e^{C_2/(\lambda_k T)} / (\lambda_k^6 T^2 (e^{C_2/(\lambda_k T)} - 1)^2)$.

For the emissivity gradient, the derivatives with respect to ϵ_ℓ (the same emissivity element), ϵ_m (a different emissivity element), and T (temperature) must be found. For the temperature gradient, only the derivatives with respect to ϵ_ℓ and T need to be considered.

First consider the emissivity gradient. As it is broken into three parts, each part individually will be considered individually and then combined to form the final answer.

For A.24, there are no factors with T or ϵ_m present, so those derivatives are zero. The derivative with respect to ϵ_ℓ is

$$\frac{N}{(1-\epsilon_\ell)^2}. \tag{A.60}$$

For (A.30), first note that there are no factors of T and that this derivative will be zero for this case.

For the derivative with respect to ϵ_ℓ, start by breaking up the two pieces to be considered as

$$-\frac{N\sigma^2}{(1-\epsilon_\ell)^3}\frac{\partial X^{-1}_{\ell,\ell}}{\partial \epsilon_\ell} - N\sigma^2 X^{-1}_{\ell,\ell}\frac{\partial 1/(1-\epsilon_\ell)^3}{\partial \epsilon_\ell}. \tag{A.61}$$

The second term is straightforward, the result being

$$-\frac{3N\sigma^2 X^{-1}_{\ell,\ell}}{(1-\epsilon_\ell)^4}. \tag{A.62}$$

The first term is somewhat more complicated. Considering only the derivative portion, this can be expanded as

$$\frac{\partial X^{-1}_{\ell,\ell}}{\partial \epsilon_\ell} = \sum_{j=1}^{n}\sum_{k=1}^{n}\left(\frac{\partial X}{\partial \epsilon_\ell}\right)_{j,k}\left(\frac{\partial X^{-1}_{\ell,\ell}}{\partial X}\right)_{j,k}. \tag{A.63}$$

Note that in the first factor, only diagonal elements have non-zero values and the (j,j)th element corresponds to ϵ_j. As such, only the (ℓ,ℓ)th element will have a non-zero derivative, collapsing the sums. The resulting derivative is then

$$\frac{2N\sigma^4(X^{-T}E_{\ell,\ell}X^{-T})_{\ell,\ell}}{(1-\epsilon_\ell)^6}. \tag{A.64}$$

Combining (A.64) and (A.62), the derivative of (A.30) with respect to ϵ_ℓ is found to be

$$\frac{2N\sigma^4(X^{-T}E_{\ell,\ell}X^{-T})_{\ell,\ell}}{(1-\epsilon_\ell)^6} - \frac{3N\sigma^2 X^{-1}_{\ell,\ell}}{(1-\epsilon_\ell)^4}. \tag{A.65}$$

For the derivative of (A.30) with respect to ϵ_m, only the $X^{-1}_{\ell,\ell}$ factor can be a function of ϵ_m. This essentially makes this derivative similar to the first term above, except that the mth element is

being considered, so the result is

$$\frac{2N\sigma^4(X^{-T}E_{\ell,\ell}X^{-T})_{m,m}}{(1-\epsilon_\ell)^3(1-\epsilon_m)^3}. \tag{A.66}$$

For the final term, (A.44), all three partial derivatives must be considered. For the derivative with respect to ϵ_ℓ, use the product rule, and write the first term as

$$\begin{aligned}-\sum_{i=1}^{N}\partial(\frac{(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^2}/\partial\epsilon_\ell)\mathbf{e}_\ell^T X^{-1}\mathbf{w}_i+\frac{(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^2}\mathbf{e}_\ell^T(\frac{\partial X^{-1}}{\partial\epsilon_\ell})\mathbf{w}_i\\ +\sum_{i=1}^{N}\frac{(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^2}\mathbf{e}_\ell^T X^{-1}(\frac{\partial\mathbf{w}_i}{\partial\epsilon_\ell}),\end{aligned} \tag{A.67}$$

and the second term as

$$\begin{aligned}\sum_{i=1}^{N}(\partial\frac{\sigma^2}{(1-\epsilon_\ell)^3}/\partial\epsilon_\ell)\mathbf{w}_i^T X^{-1}E_{\ell,\ell}X^{-1}\mathbf{w}_i+\frac{2\sigma^2}{(1-\epsilon_\ell)^3}(\frac{\partial\mathbf{w}_i^T}{\partial\epsilon_\ell})X^{-1}E_{\ell,\ell}X^{-1}\mathbf{w}_i\\ +\frac{2\sigma^2}{(1-\epsilon_\ell)^3}\mathbf{w}_i^T(\frac{\partial X^{-1}}{\partial\epsilon_\ell})E_{\ell,\ell}X^{-1}\mathbf{w}_i,\end{aligned} \tag{A.68}$$

where the factors of two come from the symmetry in the vector/matrix product.

