ASSESSING THE DEPENDENCE OF BULK ICE PROPERTIES FROM PROBES WITH ANTI-SHATTER TIPS ON ENVIRONMENTAL CONDITIONS

BY

ROBERT C. JACKSON

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Atmospheric Sciences
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Doctoral Committee:

Professor Greg McFarquhar, Chair
Professor Bob Rauber
Professor Sonia Lasher-Trapp
Dr. Jeff Stith, National Center for Atmospheric Research

ABSTRACT

Ice clouds have significant impacts on the Earth's radiative budget. Their radiative impact highly depends on ice cloud microphysical properties. Climate and weather prediction models have to make certain assumptions about how the various processes are represented. Observations of how cloud properties vary with environmental conditions can help evaluate some parameterizations used in models. However, sufficient data are not available to characterize how ice crystal properties vary as a function of environmental conditions. Furthermore, many of these assumptions are derived from historical datasets collected by in situ probes, namely optical array probes that can be contaminated by shattered artifacts generated by large particles shattering on the probe tips and inlets.

Therefore this study has two main objectives. Prior estimates of ice crystal size distributions derived from 2D Cloud Probes (2DCs) have been artificially amplified by small ice crystals generated from the shattering of large ice crystals on the probe tips. Although anti-shatter tips and algorithms exist, there is considerable uncertainty in their effectiveness. Therefore, this thesis first examines the differences in ice crystal size distributions, and bulk and optical properties from adjacent 2DCs with standard and anti-shatter tips, and processed with and without anti-shattering algorithms. The measurements were obtained from the National Research Council of Canada Convair-580 during the 2008 Indirect and Semi-Direct Aerosol Campaign (ISDAC) and the National Center for Atmospheric Research C-130 during the 2011 Instrumentation Development and Education in Airborne Science 2011 (IDEAS-2011). The 2DC size distributions are compared with those from the Holographic Detector for Clouds (HOLODEC), which has anti-shatter tips and allows for identification of shattering through spatial statistics.

The ratio of the number concentration *N* of particles with maximum dimensions 125 to 500 μm from the 2DC with standard tips to that from the 2DC with modified tips was correlated with median mass diameter and perimeter divided by area, but not with air speed, attack and attitude angles. Anti-shatter tips and algorithms reduced *N* by up to a factor of 10 for IDEAS-2011 and ISDAC, but neither alone removed all artifacts. For the period with coincident data, *N* from the HOLODEC and 2DC with modified tips are both around 5 x 10^{-3} L^{-1} μm^{-1}, suggesting that anti-shatter tips and algorithms combined remove artifacts from the 2DC for the conditions sampled during IDEAS-2011.

To assess the applicability of 2DC data obtained without anti-shatter tips previously used in parameterization schemes for numerical models and remote sensing retrievals, the impacts of artifacts on bulk microphysical and scattering properties were examined by quantifying differences between such properties derived from 2DCs with standard and anti-shatter tips, and with and without the use of shatter detection algorithms using the ISDAC and IDEAS-2011 data. Using either modified tips or algorithms changed quantities dominated by higher order moments such as ice water content, bulk extinction, effective radius, mass weighted terminal velocity, median mass diameter, asymmetry parameter and single scatter albedo at wavenumbers from 5 to 100 cm^{-1} and wavelengths of 0.5 to 5 μm by less than 20%. This is significantly less than the fractional changes in quantities dominated by lower order moments such as number concentration. The results suggest that model parameterizations and remote sensing techniques based on higher order moments of ice particle size distributions obtained in conditions similar to those sampled during IDEAS-2011 and ISDAC derived from 2DCs are not as biased by shattered remnants compared to those derived from lower order moments.

The second main objective of this thesis was to examine the dependence of ice cloud microphysical properties derived from the 2D Stereo and 2D Precipitation Probes on board the Stratton Park Engineering Company (SPEC) Learjet during the Small Ice Particles in Cirrus (SPARTICUS) experiment on temperature, and formation mechanism. An existing approach that represents a size distribution (SD) as a single gamma function using an ellipse of equally realizable solutions in (N_0, λ, μ) phase space was modified to automatically identify multiple modes in SDs and characterize each mode by such an ellipse. The dependencies of N_0, μ, λ from each mode, total number concentration, bulk extinction, ice water content, and median (mass) diameter as a function of T and formation mechanism were determined.

