Parameter Sensitivity and Tuning of Piektuk-D

by

Jason Schneider

A masters research paper
presented to the University of Waterloo
in fulfillment of the
masters research paper requirement for the degree of
Master of Mathematics
in
Computational Mathematics

Waterloo, Ontario, Canada, 2019

Examining Committee Membership

The following served on the Examining Committee for this masters research paper.

Supervisor(s): Chris Fletcher
Associate Professor, Dept. of Geography and
Environmental Management, University of Waterloo

Second Reader: Sander Rhebergen
Assistant Professor, Dept. of Applied Mathematics,
University of Waterloo

I hereby declare that I am the sole author of this masters research paper. This is a true copy of the masters research paper, including any required final revisions, as accepted by my examiners.

I understand that my masters research paper may be made electronically available to the public.

Abstract

Blowing snow is an often overlooked phenomenon with potentially large applications for climate and weather modelling. When snow begins to blow, it can clump together into snow drifts, and the increased surface area can cause snow to sublimate, especially in drier conditions. An accurate model of blowing snow could be useful for climate models as the amount and distribution of snow on the ground can change the albedo and the energy balance at the Earth's surface. Blowing snow can also impair visibility. The purpose of this study was to analyze Piektuk-D, a blowing snow model. Piektuk-D is a two-dimensional (time and height) column-based model that treats blowing snow behaviour as the sum of turbulent diffusion under the influence of wind, settling under the effect of gravity, and sublimation. Three input parameters were identified, the particle size distribution's shape parameter, a linear scaling factor determining the impact of wind speed on the turbulent diffusivity of blowing snow, and the exponent determining the order of the relationship between wind speed and the turbulent diffusivity of blowing snow. The sensitivity of Piektuk-D's output to those input parameters was tested and it was found that Piektuk-D was sensitive to every tested parameter, as well as to interactions between them. It was notably more sensitive to changes in the exponent though. Piektuk-D was also tested against some blowing snow data from Wyoming. It was found that the MSE in predicting how many particles there were at certain heights above the ground could be improved from 0.87 in the default state of the model to 0.0787 by altering the model's parameters. The generalizability of these results could not be tested as only one data set was available, however parameter combinations were found with similar MSE that generated similar values of sublimation and vertical transport to the model's default settings.

Acknowledgements

I would like to thank my research supervisor, Dr. Chris Fletcher, for all of his help and guidance along the way. I would also like to thank Dr. Stephen Déry for providing an introduction Piektuk and answering any questions that I had about it.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation for Study

Blowing snow is a common phenomenon in areas with snow. In the Arctic and the Antarctic, blowing snow occurs about 6.5% of the time [2]. Once snow is suspended in the air, the increased surface area and mixing with dry air can accelerate sublimation [3]. Observations from NASA researchers have shown that 393 ± 196 Gt of snow sublimates per year over Antarctica due to blowing snow [4]. It has been estimated that somewhere between 10% to 50% of the snow cover in these areas that is returned to the atmosphere is returned by sublimation due to blowing snow events [5]. This large range of uncertainty can partially be attributed to how difficult it is to study blowing snow sublimation as making direct observations of sublimation is very difficult. Also, since snow depth can affect the colour of the surface of the Earth (and therefore surface albedo), an accurate representation of blowing snow transport and sublimation could be helpful in accurately predicting the energy balance at the Earth's surface. Additionally, blowing snow has implications for visibility as, during a blowing snow event, visibility can be impaired.

1.2 Modelling Blowing Snow

Many blowing snow models have been produced over the years, some of which will be discussed here. In the late 1990s to early 2000s, the state-of-the-art blowing snow models were column-based models that treated blowing snow as a one-dimensional, time-evolving column using time, t, and height, z, as independent variables. Some of these models are WINDBLAST, SNOWSTORM, and Piektuk [6] [7] [3]. Piektuk, in particular, has undergone a few revisions. The first version, referred to as Piektuk-S, was a spectral model. Spectral models directly compute how many blowing snow particles are at each height and have time-evolving, discrete size distributions for particles at these heights. WINDBLAST and SNOWSTORM are also spectral models. Later versions of Piektuk (Piektuk-B, Piektuk-D, Piektuk-T) are bulk models. Bulk blowing snow models work with bulk properties like the blowing snow mixing ratio and rely on averaged quantities meant to be representative of each property at each height. They can still calculate the number of particles at each height, though they make stronger assumptions about the radii of the particles. The blowing snow mixing ratio is the mass of snow in a grid cell divided by the mass of air in the cell, measured in kg/kg.

A commonality amongst blowing snow models is the assumption that blowing snow particle radius is gamma distributed in the saltation layer [8]. The saltation layer is the area right above the ground where particles are moving but are not quite suspended yet. In terms of the models, the saltation layer is the first grid box above the ground with the suspension layer being every box above that. While in the saltation layer, particles can be thought of as “hopping” around on the surface. Past that point, spectral models have a size bin for each height and use discrete, time-evolving particle size distributions. Each bin represents the particles at a range of radii, e.g. 0.5-0.7 mm for one bin, 0.7-0.9 mm for the next. Bulk models assume that blowing snow particle radii are gamma distributed everywhere and recalculate at least one of the distribution's parameters at each height. The assumption that blowing snow particles are gamma distributed in the saltation layer is commonly found in scientific literature and has been tested against observations several times [9] [10]. The gamma distribution has two parameters which will be called α and β in this paper. Here, the particle size distribution, $f(r)$, is defined to be:

$$f(r) = \frac{1}{\Gamma(\alpha)\beta^{\alpha}} r^{\alpha-1} e^{\frac{-r}{\beta}} \tag{1.1}$$

where r is the radius of the particle, Γ is the gamma function, β is the scale parameter and α is the shape parameter of the distribution. From Figure (1.1) it should be clear that increasing α increases both the mean and deviation of the distribution.

In assuming that blowing snow particles are gamma distributed everywhere, bulk models allow the β parameter to vary with height. The equation below shows how β is calculated:

$$\beta = \frac{1}{2}\left(\frac{\rho q_b}{4\pi\rho_{ice}N}\right)^{\frac{1}{3}} \tag{1.2}$$

where ρ is the density of the air-snow mixture at a certain height, q_b is the blowing snow mixing ratio (the mass of suspended snow divided by the mass of air), ρ_{ice} is the density of ice, N is the total number of suspended particles at that height.

The interpretation of Equation (1.1) is that it takes a radius, r, as input and outputs the proportion of blowing snow particles that are at that radius. The output of the distribution multiplied by the total number of suspended particles would be the number of suspended particles with radius r.

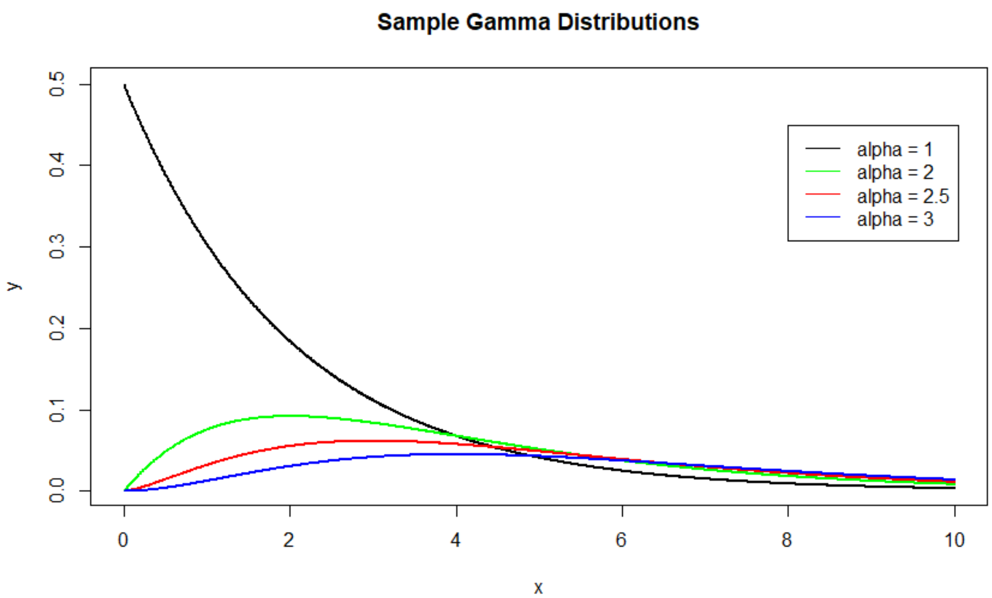

Figure 1.1: Four separate gamma distributions with $\beta = 1$ and α varying. [1]

Every spectral model mentioned uses the same formula for calculating sublimation. Here, the spectral models' method of calculating sublimation, Q_{subl}, is defined:

$$Q_{subl} = \int_0^\infty \int_0^\infty N(z) f(r,z) \frac{dm}{dt} dr dz \tag{1.3}$$

where Q_{subl} is the total amount of snow that has sublimated, $N(z)$ is the number of particles at height z, $f(r,z)$ is the particle size distribution at height z evaluated at radius r, and $\frac{dm}{dt}$ is the change in mass of one particle due to sublimation, which is shown in Equation (1.4).