For the both sets of terms, they are relatively straight forward derivatives that were calculated when finding the gradient. The result for (A.67) is

$$-\sum_{i=1}^{N}\frac{2(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^3}\mathbf{e}_\ell^T X^{-1}\mathbf{w}_i-\frac{2\sigma^2(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^5}\mathbf{e}_\ell^T X^{-1}E_{\ell,\ell}X^{-1}\mathbf{w}_i+\frac{(y_{i,\ell}-B(T)_\ell)^2X_{\ell,\ell}^{-1}}{(1-\epsilon_\ell)^4}, \tag{A.69}$$

and for (A.68) is

$$\begin{aligned}\sum_{i=1}^{N}\frac{3\sigma^2}{(1-\epsilon_\ell)^4}\mathbf{w}_i^T X^{-1}E_{\ell,\ell}X^{-1}\mathbf{w}_i+\frac{2\sigma^2(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^5}\mathbf{e}_\ell^T X^{-1}E_{\ell,\ell}X^{-1}\mathbf{w}_i\\ -\frac{4\sigma^4}{(1-\epsilon_\ell)^6}\mathbf{w}_i^T X^{-1}E_{\ell,\ell}X^{-1}E_{\ell,\ell}X^{-1}\mathbf{w}_i.\end{aligned} \tag{A.70}$$

Then, the total derivative of (A.44) with respect to ϵ_ℓ is the sum of (A.69) and (A.70).

For the derivative with respect to ϵ_m, there is a slightly different partial derivatives than in (A.67) and (A.68). Write the first term as

$$-\sum_{i=1}^{N}\frac{(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^2}\mathbf{e}_\ell^T(\frac{\partial X^{-1}}{\partial\epsilon_m})\mathbf{w}_i+\frac{(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^2}\mathbf{e}_\ell^TX^{-1}(\frac{\partial\mathbf{w}_i}{\partial\epsilon_m}), \tag{A.71}$$

and the second term as

$$\sum_{i=1}^{N}\frac{2\sigma^2}{(1-\epsilon_\ell)^3}(\frac{\partial\mathbf{w}_i^T}{\partial\epsilon_m})X^{-1}E_{\ell,\ell}X^{-1}\mathbf{w}_i+\frac{2\sigma^2}{(1-\epsilon_\ell)^3}\mathbf{w}_i^T(\frac{\partial X^{-1}}{\partial\epsilon_m})E_{\ell,\ell}X^{-1}\mathbf{w}_i. \tag{A.72}$$

Again the derivatives are straightforward and the results can be written as

$$-\sum_{i=1}^{N}-\frac{2\sigma^2(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^2(1-\epsilon_m)^3}\mathbf{e}_\ell^TX^{-1}E_{m,m}X^{-1}\mathbf{w}_i+\frac{(y_{i,\ell}-B(T)_\ell)(y_{i,m}-B(T)_m)X^{-1}_{\ell,m}}{(1-\epsilon_\ell)^2(1-\epsilon_m)^2}, \tag{A.73}$$

and

$$\sum_{i=1}^{N}\frac{2\sigma^2(y_{i,m}-B(T)_m)}{(1-\epsilon_\ell)^3(1-\epsilon_m)^2}\mathbf{e}_m^TX^{-1}E_{\ell,\ell}X^{-1}\mathbf{w}_i-\frac{4\sigma^4}{(1-\epsilon_\ell)^3(1-\epsilon_m)^3}\mathbf{w}_i^TX^{-1}E_{m,m}X^{-1}E_{\ell,\ell}X^{-1}\mathbf{w}_i. \tag{A.74}$$

So the total partial derivative of (A.44) with respect to ϵ_m is the sum of (A.73) and (A.74).