The changes in the observed codependencies between N_0, μ, and λ, as well as ice water content, extinction, and median (mass) diameter with environmental conditions indicate that particles grew to larger sizes at higher temperatures during SPARTICUS. No more than 2 modes were observed in SDs during SPARTICUS. The average boundary between the two modes was at 115 μm, similar to past studies not using probes with shatter mitigating tips and artifact removal algorithms. The bimodality of the SD was more frequent with increasing T, consistent with past studies that hypothesized that the bimodality was caused by nucleation occurring in the presence of sedimentation. Furthermore, the differences in the observed codependencies of N_0, μ, and λ between the two modes were consistent with past studies attributing these differences to particles in the mode with smaller maximum dimensions D growing primarily by deposition and the particles in the mode with larger D by both aggregation and vapor deposition.

ACKNOWLEDGEMENTS

This work was supported by the Office of Biological and Environmental Research (BER) of the U.S. Department of Energy (DOE) grants DE-SC00001279 and DE-SC0008500, as well as the National Science Foundation (NSF) grant number AS-1213311 and the NCAR Advanced Study Program. NCAR is supported by the NSF. Data were obtained from the Atmospheric Radiation Measurement Program archive, sponsored by the U.S. DOE Office of Science, BER Environmental Science Division. We would like to acknowledge operational, technical and scientific support provided by NCAR's Earth Observing Laboratory, sponsored by NSF. The authors also acknowledge Walter Strapp at Environment Canada his efforts in collecting data for this study and William Cooper for providing advice. Data were obtained from the Atmospheric Radiation Measurement Program archive, sponsored by the U.S. DOE Office of Science, BER Environmental Science Division. We are grateful for the comments from the reviwers of the two manuscripts that considerably improved the quality of Chapters 2 and 3.

TABLE OF CONTENTS

CHAPTER

1. INTRODUCTION…….1

1.1. BACKGROUND…..……..……..1

1.2. FIELD EXPERIMENTS – ISDAC, IDEAS-2011, AND SPARTICUS…..…….....7

1.3. DISSERTATION OUTLINE.……...…10

2. AN ASSESSMENT OF THE IMPACT OF ANTI-SHATTERING TIPS AND ARTIFACT REMOVAL TECHNIQUES ON CLOUD ICE SIZE DISTRIBUTIONS MEASURED BY THE 2D CLOUD PROBE…….…….21

2.1. INTRODUCTION……...21

2.2. METHODOLOGY……..…….26

2.3. INTERARRIVAL TIME ANALYSIS…….....31

2.4. UNCERTAINTY IN N(D) DUE TO SHATTERED ARTIFACTS……..……37

2.5. CONCLUSIONS……..44

3. AN ASSESSMENT OF THE IMPACT OF ANTI-SHATTERING TIPS AND ARTIFACT REMOVAL TECHNIQUES ON BULK CLOUD ICE MICROPHYSICAL AND OPTICAL PROPERTIES MEASURED BY THE 2D CLOUD PROBE……..…….69

TABLE OF CONTENTS (CONT.)

3.1. INTRODUCTION……...….69

3.2. METHODOLOGY ……..73

3.3. CONTRIBUTIONS OF SHATTERED PARTICLES TO BULK PARAMETERS....76

3.4 DISCUSSION AND CONCLUSIONS…….……82

4. THE DEPENDENCE OF ICE GAMMA SIZE DISTRIBUTIONS EXPRESSED AS VOLUMES IN N_0-λ-μ PHASE SPACE ON ENVIRONMENTAL CONDITIONS FOR CIRRUS OBSERVED DURING SMALL ICE PARTICLES IN CIRRUS (SPARTICUS)……99

4.1. INTRODUCTION…….....99

4.2. CLASSIFICATION OF METEOROLOGICAL CONDITIONS DURING SPARTICUS……103

4.3. PROCESSING OF SPARTICUS DATA……....105

4.4. SPARTICUS MICROPHYSICAL PROPERTIES AND IGF FITS……114

4.5. CONCLUSIONS……..125

4.6 ACKNOWLEDGMENTS…….128

5. CONCLUSIONS AND RECOMMENDATIONS……..148

5.1. SUMMARY……..148

5.2. PRINCIPAL CONCLUSIONS…….149

TABLE OF CONTENTS (CONT.)