The rate of change of mass of a single blowing snow particle, $\frac{dm}{dt}$, is given by:

$$\frac{dm}{dt} = \frac{2\pi r(R_h - 1)}{\frac{L_s}{K N_{Nu} T_a}\left(\frac{L_s}{R_v T_a}\right) + \frac{R_v T_a}{N_{Sh} D e_i}} \tag{1.4}$$

where r is the radius of the particle, R_h is the relative humidity with respect to ice, L_s is the latent heat of sublimation of snow, K is the thermal conductivity of air, N_{Nu} is the Nusselt number, calculated as a parametrization of the Reynolds number, T_a is the ambient air temperature, R_v is the gas constant for water vapour, N_{Sh} is the Sherwood number (taken to be equal to the Nusselt number in this case), D is the molecular diffusivity of water vapour in air (a constant), e_i is the saturation water vapour pressure with respect to ice at the ambient temperature T_a (it is a function of T_a). Each model may use a different parametrization for the Nusselt or Sherwood numbers, but otherwise the spectral models calculate sublimation using Equations (1.3) and (1.4). Equation (1.4) was derived in 1966 by Thorpe and Mason [11].

Bulk models like newer versions of Piektuk do not use Equation (1.3) to calculate sublimation. The bulk models use a similar equation, also derived from Thorpe and Mason, which defines the sublimation, S_b, as follows:

$$S_b = \frac{1}{\rho} \int_0^\infty f(r) \frac{dm}{dt} dr \tag{1.5}$$

where S_b is the mass of snow that has sublimated, $f(r)$ is the gamma distribution, $\frac{dm}{dt}$ is the change in mass over time due to sublimation, and ρ is the density of ice. The $\frac{dm}{dt}$ in Equation (1.3) is different from the one that the spectral models use. It is given by Equation (1.6):

$$\frac{dm}{dt} = \frac{2\pi N_u(\frac{q_v}{q_{is}} - 1)}{F_k + F_d} \tag{1.6}$$

where N_u is the Nusselt number, q_v is the water vapour mixing ratio, q_{is} is the saturation water vapour mixing ratio with respect to ice (the water vapour mixing ratio at which sublimation would stop occurring), F_k and F_d represent respectively conduction and diffusion involved in phase changes between snow and water vapour. From here, sublimation is taken as a single integral in accordance with Equation (1.5).

The key difference between the methods utilized by spectral and bulk models to calculate sublimation is that the spectral models use a unique, discrete particle size distribution for each height and a double integral as shown in Equations (1.3) and (1.4). The bulk models use a gamma distribution at every height and a single integral as per Equations (1.5) and (1.6). This leads to sublimation being calculated much faster for bulk models.

Piektuk, WINDBLAST, and SNOWSTORM all model the effect of wind upon snow as turbulent diffusion, though their ways of modelling turbulent diffusion vary. Piektuk-S and WINDBLAST model turbulent diffusion in the same way [3] [7]. The number density for particles in grid cell i, F_i, is defined here:

$$\frac{\partial F_i}{\partial t} = \frac{\partial}{\partial z}\left[K_s \frac{\partial F_i}{\partial z}\right] \tag{1.7}$$

where F_i is the is the number density of particles in grid cell i, i represents index of the grid cell, t is time, z is height, and K_s is the eddy diffusivity, a constant derived differently for Piektuk-S and WINDBLAST. Equation (1.7) was derived from a result by Shiotani and Arai [12]. Equation (1.7) is very similar to the one-dimensional heat equation.

SNOWSTORM handles turbulent diffusion differently from the other models because it uses different prognostic equations with a different origin. SNOWSTORM's prognostic equations were derived from the Navier-Stokes equations rather than the Shiotani and Arai result [6]. Instead of explicitly calculating the particle number density like WINDBLAST and Piektuk-S, SNOWSTORM works with particle drift density and uses particle mass instead of number [6]. Below is SNOWSTORM's equivalent to Equation (1.7), which is one of SNOWSTORM's four prognostic equations. It defines η_r, the mass concentration of blowing snow particles of radius r:

$$\rho\frac{\partial\eta_r}{\partial t} = -\frac{\partial}{\partial z}(\rho\omega'\eta_r') + \rho\left(\frac{\partial\eta_r}{\partial t}\right)_{sub} \tag{1.8}$$

where η_r is the mass concentration of particles of radius r, ρ is the total density of the grid cell (accounting for air and snow), ω' is an operator that gets the vertical turbulent flux of the quantity next to it, $\omega' = K_x\frac{\partial}{\partial z}$, where K_x is the eddy diffusivity for the quantity following ω', and η_r' is η_r where the prime is used to designate that first order closure has been applied, the product $\rho\omega'\eta_r'$ represents the vertical turbulent flux of suspended snow particles, and $\left(\frac{\partial\eta_r}{\partial t}\right)_{sub}$ is the change in mass due to sublimation.

1.3 Piektuk

Piektuk-D was chosen as the subject of this study. Piektuk-D is the double-moment, bulk variant of Piektuk. Being a double-moment model, it explicitly calculates two moments of the particle size distribution, as opposed to the first bulk variant, Piektuk-B, which only calculates one. Part of the reason for choosing Piektuk-D is that Piektuk has seen greater development than other blowing snow models, having undergone three major revisions [13] [14] [15]. Piektuk-D being a bulk model was also a factor. While all of the spectral models have similar runtimes [8], moving from a spectral model to a bulk model made Piektuk roughly 100 times faster [13]. With a typical 48-hour simulation taking around five seconds on the bulk version of Piektuk, a comparable simulation on a spectral model would take about eight minutes. Considering the short time frame of this Masters Research Paper (MRP) of about three months, the shortened runtime of the bulk model was a substantial asset. Had a spectral model been used instead, statistical emulation of the model may have been required to get an adequate sample of the parameter space. Likewise, any problems that may have arisen requiring simulations to be rerun would have been a more substantial set back, especially considering the time allotted for the project.

Chapter 2

Data and Methods

2.1 Piektuk-D

Piektuk-D is a one-dimensional, time-evolving, column-based blowing snow model [14]. Its independent variables are time, t, and height, z. Piektuk-D has four prognostic equations that it uses to model blowing snow. These equations model blowing snow as a combination of turbulent diffusion, settling, and sublimation. The equations are shown below. The blowing snow mixing ratio, q_b, (from [13]) is shown:

$$\frac{\partial q_b}{\partial t} = \frac{\partial}{\partial z}\left(K_b \frac{\partial q_b}{\partial z} + v_b q_b\right) + S_b \tag{2.1}$$

where q_b represents the blowing snow mixing ratio, K_b is the eddy diffusivity for blowing snow, v_b represents the average terminal velocity of blowing snow particles with a given size distribution, and S_b represents sublimation. It is worth noting that S_b will be negative if sublimation is occurring since sublimation will be decreasing the blowing snow mixing ratio.

The water vapour mixing ratio, q_v, (from [13]) is shown:

$$\frac{\partial q_v}{\partial t} = \frac{\partial}{\partial z}\left(K_v \frac{\partial q_v}{\partial z}\right) - S_b \tag{2.2}$$

where q_v is the water vapour mixing ratio, K_v is the eddy diffusivity of water vapour, and S_b is the same sublimation term as in Equation (2.1). In this case, the sublimation term is "$-S_b$" because if sublimation is occurring, S_b itself will be negative and sublimation is a source for water vapour. There is no $v_v q_v$ term because water vapour is assumed not to settle under the effect of gravity.

The prognostic equation for air temperature, T_a, (from [13]) is shown:

$$\frac{\partial T_a}{\partial t} = \frac{\partial}{\partial z}\left(K_h \frac{\partial T_a}{\partial z}\right) + \frac{S_b L_s}{c_p} \tag{2.3}$$

where T_a is the ambient air temperature, K_h is the eddy diffusivity for heat, S_b is the same sublimation term from Equation (2.1) and Equation (2.2), L_s is the latent heat of sublimation of snow and c_p is the heat capacity for air. L_s and c_p are constants that convert the amount of sublimation to the temperature change caused by sublimation.

Finally, the prognostic equation for particle number, N, (from [14]) is shown:

$$\frac{\partial N}{\partial t} = \frac{\partial}{\partial z}\left(K_N \frac{\partial N}{\partial z} + v_N N\right) + S_N \tag{2.4}$$

where N is the particle number, K_N is the eddy diffusivity for particle number, v_N is the terminal velocity for blowing snow particles, S_N is the rate of change of particle number due to sublimation. The method of calculating S_N is skipped here for brevity, though note that S_N depends upon S_b from the previous equations [14].

In more detail, the four prognostic variables are the blowing snow mixing ratio, the water vapour mixing ratio, the ambient air temperature and the particle number. The blowing snow mixing ratio is the amount of snow divided by the amount of air in a grid cell, both measured in kg. The water vapour mixing ratio is the amount of water vapour divided by the amount of air in a grid cell, both of which are also measured in kg. T_a is the ambient air temperature measured in Kelvin. The particle number is a count of the number of blowing snow particles in the grid cell.

The interconnection between Equations (2.1) - (2.4) lies in the sublimation and settling terms. Sublimation, S_b appears in all four prognostic equations for Piektuk-D, excluding the equation for particle number which uses S_N which depends on S_b. S_b, in turn, depends upon the water vapour mixing ratio, q_v, which is represented in Equation (2.2). q_v is used to determine whether sublimation can occur as no sublimation can occur if the air is already saturated with water vapour. It is also used to determine the rate of sublimation. Likewise, the particle size distribution, $f(r)$, appears in both the settling and sublimation terms of the prognostic equations and the β parameter of $f(r)$ depends upon q_b and N as per Equation (1.2).