Finally, consider the derivative with respect to T. Again using the product rule, write

$$\begin{aligned}-\sum_{i=1}^{N}(\partial\frac{(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^2}/\partial T)\mathbf{e}_\ell^TX^{-1}\mathbf{w}_i+\frac{(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^2}\mathbf{e}_\ell^TX^{-1}\frac{\partial\mathbf{w}_i}{\partial T}\\-\frac{2\sigma^2}{(1-\epsilon_\ell)^3}(\frac{\mathbf{w}_i^T}{\partial T})X^{-1}E_{\ell,\ell}X^{-1}\mathbf{w}_i.\end{aligned} \tag{A.75}$$

The derivative is

$$\begin{aligned}\sum_{i=1}^{N}\frac{w'_{i,\ell}}{(1-\epsilon_\ell)^2}\mathbf{e}_\ell^TX^{-1}\mathbf{w}_i+\frac{(y_{i,\ell}-B(T)_\ell)}{(1-\epsilon_\ell)^2}\mathbf{e}_\ell^TX^{-1}\Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})}\mathbf{w}'_i\\-\frac{2\sigma^2}{(1-\epsilon_\ell)^3}\mathbf{w}_i'^T\Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})}X^{-1}E_{\ell,\ell}X^{-1}\mathbf{w}_i.\end{aligned} \tag{A.76}$$

For the temperature portion of the Hessian, start by considering the derivative of the gradient

with respect to T. Since the change is the covariance matrix and this introduces no factors that are functions of T, the same derivative as before is obtained, with the change of the X including the noise. This is written as

$$\sum_{i=1}^{N} \mathbf{w}_i''^T \Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})} X^{-1} \mathbf{w}_i + \mathbf{w}_i'^T \Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})} X^{-1} \Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})} \mathbf{w}_i', \tag{A.77}$$

where $\mathbf{w}_i''$ is the sum of

$$\frac{-C_1 C_2^2 e^{\frac{C_2}{\lambda_k T}}}{\lambda_k^7 T^4 (e^{C_2/(\lambda_k T)} - 1)^2}, \tag{A.78}$$

$$\frac{-2C_1 C_2 e^{\frac{C_2}{\lambda_k T}}}{\lambda_k^6 T^3 (e^{C_2/(\lambda_k T)} - 1)^2}, \tag{A.79}$$

and

$$\frac{C_1 C_2^2 e^{\frac{2C_2}{\lambda_k T}}}{\lambda_k^7 T^4 (e^{C_2/(\lambda_k T)} - 1)^3}. \tag{A.80}$$

The partial with respect to ϵ_ℓ is a little more complex, although it follows a similar path to the derivative without the noise and to the derivatives above. Apply the product rule, with the derivative terms being written as

$$\sum_{i=1}^{N} \mathbf{w}_i'^T \left(\frac{\partial \Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})}}{\partial \epsilon_\ell}\right) X^{-1} \mathbf{w}_i + \mathbf{w}_i'^T \Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})} \left(\frac{\partial X^{-1}}{\partial \epsilon_\ell}\right) \mathbf{w}_i + \mathbf{w}_i'^T \Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})} X^{-1} \frac{\partial \mathbf{w}_i}{\partial \epsilon_\ell}. \tag{A.81}$$

All of these terms are straightforward and the result is

$$\begin{aligned} \sum_{i=1}^{N} \frac{1}{(1-\epsilon_\ell)^2} \mathbf{w}_i'^T E_{\ell,\ell} X^{-1} \mathbf{w}_i - \frac{2\sigma^2}{(1-\epsilon_\ell)^3} \mathbf{w}_i'^T \Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})} X^{-1} E_{\ell,\ell} X^{-1} \mathbf{w}_i \\ + \frac{y_{i,\ell} - B(T)_\ell}{(1-\epsilon_\ell)^2} \mathbf{w}_i'^T \Lambda_{\boldsymbol{\epsilon}/(1-\boldsymbol{\epsilon})} X^{-1} \mathbf{e}_\ell. \end{aligned} \tag{A.82}$$

The Hessian matrix is then formed by placing the derivatives in the appropriate indices corresponding to the partials taken for the first and second derivatives. Index (j,k) corresponds to ϵ_j for the first derivative and ϵ_k for the second for j and k less than $n+1$ and index $n+1$ corresponding to T.

CURRICULUM VITAE

David A. Neal

Published Conference Papers

- Correlated maximum Likelihood Temperature/Emissivity Separation of Hyperspectral Images, David Neal, Todd Moon, Jake Gunther, and Gus Williams, in *2015 49th Asilomar Conference on Signals, Systems, and Computers*, November 2015.

- Performance of Maximum Likelihood Temperature/Emissivity Separation of Hyperspectral Images With Correlated Gaussian Downelling Radiance, David Neal, Todd Moon, Jake Gunther, and Gus Williams, in *2016 50th Asilomar Conference on Signals, Systems, and Computers*, November 2016.