5.3. FUTURE STUDIES……..152

REFERENCES……154

APPENDIX A: CODES FOR ANALYSIS OF PROBE DATA…….168

A.1. INTRODUCTION……168

A.2. PROCESSING METHODS……..169

A.3. TUTORIAL……...……176

A.4. VIEWING SIZE DISTRIBUTIONS AND CONCENTRATIONS……..183

A.5. VIEWING IMAGES……..184

A.6. MATLAB CODE……...188

CHAPTER 1

INTRODUCTION

1.1 Background

Ice clouds cover around 30% of the planet (Wylie et al. 2005), and have significant effects on the Earth's radiative budget. Ice clouds warm the Earth by absorbing outgoing longwave radiation and emitting radiation to the surface, and cool the Earth by reflecting sunlight. The solar heating rates within cirrus associated with convective clouds reach 1.5 K day^{-1}, and the infrared heating rates reach 20 to 30 K day^{-1} (Ramaswamy and Ramanathan 1987, Ackerman et al. 1998). Furthermore, Ramanathan and Collins (1991) hypothesized that, as sea surface temperatures increase, the coverage of cirrus anvils in the tropics that reflect solar radiation away from Earth increases and hence transport heat away from the Tropics. But, Lindzen et al. (2001) hypothesized that increased sea surface temperatures in the Tropics could result in decreased cirrus coverage, increasing the amount of infrared radiation emitted to space. Hartmann and Michelsen (2002) found no evidence for the hypothesis presented by Lindzen et al. (2001), while Lin et al. (2002) suggested that this process may provide a weak positive feedback. Therefore, processes that control the coverage of cirrus and its radiative impact are still poorly known.

Cirrus form in the presence of a variety of meteorological conditions, including low pressure systems and fronts, and outflow regions of warm conveyor belts, jet streams, and thunderstorm anvils. The ice particles can form by either homogenous nucleation or heterogeneous nucleation. Homogeneous nucleation occurs when a liquid particle freezes into ice without the influence of an aerosol. The number of ice crystals produced via homogeneous

nucleation is controlled by the number of supercooled droplets which increases with vertical velocity for a given temperature and aerosol concentration (Koop et al. 2000). Heterogeneous nucleation occurs when a droplet freezes with the aid of an aerosol called an ice nucleus (Pruppacher and Klett 1997). For cirrus with T < about -40°C, homogenous nucleation becomes possible. For T > about -40°C, heterogeneous nucleation is the only mechanism by which ice crystals can form. After nucleation, ice crystals can undergo growth by vapor diffusion, aggregation, or accretion of supercooled water on the crystal. They also can sublimate into water vapor. The relative importance of these processes depends on environmental conditions such as temperature, relative humidity, vertical velocity, and the concentration and composition of ice nuclei.

Because the spatial and temporal scales at which these processes occur are much shorter than can be used in large scale models due to computing time limitations, current general circulation models (GCMs) cannot explicitly represent nucleation and growth processes. Typically, variables such as number concentration N and ice water content IWC are prognosed in GCM parameterization schemes (Morrison et al. 2011), whereas other variables used such as bulk extinction β, mass weighted fall speed v_m, single scatter albedo ω, and asymmetry parameter g are derived from N or IWC or other diagnosed relations that depend on assumptions about the distributions of ice particle shapes and sizes. Assumptions in the parameterizations can have big impacts on model simulations. For example, assumptions about v_m control cloud coverage and lifetime (Jakob and Klein 1999; Sanderson et al. 2008). The properties of the particle size distribution are typically assumed to follow gamma distributions (i.e. Ulbrich 1983). Two of the gamma parameters can be derived from the prognosed N and IWC, whereas the third must be assumed or follow a diagnostic relation. Assumptions about how N and IWC

vary as a function of temperature have been developed (McFarquhar and Heymsfield (1997), McFarquhar and Heymsfield (2002), Heymsfield et al. 2003; 2013). But, dependence on temperature is not sufficient to explain all correlations. For example, Lawson et al. (2010) found that *IWC* in tropical anvil cirrus decreased with increasing distance from convection and with anvil age. GCMs are quite sensitive to the parameterization used to represent ice cloud processes and have been noted as a weak link in the use of GCMs (Stephens 2005). Therefore, to improve and evaluate the representation of processes in models and retrieval schemes, knowledge of how ice properties depend on environmental conditions is required.