In addition to the prognostic equations, there are emergent quantities that the prognostic equations do not directly calculate. This includes vertical transport (measured in kg/m), visibility (how far somebody should be able to see given the blowing snow conditions, measured in m), radar reflectivity, and the particle size distribution. As previously mentioned, the particle sizes are assumed to be gamma distributed. Out of the two parameters, α and β, β is recalculated at each height as the model runs whereas α is held fixed at every height throughout the entire run.

To solve the prognostic equations, Fortran 77 code supplied by Piektuk-D's creator, Stephen Déry, was employed [16]. The code is documented on his website and available upon request for non-commercial purposes. In this Fortran 77 code, the prognostic equations are solved using a finite difference scheme with the height variable, z, spaced logarithmically to provide higher resolution near the surface. As input, the code takes temperature, wind speed, humidity, and atmospheric pressure as its meteorological forcings. It outputs visibility, sublimation, and vertical transport. Since particle number was used for validation later on in this study, the code had to be modified to output the particle number at each height. It also needed to be modified to read in more parameters as input. Additionally, for this study, a bash script was written to call the Fortran code multiple times and send it new parameters each time. This bash script is available in Appendix A.4. It reads parameters from a data file holding a list of parameters. It iterates over the length of this list. At each iteration, it modifies the file that the Fortran code reads its parameters from and calls the Fortran code.

In summary, Piektuk-D consists of four main prognostic equations that calculate the blowing snow mixing ratio, q_b, the water vapour mixing ratio, q_v, the particle number, N, and the ambient temperature T_a. These equations depend upon the bulk method of calculating sublimation shown in Equation (1.5) which in turn depends upon the bulk method of calculating the change in mass of a blowing snow particle due to sublimation shown in Equation (1.6). The prognostic equations also depend upon the particle size distribution shown in Equation (1.1) which uses a fixed shape parameter, α, and a varying scale parameter β. β evolves in time at each height according to equation (1.2).

2.2 Parameter Selection

For validation and parameter sensitivity testing, three parameters were selected. These parameters were α, a, and b, where α is the previously mentioned shape parameter of the particle size distribution. α primarily affects how particles settle under the effects of gravity. a and b will be introduced later in this section. a and b impact the relationship between wind speed and the rate of turbulent diffusion.

2.2.1 The Gamma Distribution Shape Parameter, α

The first of the selected parameters is α, which is the shape parameter of the particle size distribution, as shown in Equation (1.1). The particle size distribution factors into the settling velocity, V_B in Equation (2.1). V_B is calculated as follows:

$$V_B = \frac{\int_0^\infty v(r) r^5 f(r) dr}{\int_0^\infty r^5 f(r) dr} \tag{2.5}$$

where $v(r)$ is the terminal velocity for a snow particle of radius r and $f(r)$ is the particle size distribution (gamma distribution) evaluated at radius r. The gamma distribution, $f(r)$ is given by:

$$f(r) = \frac{1}{\Gamma(\alpha)\beta^\alpha} r^{\alpha-1} e^{\frac{-r}{\beta}} \tag{2.6}$$

which is a gamma distribution with shape parameter α and scale parameter β. Some information about α is known. α is typically between 2 and 20 [15]. α should increase with height, however, the formulation of Piektuk-D requires that α is given one value that is held fixed for every height [14]. In choosing which parameter values to test, the available data, as well as the properties of the gamma distribution, were taken into consideration. The range of alpha from 1.1 to 20 was tested, though it was found that values of alpha above 9 caused numerical issues that could not be resolved. The α values tested ended up being in the range of 1.1 to 9. 1.1 was chosen as the minimum because the gamma distribution undergoes a radical change at $\alpha = 1$ where it shifts from predicting no particles of radius 0 to predicting that the majority of particles have radius 0, which is unphysical. α can be calculated directly from validation data. However, since Piektuk-D requires a fixed α, there is uncertainty in which value to use. What has been commonly done in the past is to use the α value calculated for the saltation layer as a representative value for the entire column [8] [15]. No justification for this could be found in the available literature, so α was tuned as a parameter in this study.

α, being a shape parameter, has a large effect on the mean and deviation of the gamma distribution. Increasing it will increase both the average particle radius and the spread of particle radii.

2.2.2 Turbulent Diffusion Parameters

The other two parameters tested appear in the part of the system that handles turbulent diffusion. They are embedded within the eddy diffusivities (K in the equations). The general structure of all four eddy diffusivities that appear in each prognostic equation is very similar. For the below example, allow "G" to represent the particle number, blowing snow mixing ratio, water vapour mixing ratio, or air temperature interchangeably. The general form of an eddy diffusivity, denoted K_G here, is given by:

$$K_G = \zeta_G K_M \tag{2.7}$$

where ζ_G is a constant specific to each prognostic equation (though it is always taken to be 1 in the current Piektuk-D program), and K_M is the eddy diffusivity for momentum, which is the same for all four prognostic equations. This is the same momentum eddy diffusivity from Rouault et al.'s 1991 article on modelling ocean spray droplet dispersion [17]. From here, the eddy diffusivities of all four prognostic equations are identical, since K_M is identical in all four cases. K_M is defined as follows:

$$K_M = l u_* \tag{2.8}$$

where l is the mixing length and u_* is the friction velocity. The mixing length represents how far a clump of snow particles will move under the influence of wind before being broken up. It contains one tunable parameter. This parameter controls mixing length to prevent it from getting too large at high altitudes and therefore does not have a large impact until the mixing length starts to become large. A previous study found that Piektuk is not very sensitive to this parameter [8]. The friction velocity, u_*, contains two tunable parameters as shown below:

$$u_* = aU_{10}^b \tag{2.9}$$

where a and b are the tunable parameters and U_{10} is the wind speed at 10 m above the ground. The a and b parameters are not named quantities. They can both be interpreted as the sensitivity of diffusion to wind speed, where a is a linear scaling factor, and b determines the power of the relationship. Since the dependence on a is purely linear, halving a will halve the amount of diffusion that occurs and doubling a will double the amount of diffusion that occurs. If b is large, then the model becomes more sensitive to small changes in wind speed when the wind speed is large. If b is small, then the model becomes more sensitive to small changes in wind speed when the wind speed is small.

The tested values of a were 0.001 to 0.05. This range was selected because the default value was 0.02264 and the range of 0.001 to 0.05 covers a reasonably wide space around it without requiring the increment on a to be too large. The range of b tested was from 0.3 to 1.75. The default value of b is 1.295. Initially, it was thought that larger values of b could be tested, though that was not possible. The upper end of 1.75 was selected because numerical issues arose with calculating sublimation with higher values of b and cases would arise where there would be substantial blowing snow transport, the air would be subsaturated with humidity, and the model would predict precisely zero millimetres of sublimation. This is clearly wrong, so b was capped at 1.75. 0.3 was chosen as the bottom end of the range to try to cover a large range of parameter values without changing the model's behaviour too substantially. Going below 0 would not make any sense as it would cause smaller wind speeds to bring about more diffusion, with a windless day leading to an undefined level of blowing snow diffusion. Values above one would preserve the superlinear,

Parameter	Description	Min	Max	Default	Units
α	Shape parameter of gamma distribution	1.1	9.0	2	unitless
a	Converts wind speed to diffusivity	0.001	0.05	0.02264	m
b	Order of relationship between wind speed and diffusivity	0.3	1.75	1.295	unitless

Table 2.1: Summary of the three tested parameters

subquadratic dependence that diffusion has on wind speed in the default model, while values below one would change it to a sublinear dependence. Table (2.1) summarizes the selected parameters, gives a short description, the tested range, the default values, and the units.

2.3 Model Validation

The validation data used in this study was gathered in Southeastern Wyoming by R.A. Schmidt [1]. It was collected using a snow catcher that had five snow traps set up at five different heights which caught blowing snow particles as they blew around. This produced a particle flux which could be integrated to determine how many individual blowing snow particles there were at each height. This integrated flux was reported in their paper [1]. A photo of the apparatus used is shown in Figure (2.1).

There are two other blowing snow experiments frequently referenced in the literature

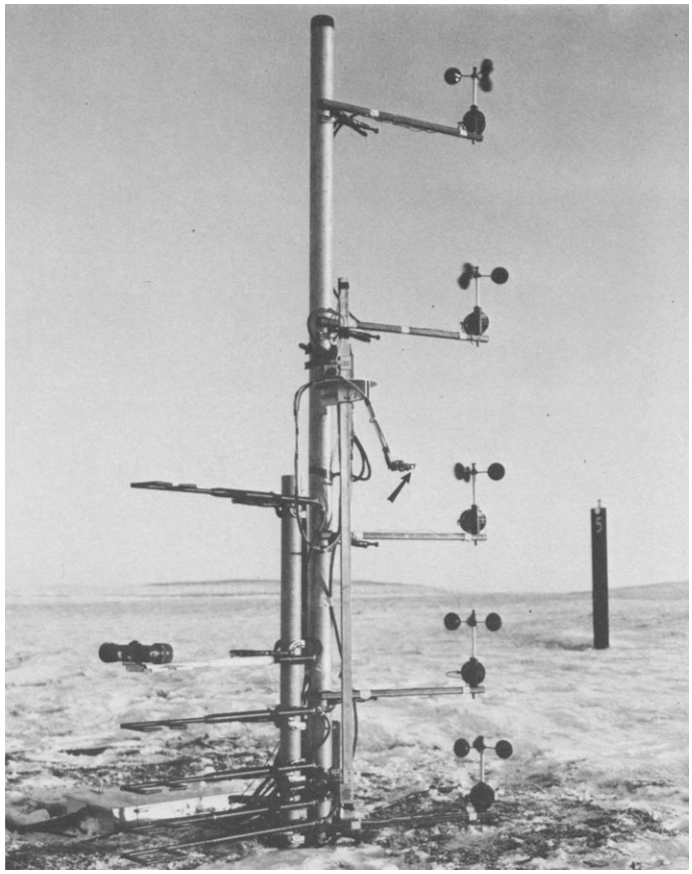

Figure 2.1: The blowing snow particle catcher employed by R.A. Schmidt for their blowing snow study [1]

.

that produced quantitative data that could have been useful here. These are from the Byrd Snow Drift Project [9] and an experiment from the Halley station in Antarctica [18]. While some data is available in these papers, nothing that could be used to validate Piektuk was found there. Unfortunately, in the short timeframe of an MRP, useful data could not be procured from these studies.

The validation data used was run 2 in the Schmidt paper [1]. The other runs could not be used due to either lack of meteorological forcing data or other problems that the author reported in the study. The model was set up with the forcings prescribed in that run and allowed to run for ten minutes, then particle number, sublimation, and total vertical transport were output. This is very similar to how Piektuk-T and Piektuk-D were compared in the original Piektuk-T validation paper [15]. It is worth noting that, in that study, no parameter tuning or sensitivity testing was conducted on either model.

Two metrics were used for validation, mean squared error (MSE) and mean absolute deviation (MAD). These metrics are shown in Equation (2.10), which is the MSE and Equation (2.11), which is the MAD.

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{n} \left(N_i - \hat{N}_i \right)^2 \tag{2.10}$$

where n is the number of data points, N_i is the observed particle number at height i, and $\hat{N}_i$ is the model's predicted particle number at height i.

$$\mathrm{MAD} = \frac{1}{n} \sum_{i=1}^{n} \left| N_i - \hat{N}_i \right| \tag{2.11}$$

where n is the number of data points, N_i is the observed particle number at height i, and $\hat{N}_i$ is the model's predicted particle number at height i.

The practical difference between MSE and MAD is in how much they value having the model output at every height be close to the observed value. MSE will penalize one result being very far from the observed value more than MAD will. For example, a model with four accurate values and one highly inaccurate value will have a much lower MAD than MSE. As such, both will be reported here, though the sensitivity of MSE to cases with one or two highly inaccurate values will likely make it a more useful validation metric.

2.4 Parameter Sensitivity

Parameter sensitivity testing was conducted using the Fourier Amplitude Sensitivity Testing (FAST) method of Saltelli et al. [19]. For this analysis, Iooss et al.'s R package "Sensitivity" was employed [20]. The sampling for the parameter space of the model used for sensitivity testing was uniform as defined by the Sensitivity package using 4000 samples per parameter (12 000 samples total). The FAST method is based around the conditional variance for each of the parameters. The conditional variance of a variable is the amount of variance left once all of the other variables are accounted for. The FAST method produces estimates of how important each parameter is. Embedded within this estimate is how much of the importance is due to the parameter itself and how much is due to its interactions with other parameters.

Chapter 3

Results

Figure (3.1) is a scatterplot matrix showing how the input parameters interact with each other and each output variable. Along the diagonal are the input parameters and output variables. When looking at any plot in Figure (3.1), looking to the right at the diagonal shows which variable is on the y-axis, looking up to the diagonal shows which variable is on the x-axis. In creating Figure (3.1), some values had to be removed. Roughly 2.7% of the data showed extreme sublimation values of greater than 44 000 mm. For a ten minute period, 44 000 mm of sublimation is not physically plausible. The rest of the output was between 0 and 1.15 mm which seemed more reasonable. The parameter combinations leading to these extreme values of sublimation are contained in certain pockets of high α combined with specific values of a. These extreme values represent a small part of the output data and their inclusion would obscure many of the results seen in this section by altering the scales of plots severely. As such, these results have been excluded. In Figure (3.1) the correlation between sublimation and vertical transport is noticeable. Sublimation and vertical transport have a Pearson correlation coefficient of 0.473. However, the relationship does not appear linear and neither sublimation nor transport is normally distributed (this is shown below in Figure (3.3)), which are both assumptions of the Pearson correlation coefficient. The Spearman correlation coefficient between sublimation and transport is 0.797. It is likely more appropriate as it makes no assumptions about the distribution of the data or linearity of the relationship. This correlation is to be expected as higher amounts of vertical transport should be indicative of more blowing snow, which should lead to more sublimation.

The key purpose of Figure (3.1) is to display how the parameters interact with output variables. Both a and b are required to be relatively high, with the threshold for b being notably higher, to output high values of sublimation and transport. However, a high value of one parameter alone is not sufficient to cause high output values. Increasing α can increase sublimation up to a certain point, after which it begins to drop off rapidly. α is also required to be somewhat high (though not too high) to get high values of transport. These effects could be due to α's effect on particle size. In general, α will cause blowing snow particles to become larger and weigh more. Since larger particles can have a longer lifetime when sublimating, sublimation can increase. Likewise, since transport is measured in kg/m, larger particles should lead to more mass being transported and increase vertical transport. After a certain point, the particles should become large enough that the settling effect of gravity is too great for them to move around much, preventing there from being much blowing snow. It is worth noting that this effect would be exclusively due to the settling and sublimation terms in the prognostic equations as the eddy diffusivities (K values) have no dependence on the particle size distribution. The scaling on MSE (with values upwards of 15 000) makes this graph all but useless for finding accurate areas of the model. The "tendril-like" structures that seem to appear in Figure (3.1) where MSE is plotted against sublimation or transport could be due to those being somewhat rare,

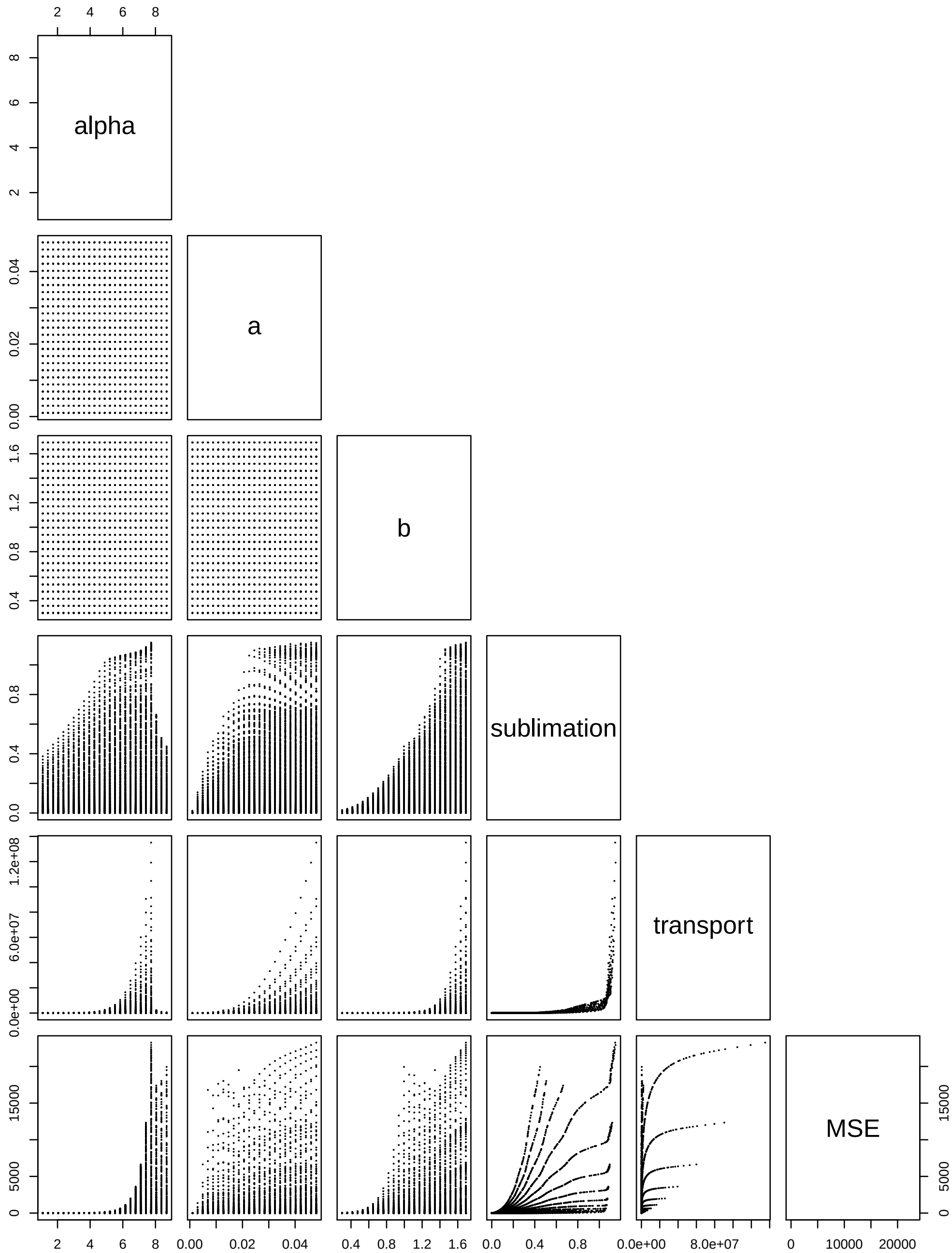

Figure 3.1: How the parameters and response variables interact with each other

extreme values and most of the sample being pushed mostly into one small region on that scale.