Many parameterizations of ice clouds in GCMs, as well as many past observations of cirrus microphysical properties, were derived from ice crystal size distributions collected by probes mounted on aircraft (e.g, McFarquhar and Heymsfield 1997; and Heymsfield et al. 2003; 2013). More examples of such observations and parameterizations are listed in Chapter 3. Two main types of probes have been used to collect particle size distributions on aircraft. The first type of probe, the forward scattering probe derives the size of individual particles from the amount of light forward scattered by particles passing through a sample volume illuminated by the laser. A schematic diagram of a forward scattering probe is shown in Figure 1.1. A laser beam is directed into a mirror which subsequently reflects the beam into the sample volume. The scattered intensity between 4 to 12 degrees is then measured and converted to a particle size assuming Mie theory (Mie 1908). This class of probe typically measures particles with maximum dimension *D* from 2 to around 50 μm, with some versions of these probes measuring particles as large as 100 μm.

Another type of probe used to infer particle size distributions is the optical array probe. The optical array probe records the silhouette of a particle as it occults a laser beam shining onto

a photodiode array. Fast response electronics clock the photodiode array with sufficient frequency so that a slice is recorded each time the aircraft travels the width of the photodiode element. These slices are then combined together to make a silhouette of a particle such as that shown in Figure 1.2. This class of probes is nominally used to measure ice particles with D from ~50 μm to 2 cm. Particles with D < ~50 μm, where the size of the particle approaches the size of the photodiode array, are not well detected because the response time of the probe at high airspeeds is too long. Further, Baumgardner and Korolev (1997) showed the depth of field, and hence its sample volume, is poorly defined for D < ~150 μm. Table 1.1 shows a list of commonly used forward scattering and optical array probes along with the size range they measure. Different optical array probes have different photodiode widths that are tailored to measure specific sizes of particles. For example, differing versions of the 2D Precipitation Probe have a photodiode width varying from a 100 to 200 μm that is more suitable for measuring precipitation. On the other hand, differing versions of the 2D Cloud Probe (2DC) and Cloud Imaging Probe can have 15 to 50 μm resolutions suitable for measuring drizzle and cloud ice particles. Since D can vary by four orders of magnitude in cloud, multiple cloud probes are typically installed on an aircraft during an observational program.

There are many different factors that contribute to measurement uncertainties in the number distribution function $N(D)$ derived from forward scattering and optical array probes. Since ice particles are non-spherical, Mie theory is not applicable for deriving D from the forward scattered intensity for ice particles (Baumgardner et al. 1992). Furthermore, for optical array probes, the sample volume depends on the size of the particle for particles with D < 150 μm due to diffraction effects. The limited depth of field also leads to the imaging of out of focus particles (Baumgardner and Korolev 1997; Korolev 2007) for which sizing and concentration

corrections are uncertain. Furthermore, probe response times can affect the amount of particles $D < 150$ μm detected by optical array probes (Lawson 2006).

Another uncertainty is associated with the measurement of remnant, rather than real, ice ice crystals. Many studies have shown that the shattering of large ice crystals on the tips and inlets of forward scattering and optical array probes can cause significant overestimates in the number distribution function N(D), especially for $D < 500$ μm (i.e. Gardiner and Hallett 1985, Gayet et al. 1996, Field et al. 2003, 2006, Korolev and Isaac 2005, Heymsfield 2007, McFarquhar et al. 2007, 2011, Jensen et al. 2009, Zhao et al. 2011, Lawson 2011, Febvre et al. 2012, Korolev et al. 2011, 2013a, 2013b). Therefore, prior use of data from these probes uncorrected for these shattering events, including data that were used to construct many cirrus parameterizations in GCMs, could provide misleading results. It is not well known the degree to which these shattering events affect the values of different moments derived from the size distribution. Therefore, one goal of this thesis is to establish the degree to which ice particle shattering impacts bulk ice and optical properties used to derive parameterizations.