3.1 Sensitivity

For sensitivity testing, the extreme sublimation values ($> 40\ 000$ mm) were also left out. This is because the amount of sublimation for these runs was so very different from the others, which were all under 1.15 mm. Since the difference in sublimation between these runs is so extreme (with a gap of four orders of magnitude), including them in the sensitivity tests leads to every variable appearing equally important with the vast majority of the impact of each parameter's influence being in the interaction component. Excluding these values leads to results that somebody would be more likely to encounter when running the model in a way that provides reasonable output. In constructing Figure (3.2), the runs with extreme sublimation were removed.

Figure (3.2) shows that sublimation is most sensitive to b and similarly sensitive to a and α. This could make sense comparing to Figure (3.1) as b appears to have a stronger relationship to sublimation than the other two parameters. All three parameters have a substantial interaction component as well. It would make sense to see a large interaction component referring to Figure (3.1) as there seems to be a large spread in which parameter values can output each sublimation value. Figure (3.2) also shows b as the most important parameter for vertical transport, though a seems to be more important than α. Transport does seem less sensitive to b than sublimation is.

The sensitivity of b may depend upon the actual value of U_{10} in the data set. Recall that $K_M \propto aU_{10}^b$. In the data set used, the value of U_{10} was 14.8. Due to the nonlinear

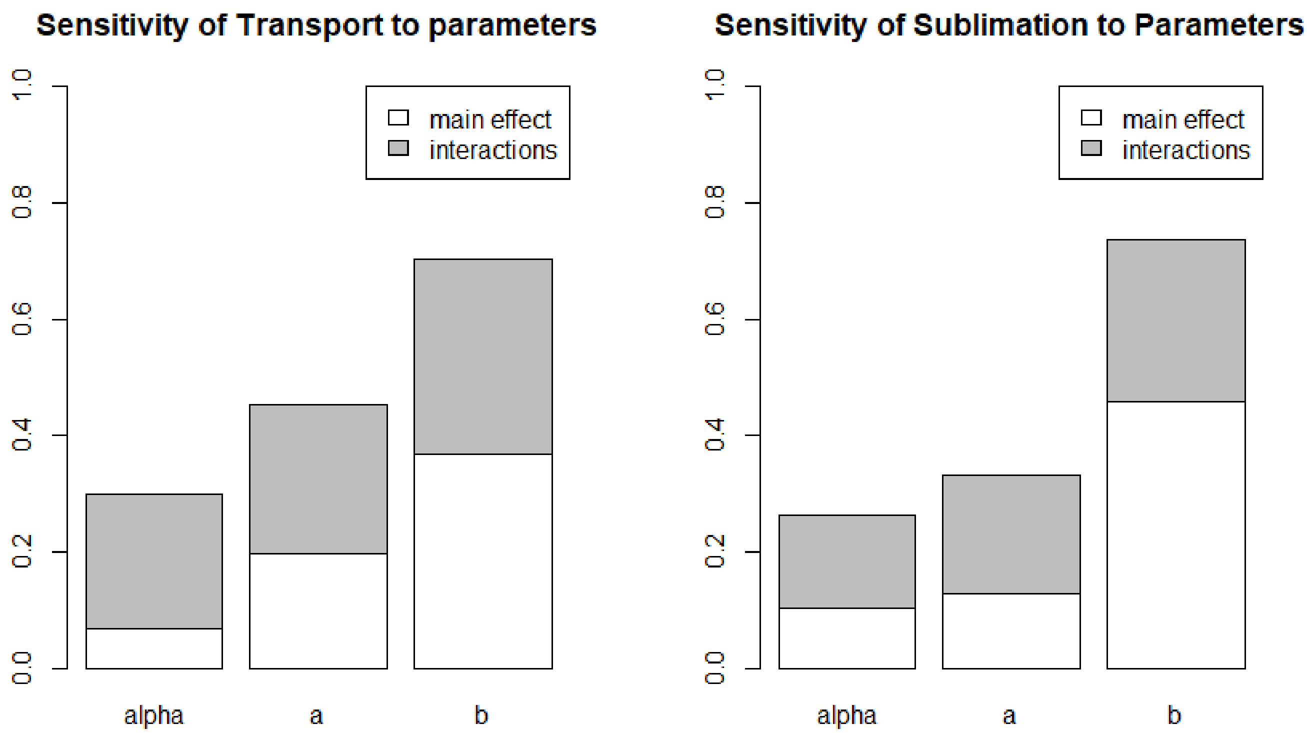

Figure 3.2: Sensitivity of sublimation to the input parameters

nature of b, smaller values of U_{10} may lead to the model becoming less sensitive to b. This was not tested empirically due to the time frame of this MRP.

3.2 Distributions of Outputs

Figure (3.3) shows a kernel density estimate of the distributions of MSE, MAD, sublimation, and vertical transport. In order to display the distributions in a meaningful way, the x-axes had to be truncated. For sublimation, 2.7% of the data had to be removed

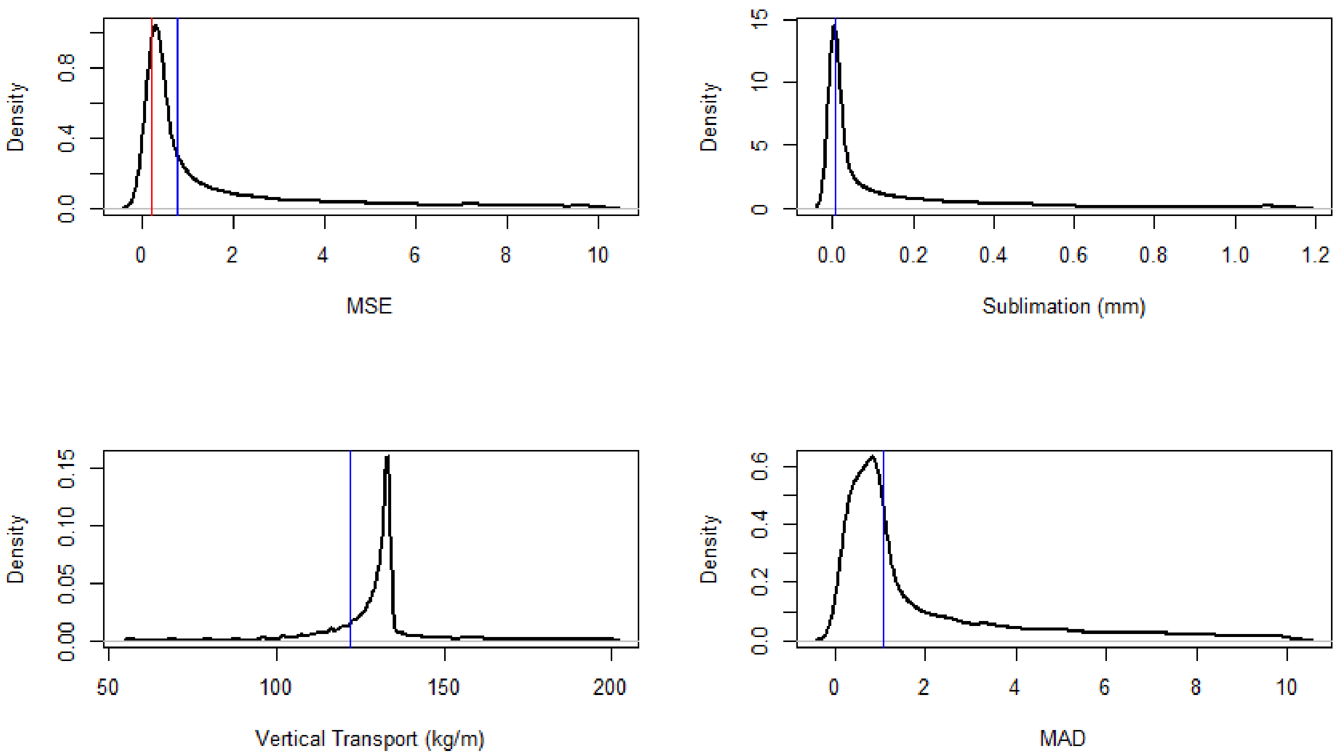

Figure 3.3: The distributions of the outputs of the model as well as the MSE and MAD. The red line in the MSE plot is at 0.15 which is the cut-off point for data to appear in the plots of the previous section. The blue line represents where the model's default parameters lie

since sublimation can go as high as 2 311 090. For MSE, 37% of the data was removed. For MAD, 44% had to be removed. For vertical transport, 40% of the data was removed. While a lot of data was removed, there were no discernible trends in any of it and including it made the plots difficult to read.

The MSE, sublimation, and MAD distributions have a long tail going off to the right, somewhat resembling a log-normal distribution. Vertical transport appears to have a peak around the 120 - 140 range which is notably sharper on the right. The MAD distribution is also noticeably wider than the MSE distribution with a much shallower decline.

3.3 Validation

The top 5% most accurate model runs had an MSE of 0.15 or less, with the lowest MSE being 0.0758. Since 15 625 simulations were used for validation, the top 5% ended up being 781 runs. For MAD, the top 5% model runs had an error of 0.28 or less with the lowest being 0.151.

From Figure (3.4), it is clear that certain parameter combinations have lower MSE than others on the data set used here. To achieve MSE < 0.15 on this data, it is necessary to restrict $\alpha < 5.84$. Since increasing α increases the mean and spread of the particle size distribution, increasing α will cause the mean terminal velocity of blowing snow particles to go up. If α is set too high, it should cause blowing snow particles to be unable to travel very high in the air as they are falling too quickly. This is demonstrated in Figure (3.1) where, once α gets past around 6, the amount of sublimation and transport drops off very fast. As stated in the introduction, values of α are known to change with meteorological conditions as well as height [15]. Since α for this data set was calculated as 3.5 in the saltation layer,

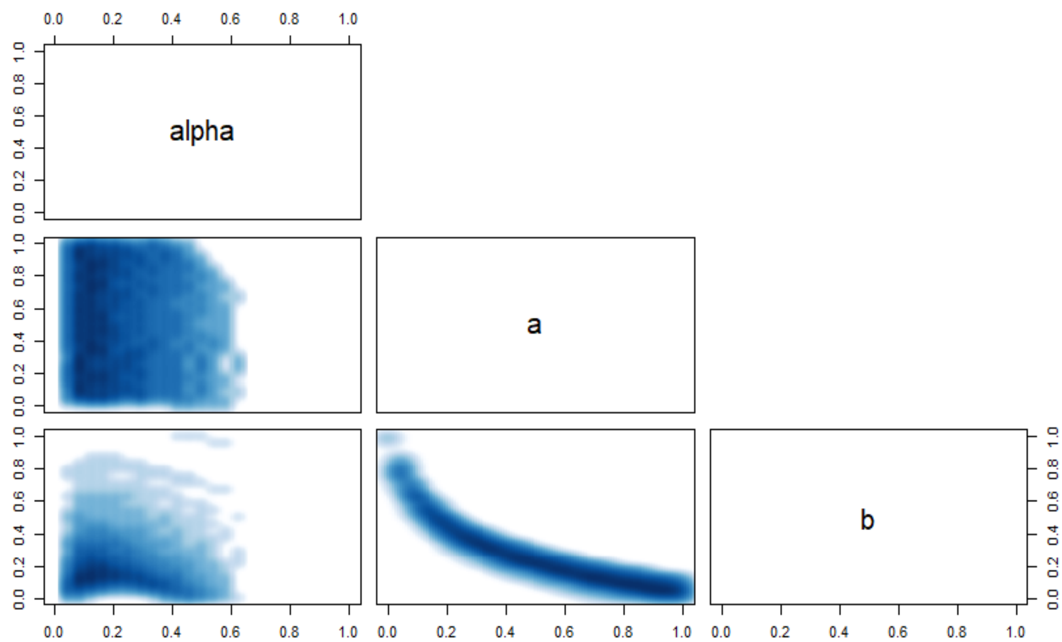

Figure 3.4: The density of cases with MSE < 0.15. Areas shaded with a darker blue have a greater density of accurate cases. Areas that are white have no model runs with MSE < 0.15. In this plot, all of the parameters are scaled to belong to [0,1].

and values of 3.5 or lower appear to have a high density of low MSE outputs, compensating for α's increase with height appears to be unnecessary for this data set.

Nearly all tested values of a and b are capable of achieving MSE< 0.15, though there appears to be a non-linear relationship between them where high values of one parameter require low values of the other. Examining the plot in Figure (3.4) that shows α and b together shows clearly that there are more parameter combinations with low MSE and low values of b than there are with low MSE and high values of b. This sort of relationship between a and b can potentially be explained by the way that the eddy diffusivities depend on them.

One way of interpreting these results is that a and b must compensate for each other to produce accurate outputs. For all four eddy diffusivities, K, $K \propto aU_{10}^b$. This term was examined in greater detail. It was found that for every parameter combination tested, aU_{10}^b varied between 0.00224 and 4.589. However, for the MSE < 0.15 cases, aU_{10}^b only varied between 0.0702 and 0.179. Based on this, it appears that, for the data set used, there is a narrow range of good values for the eddy diffusivities and that choices of a and b must somewhat compensate with each other to produce an overall reasonable eddy diffusivity.

Since there was only one set of data available for validation of the model, two other outputs from the model were examined, sublimation and quantity of vertical transport of blowing snow. Since there was no observational data on sublimation or vertical transport available, the outputs could not be directly validated, though they were compared to the default model. Table (3.1) shows the five runs with the lowest MSE, their parameter values, their MSE (calculated from particle number), and how they compare to the default model.

Table (3.1) shows that the five model runs with the lowest MSE have similar values of α and a variety of a and b values. The sublimation values of the top five model runs are

Run Number	α	a	b	Sublimation (mm)	Transport (kg/m)	MSE	MAD
Default	3.5	0.0255	1.295	0.000738	122	0.83	1.06
1	3.312	0.00492	1.228	0.000747	131	0.0758	0.162
2	3.312	0.0147	0.764	0.000751	132	0.0758	0.231
3	3.312	0.0108	0.880	0.000756	132	0.0758	0.227
4	3.312	0.0441	0.358	0.000758	131	0.0759	0.161
5	3.628	0.0304	0.474	0.000730	132	0.076	0.157

Table 3.1: The results of the top 5 most accurate (by MSE) model runs and how their parameters, sublimation, and vertical transport estimates compare to the default model

close both to each other and the default model. The values of vertical transport of the top five model runs are very close to each other but somewhat different from the default model. For reference, throughout all of the parameter combinations tested, the model's lowest predicted value of sublimation was on the order of 10^{-11} and the highest was 2 311 090 mm. The lowest predicted value of vertical transport was 56 kg/m and the highest was 135 113 824 kg/m. Therefore, the difference in vertical transport between the low MSE runs and the default model is not necessarily negligible, though it is quite small compared to what the model is capable of outputting. The difference is still around 9% though, so it could be indicative of the default model underestimating transport somewhat for this data.

Figure (3.5) shows how every variable, both input and output, relate to each other in the space of low MSE. It is worth noting that the values of vertical transport are quite restricted here. When MSE < 0.15, transport only ranges between 125 kg/m and 133 kg/m. For reference, in the space of MSE < 1, vertical transport can range from 56 kg/m to 3715 kg/m. As such, the plots involving vertical transport here may not be representative of

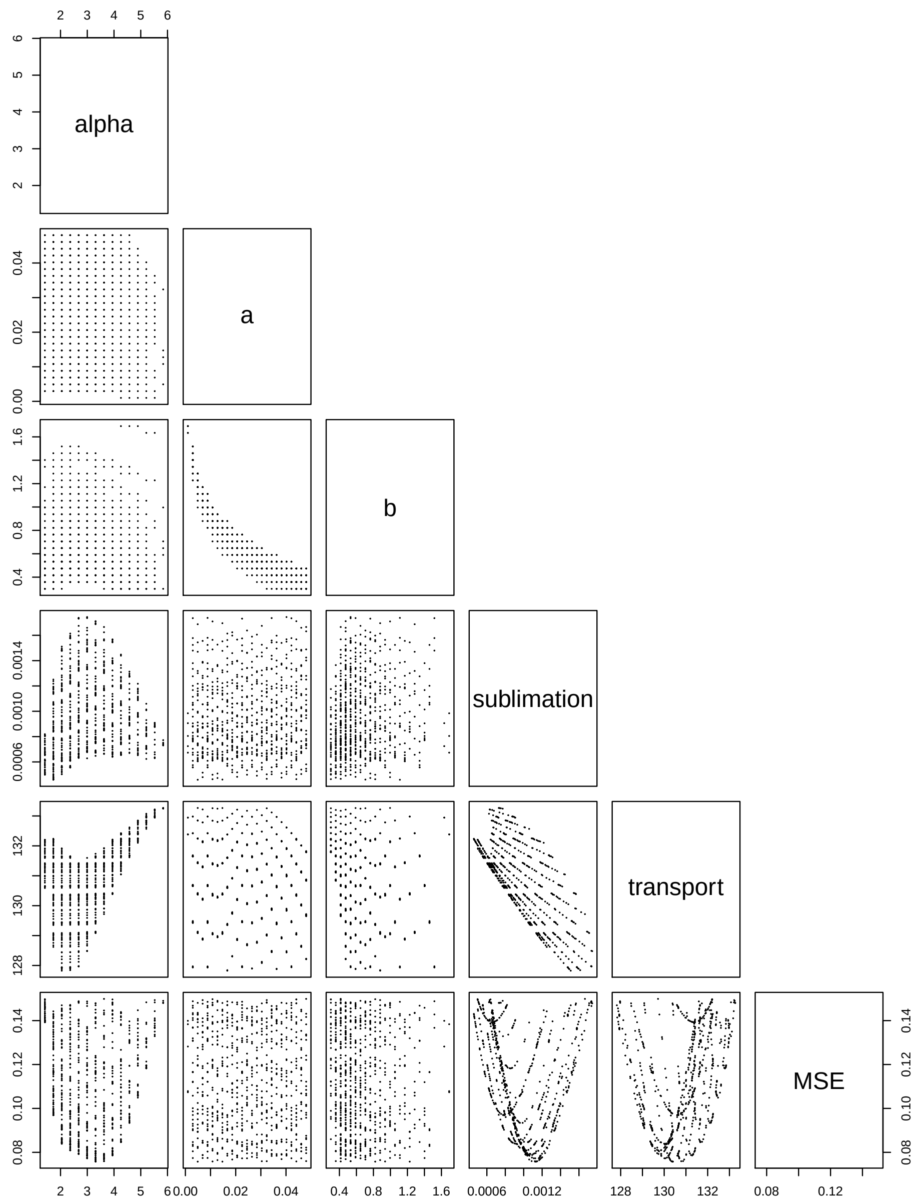

Figure 3.5: The relationship between the different input and output variables of the model in the space of MSE < 0.15

how the model responds in the greater space of outputs that the model is capable of. Interestingly, in Figure (3.5) sublimation and transport appear negatively correlated. This may be transient behaviour in the space of low MSE in general, or just in the space of low MSE for this particular data set. Another thing to note is that a and MSE appear to have little relationship with each other, whereas the plot comparing b to MSE becomes much sparser as b grows. This could be due to eddy diffusivity growing too quickly with large values of b.