Two main different methods exist to mitigate shattered artifacts in data collected by forward scattering and optical array probes:

1. the elimination of artifacts in the collected data, and
2. the use of probe tips modified to deflect the artifacts away from the sample volume

Field et al. (2003) developed an algorithm to remove artifacts by eliminating all particles when interarrival times Δt within the sample volume were less than some threshold. This was done by fitting a curve to the normalized frequency distribution of Δt for all particles in the sample volume and choosing a threshold Δt for classifying particles as shattered artifacts, as shown in

Figure 1.3. Table 1.1 lists the probes for which this technique can be used to remove shattered artifacts. Two peaks are present in Figure 1.3. The peak corresponding to shattered artifacts has a shorter mode because particles that shatter into pieces arrive closely together in the sample volume. Natural ice particles observed in the cloud arrive typically further apart, so that real particles are contained in the peak with a longer interarrival time. There is some uncertainty in determining the threshold for separating real particles and artifacts especially given that the peaks corresponding to real and shattered particles can overlap, as shown by the blue line in Figure 1.3. The threshold can also vary during the flight because the concentration throughout the flight varies tremendously. Furthermore, diffraction fringes can produce short interarrival times when they are identified as separate particles which can lead to the erroneous rejection of real particles (Korolev and Field 2014). Korolev and Isaac (2006) also removed shattered artifacts from optical array probe data by removing particles based on particle morphology.

Another method to eliminate shattered artifacts is the modification of the probe inlets and tips to sweep shattered artifacts away from the sample volume. For forward scattering probes, the inlet tube in in Figure 1.4c can be replaced by an open path design to reduce the number of surfaces upon which ice crystals can shatter. This has been done for the Cloud Droplet Probe, as shown in the top panel in Figure 1.4a, where the shroud of the Forward Scattering Spectrometer Probe (FSSP) has been removed for some field projects as shown in Figure 1.4b. Figure 1.5 shows how the tips of optical array probes have been modified to reduce the number of shattered artifacts entering the sample volume. For the unmodified 2DC probe in Figure 1.5a, ice crystals can shatter on the 2DC hemispheric tips and enter the sample volume. However for the modified 2DC in Figure 1.5b, the tips are cut at an angle so that the shattered artifacts are not as likely to enter the sample volume. The probe housing surrounding the laser has also been removed,

further reducing the number of surfaces for possible shattering. Studies differ on whether or not algorithms remove more shattered artifacts than modified tips (Lawson 2011; Korolev 2011; 2013b). Therefore, it is important to determine the efficacy of algorithms and modified tips for producing more accurate ice sampling over a wide set of probes, configurations, and cloud sampling conditions. Further, comparison of modified and standard probe performance in a variety of conditions can provide guidance on the bias produced by shattering in historical cirrus datasets.

1.2. Field experiments – ISDAC, IDEAS-2011, and SPARTICUS

In this thesis, data from three different field experiments where aircraft sampled ice clouds are used to address how cirrus microphysical properties vary as a function of temperature, and formation mechanism. First, the uncertainity in 2DC ice measurements due to shattering is quantified, and the optimum technique to reduce the shattering bias in microphysical properties derived from airborne measurement probes is determined. In order to accomplish this objective, data from two different field experiments sampling ice clouds are used. In the first experiment, the Indirect and Semi-Direct Aerosol Campaign (ISDAC), the National Research Council of Canada Convair-580 with over 40 different instruments conducted 27 sorties over the North Slope of Alaska with the primary focus on collecting a dataset of arctic mixed phase cloud and aerosol properties to test hypotheses regarding ice and mixed-phase cloud-aerosol interactions (McFarquhar et al. 2011). For one sortie on 30 April 2008 a 2DC with standard tips and a 2DC with modified tips were installed adjacent to one another and were used to sample ice clouds in the vicinity of Fairbanks, Alaska as shown in Figure 1.6. These data are used to estimate the impact of shattered artifacts on ice particle size distributions and bulk cloud collected from 2DCs.