Chapter 4

Discussion

4.1 Generalizing the Results

The decrease in MSE from 0.83 in the default model to 0.0758 for the tuned case with the lowest MSE is quite substantial though it does come with some limitations. A notable factor is that only one data set was available for validation. Taking that into account, there was no way to check how well these low MSE parameter combinations would generalize to other cases so there may be substantial overfitting to this data set. One good sign is that the amount of sublimation and vertical transport predicted by low MSE model runs are close to those predicted by the default model, and the default model has been shown to be reasonably accurate with respect to sublimation and overall vertical transport in a few different locations [8] [15]. However, this is far from definitive proof that these low MSE parameter combinations would have low MSE on different data sets.

One potential issue with the generalizability of these results is that each of the chosen parameters may change depending on the data set used. For example, the amount of turbulent diffusion that occurs should depend upon how wet or dense the snow is, which could, in turn, depend on the temperature. However, in the current formulation of the model, the amount of turbulent diffusion of the model only depends upon aU_{10}^{b} with a and b being the tuned parameters and U_{10} being the wind speed at 10 m above the ground. As such, different data sets could have different optimal a and b values depending on the properties of the snow in that area at that time.

Something that may be worthy of further study is the relationship between wind speed and eddy diffusivity. The model's default parameters gave eddy diffusivity a superlinear dependence upon wind speed, $K_M \propto U_{10}^{1.295}$. However, many of the low MSE model runs used a linear or sublinear relationship between the eddy diffusivity and b. The fourth lowest MSE run used a b value of 0.358. These sublinear values were quite common. Referring back to Figure (3.4), it appears that there was a greater density of low MSE outputs near small values of b than near large values of b. No reference could be found in the available literature on Piektuk as to why b was initially selected to be 1.295. That choice may be worth revisiting.

A potential next step for validating Piektuk would be to redo similar analysis and tuning with different data. Hopefully, there would be some non-empty intersection between the areas in the parameter space that are accurate for many different data sets.

4.2 Potential Applications of Piektuk

One potential area of application for Piektuk would be for climate models. As mentioned previously, it has been estimated that somewhere between 10% to 50% of the snow cover in the Arctic and Antarctic is returned to the atmosphere by sublimation due to blowing snow [5]. This is potentially such a substantial effect that if climate models are not treating this directly, they are could be compensating for it in some way through a parametrization, by underestimating snowfall, or through some other mechanism. Especially when there is little snow on the ground, a 10% to 50% change in snow depth could substantially alter albedo.

While it was not mentioned much in this paper, Piektuk-D also provides estimates of visibility and radar reflectivity. Visibility estimates are provided in metres and are supposed to represent how far somebody should be able to see given the blowing snow conditions. Piektuk-D does not account for precipitation when calculating radar reflectivity or visibility. Piektuk-D can be fed meteorological data and get an estimate as to how far somebody should be able to see during a snow storm. Visibility is mostly predicted from radar reflectivity which, due to the prevalence of ground radar, would provide another way to validate the model in general as well as a way to validate the mechanism providing the visibility estimates. A key advantage of Piektuk-D over ground radar is that Piektuk-D could be fed data from a numerical weather prediction model and get an estimate of visibility for the future.

Chapter 5

Conclusion

In this masters research paper, Piektuk-D was evaluated and validated against observational data. It was found that Piektuk-D's assumption that α can be held fixed throughout the column of blowing snow at a value calculated in the saltation layer was fine in the data used. It was also found that Piektuk-D's prediction of particle number could be improved through parameter tuning, though there are notable reasons to believe that these results may not generalize to other cases or may require more tuning for different areas. Through sensitivity testing, the outputs of Piektuk-D were found to be very sensitive to the power of the dependence that eddy diffusivity has on wind speed and less sensitive to α and the linear scaling coefficient of the dependence that eddy diffusivity has on wind speed. There may be the potential for improvement to Piektuk-D by tuning parameters, but the results here are not necessarily robust and more testing with more data sets would be necessary to draw any conclusions with reasonable certainty.

References

[1] R.A. Schmidt. Vertical profiles of wind speed, snow concentration, and humidity in blowing snow. *Boundary Layer Meteorology*, 23:223–246, 1982.

[2] S.J. Déry and M.K. Yau. A climatology of adverse winter-type weather events. *Journal of Geophysical Research*, 104(D14):16657–16672, 1999.

[3] S.J. Déry, P.A. Taylor, and J. Xiao. The thermodynamic effects of sublimating, blowing snow in the atmospheric boundary layer. *Boundary Layer Meteorology*, 89:251–283, 1998.

[4] S.P. Palm, V. Kayetha, Y. Yang, and R. Pauly. Blowing snow sublimation and transport over antarctica from 11 years of calipso observations. *The Cryosphere*, 11:2555–2569, 2017.

[5] G.E. Liston. Representing subgrid snow cover heterogeneities in regional and global models. *Journal of climate*, 17(6):1381–1397, 2004.

[6] R. Bintanja. Snowdrift suspension and atmospheric turbulence. part 1: Theoretical background and model description. *Boundary Layer Meteorology*, 95:369–395, 2000.

[7] G.W. Mann. *Surface Heat and Water Vapour Budgets over Antarctica*. PhD thesis, The University of Leeds, Leeds, UK, 1998.

[8] J. Xiao, R. Bintanja, S.J. Déry, G.W. Mann, and P.A. Taylor. An intercomparison among four models of blowing snow. *Boundary-Layer Meteorology*, 97(1):109–135, 2000.

[9] W.F. Budd, W.R.J. Dingle, and U. Radok. The byrd snow drift project: outline and basic results. In M.J. Rubin, editor, *Studies in Antarctic Meteorology*, volume 9. American Geophysical Union, Washington, USA, 1966.

[10] RA Schmidt. A system that measures blowing snow. *USDA Forest Service Research Paper*, 194:1–80, 1977.

[11] A.D. Thorpe and B.J. Mason. The evaporation of ice spheres and ice crystals. *British Journal of Applied Physics*, 17(4):541, 1966.

[12] M. Shiotani and H. Arai. On the vertical distribution of blowing snow. *Physics of Snow and Ice: proceedings*, 1(2):1075–1083, 1967.

[13] S.J. Déry and M.K. Yau. A bulk blowing snow model. *Boundary Layer Meteorology*, 99:237–251, 1999.

[14] S.J. Déry and M.K. Yau. Simulation of blowing snow in the Canadian arctic using a double-moment model. *Boundary Layer Meteorology*, 99:296–316, 2001.

[15] J. Yang and Yau M.K. A new triple-moment blowing snow model. *Boundary Layer Meteorology*, 126:137–155, 2008.

[16] S.J. Déry. Piektuk blowing snow model. http://hydrology.princeton.edu/~sdery/datafiles/piektuk.htm, 2004. Accessed: 2019-12-11.

[17] R.R. Rouault, P.G. Mestayer, and R. Schiestel. A model of evaporating spray droplet dispersion. *Journal of Geophysical Research*, 96:7181–7200, 1991.

[18] G.W. Mann, P.S. Anderson, and S.D. Mobbs. Profile measurements of blowing snow at halley, antarctica. *Journal of Geophysical Research*, 105:24491–24508, 2000.

[19] A. Saltelli, M Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola. *Global Sensitivity Analysis: The Primer*. Wiley, 2008.

[20] Bertrand Iooss, Alexandre Janon, Gilles Pujol, with contributions from Baptiste Broto, Khalid Boumhaout, Sebastien Da Veiga, Thibault Delage, Jana Fruth, Laurent Gilquin, Joseph Guillaume, Loic Le Gratiet, Paul Lemaitre, Barry L. Nelson, Filippo Monari, Roelof Oomen, Oldrich Rakovec, Bernardo Ramos, Olivier Roustant, Eunhye Song, Jeremy Staum, Roman Sueur, Taieb Touati, and Frank Weber. *sensitivity: Global Sensitivity Analysis of Model Outputs*, 2019. R package version 1.16.1.

APPENDICES

Appendix A

Plotting and Analysis Code

A.1 Introduction

The appendix is mostly just to show the code that was used to do analysis or make plots.

A.2 Sensitivity Testing Code

This code was used for sensitivity testing. It generates the parameters using the sensitivity package in R [20]. From there, the parameters need to be output and loaded into Piektuk-D. Then, the code loads the output in and analyzes/plots it.

```
require(sensitivity)
require(fields)
require(BBmisc)
require(RColorBrewer)
```

```
require(spatstat)
require(stringr)
require(pracma)

#This gets the parameter space
x = fast99(model=NULL,factors=c('alpha','a','b'),q=c('qunif','qunif','qunif'),q.arg = l
list(min=0.001,max=0.05),list(min=0.3,max=1.75)),n=4000)
#If alpha goes above 9, the code bugs out and predicts 0 sublimation
#If b goes above 1.75, it can start introducing NAs with certain
#parameter combinations
#This is the number of runs kept
kept_runs = 12000
#transport and sublimation are preallocated here
transport = matrix(0,kept_runs,1)
sublimation = matrix(0,kept_runs,1)
#This reads in sublimation and transport in the format that
#Piektuk-D creates them
for (i in c(1:kept_runs-1)){
  name = paste('total',i,sep='')
  name = paste(name,'.dat',sep='')
  tmp = read.table(paste('C:\\Users\\Jasn\\Desktop\\Oct30Sims.new\\piektuk2\\',name,sep
  transport[i+1] = tmp[4][[1]]
  #This handles scientific notation. Since R will read the "-" as a "." for some reason
  if (str_detect(substring(colnames(tmp[4])[1],2,500),'E.')){
    sublimation[i+1] = as.double(str_replace(substring(colnames(tmp[4])[1],2,500),'E.',
```

```
  }
  else{
    sublimation[i+1] = as.double(substring(colnames(tmp[4])[1],2,500))
  }
}
```

A.3 Plotting Code

This is the R code used to generate the plots

```
require(stringr)

#kept_runs is the number of model runs that are read in
kept_runs = 15625
#transport and sublimation are preallocated here
transport = matrix(0,kept_runs,1)
sublimation = matrix(0,kept_runs,1)

#This reads in the transport and sublimation from the raw outputs from Piektuk
for (i in c(1:kept_runs)){
  name = paste('total',i,sep='')
  name = paste(name,'.dat',sep='')
  #This points to the directory
  tmp = read.table(paste('C:\\Users\\Jasn\\Desktop\\Full_Sampling\\piektuk2\\',name,sep
```

```
  transport[i+1] = tmp[4][[1]]
  #This handles scientific notation. Since R will read the "-" as a "." for some reason
  if (str_detect(substring(colnames(tmp[4])[1],2,500),'E.')){
    sublimation[i+1] = as.double(str_replace(substring(colnames(tmp[4])[1],2,500),'E.',

  }
  else{
    sublimation[i+1] = as.double(substring(colnames(tmp[4])[1],2,500))
  }
}

#MADev is the Mean Absolute Deviations, read in from the path below
MADev = read.csv('C:\\Users\\Jasn\\Desktop\\Full_Sampling\\EL1.csv',sep=',')
colnames(MADev) = 'error'
MADev = MADev$error
#This reads in the parameters used to run the model
parms = read.csv('C:\\Users\\Jasn\\Desktop\\Full_Sampling\\piektuk2\\parm_holder.dat',s
parmsand = parms
parmsand$sub = sublimation
parmsand$tra = transport

#This reads in the errors since I calculated that in Python
Error = read.csv('C:\\Users\\Jasn\\Desktop\\Full_Sampling\\Error.csv',header=FALSE,sep=
colnames(Error) = c('error')
Error = Error$error
```

```
colnames(parms) = c('alpha','a','b')


#This produces the density plot with whatever condition is used in the line below
#By default, I have it set to Error<0.15, though sublimation or transport can also
#be used
parmsubset = parms[Error<0.15,]
parmsubset2 = parmsubset
parmsubset2$alpha = parmsubset2$alpha-min(parms$alpha)
parmsubset2$alpha = parmsubset2$alpha/max(parms$alpha-min(parms$alpha))
parmsubset2$a = parmsubset2$a-min(parms$a)
parmsubset2$a = parmsubset2$a/max(parms$a-min(parms$a))
parmsubset2$b = parmsubset2$b-min(parms$b)
parmsubset2$b = parmsubset2$b/max(parms$b-min(parms$b))
pairs(parmsubset2,panel=function(x,y,...){smoothScatter(x,y,add=T,nrpoints=0,
    colramp=colorRampPalette(c(rep('white',5),blues9)))},
     upper.panel = NULL,xlim=c(0,1),ylim=c(0,1),main='')


#This creates a similar density plot to the above except it shows where
#sublimation values are reasonable
#In this case defined to be between 0.2 mm and 0.002 mm
parmsubset = parms[(sublimation<0.2 & sublimation > 0.002),]
```

```
parmsubset2 = parmsubset
parmsubset2$alpha = parmsubset2$alpha-min(parms$alpha)
parmsubset2$alpha = parmsubset2$alpha/max(parms$alpha-min(parms$alpha))
parmsubset2$a = parmsubset2$a-min(parms$a)
parmsubset2$a = parmsubset2$a/max(parms$a-min(parms$a))
parmsubset2$b = parmsubset2$b-min(parms$b)
parmsubset2$b = parmsubset2$b/max(parms$b-min(parms$b))
pairs(parmsubset2,panel=function(x,y,...){smoothScatter(x,y,add=T)},upper.panel = NULL,
title('Density of Reasonable Sublimation')


#This creates a similar density plot except for vertical transport
#In this case, between 80 kg/m and 150 kg/m
parmsubset = parms[(transport>80 & transport < 150),]
parmsubset2 = parmsubset
parmsubset2$alpha = parmsubset2$alpha-min(parms$alpha)
parmsubset2$alpha = parmsubset2$alpha/max(parms$alpha-min(parms$alpha))
parmsubset2$a = parmsubset2$a-min(parms$a)
parmsubset2$a = parmsubset2$a/max(parms$a-min(parms$a))
parmsubset2$b = parmsubset2$b-min(parms$b)
parmsubset2$b = parmsubset2$b/max(parms$b-min(parms$b))
pairs(parmsubset2,panel=function(x,y,...){smoothScatter(x,y,add=T)},upper.panel = NULL,
title('Density of Reasonable Transport')
```

```
#This plot combines low error, reasonable sublimation, and reasonable transport
parmsubset = parms[(transport>80 & transport < 150 & sublimation < 0.2 & sublimation >
parmsubset2 = parmsubset
parmsubset2$alpha = parmsubset2$alpha-min(parms$alpha)
parmsubset2$alpha = parmsubset2$alpha/max(parms$alpha-min(parms$alpha))
parmsubset2$a = parmsubset2$a-min(parms$a)
parmsubset2$a = parmsubset2$a/max(parms$a-min(parms$a))
parmsubset2$b = parmsubset2$b-min(parms$b)
parmsubset2$b = parmsubset2$b/max(parms$b-min(parms$b))
pairs(parmsubset2,panel=function(x,y,...){smoothScatter(x,y,add=T)},upper.panel = NULL,
title('Density of Reasonable Transport, Sublimation and Error')
```

```
#This is the other type of plot, not based on density and compares inputs to outputs
#The condition for this one is on the variable "Low_MSE" and how that gets subset
total_plot = parms
total_plot$sublimation = sublimation
total_plot$transport = transport
total_plot$MSE = Error
#pairs(total_plot[total_plot$MSE<1,],panel=function(x,y,...){smoothScatter(x,y,add=T)},
Low_MSE <- total_plot[total_plot$MSE<0.15,]
Low_MSE2 = Low_MSE
Low_MSE$alpha = Low_MSE$alpha - min(total_plot$alpha)
```

```
Low_MSE$alpha = Low_MSE$alpha/(max(total_plot$alpha)-min(total_plot$alpha))
Low_MSE$a = Low_MSE$a - min(total_plot$a)
Low_MSE$a = Low_MSE$a/(max(total_plot$a)-min(total_plot$a))
Low_MSE$b = Low_MSE$b - min(total_plot$b)
Low_MSE$b = Low_MSE$b/(max(total_plot$b)-min(total_plot$b))
Low_MSE$sublimation = Low_MSE$sublimation - min(Low_MSE2$sublimation)
Low_MSE$sublimation = Low_MSE$sublimation/(max(Low_MSE2$sublimation)-min(Low_MSE2$subli
Low_MSE$transport = Low_MSE$transport - min(Low_MSE2$transport)
Low_MSE$transport = Low_MSE$transport/(max(Low_MSE2$transport)-min(Low_MSE2$transport))
Low_MSE$MSE = Low_MSE$MSE - min(Low_MSE2$MSE)
Low_MSE$MSE = Low_MSE$MSE/(max(Low_MSE2$MSE) - min(Low_MSE2$MSE))
pairs(Low_MSE,upper.panel=NULL,pch=19,cex=0.1,xlim=c(0,1),ylim=c(0,1))
title('Sublimation < 1.15 mm')

#Top 10% is MSE<0.23
#Top 5% is MSE<0.15


#This shows the distributions of MSE, MAD, sublimation, and transport
par(mfrow=c(2,2))
plot(density(total_plot$MSE[total_plot$MSE<10]),lwd=2,xlab='MSE',main='')
abline(v=0.23,col='red')
abline(v=0.8,col='blue')
plot(density(total_plot$sublimation[total_plot$sublimation<100]),xlab='Sublimation (mm)
     main='')
```

```
abline(v=0.0073,col='blue')
plot(density(total_plot$transport[total_plot$transport<200 & total_plot$transport>0]),l
     ,xlab='Vertical Transport (kg/m)',main='')
abline(v=122,col='blue')
plot(density(MADev[MADev<10]),xlab='MAD',lwd=2,main='')
abline(v=1.06,col='blue')
```

A.4 Call Piektuk-D Script

```
#!/bin/bash
#Path to the file holding the list of parameter values to test
input="/path/to/parameterholder.dat"
i=0
#While loop to scan through the parameterholder file
while IFS= read -r line
do
    #This prints which iteration the script is on
    echo $i
    #This removes the parameters from the current parms.dat file
    #that the Fortran code reads
    sed -i '1d' parms.dat
    #This adds the new parameters to the parms.dat so that
    #the fortran code can read it
    echo "$line" >> parms.dat
    #Calls the Piektuk-D code
```

```
    #the "$i" part is to tell the code which iteration it
    #is on so it knows what to call the output files
    ./piektuk.out $i
    #increments i
    i=$((i+1))
done < "$input"
```