The second dataset used is from the Instrumentation Development and Education in Airborne Science 2011 project (IDEAS-2011) conducted by the National Center for Atmospheric Research (NCAR). IDEAS-2011 was designed to test new instrumentation as well to educate students on the operation and use of airborne instrumentation. In this project a 2DC with standard tips was mounted on the same pod of the National Science Foundation (NSF)/NCAR C-130 as a 2DC with modified tips, as shown in Figure 1.7. A Holographic Detector for Clouds (HOLODEC) was also installed on the C-130. The HOLODEC derives ice crystal size distributions from reconstructed holograms (Fugal and Shaw 2009; Fugal et al. 2003) instead of using the occultation of photodiodes as employed by optical array probes. Therefore, as the HOLODEC uses a different technique to measure N(D), it provides an independent measure of N(D) that can be used to assess the efficacy of shattered artifact removal algorithms and modified tips. Although there were 9 flights during IDEAS-2011, only two are used in this study. For two winter storms sampled on 25 October and 1 November 2011 the NSF-NCAR C-130, measuring ice clouds with a wide range of habits and particle sizes, flew a spiral ascent followed by a spiral descent in the vicinity of Cheyenne, Wyoming at temperatures from -40 to 0℃. The ascending spiral was flown at a slower speed than the descending spiral to increase the dynamic range of true airspeeds covered, so that the impact of air speed on shattering could also be examined.

Using the IDEAS-2011 and ISDAC datasets, the uncertainty in bulk microphysical properties used in parameterization schemes derived from past 2DC data sets not corrected for shattering is quantified. In particular, N, β, IWC, median diameter D_m, median mass diameter D_{mm}, effective radius, ω, and asymmetry parameter g are derived from standard and modified 2DC data with and without the application of shattering artifact removal algorithms. The four

versions of the generated bulk properties are compared against each other in order to estimate the contribution of shattered artifacts to N, β, IWC, median diameter D_m, median mass diameter D_{mm}, effective radius, ω, and g.

The final part of this study applies the understanding gained from the first part of the study to examine how mid-latitude cirrus microphysical properties vary as a function of environmental conditions, using data collected during the Small Ice Particles in Cirrus (SPARTICUS) project. During SPARTICUS, the Stratton Park Engineering Company (SPEC) Learjet flew 101 missions through mid-latitude cirrus in the US Central Plains from January to June 2010. Many of the flights were focused in the area around the Southern Great Plains site in Lamont, OK where spiral flight tracks were conducted in the vicinity of ground-based radars and lidars. Synoptically, convectively, and orographically generated cirrus at temperatures of -70 to 0℃ were sampled during SPARTICUS. This ice cloud dataset is unprecedented in terms of the number of hours of data that were collected, and the time period over which the data were collected using shatter-reducing probe tips to sample ice clouds. Most previous experiments have conducted fewer flights over a shorter, typically 1-month period. A much wider range of meteorological conditions were sampled during the 6 months of SPARTICUS than during a typical single field project sampling midlatitude cirrus. Furthermore, many past cirrus campaigns such as the Tropical Warm Pool –Ice Crystal Experiment (May et al. 2008) and the Tropical Composition, Cloud, and Climate Coupling project used data from probes that are likely more influenced by shattered artifacts (McFarquhar et al. 2007; Jensen et al. 2010). Thus, the influence of shattered artifacts is expected to be less for the observations collected during SPARTICUS than other campaigns samping cirrus. The probes used for SPARTICUS were the 2D Stereo Probe, 2DP and "Fast" FSSP shown in Figure 1.8. Shattered artifact removal algorithms can be

applied to data from these probes, and the 2DS was equipped with shatter reducing tips. For all 3 projects, 2.3 μm resolution images of ice particles were taken by a Cloud Particle Imager (CPI) and are used to provide information about the distributions of particle habits. Although shattered artifacts may also impact CPI data, the CPI is only used in this study to identify particle habit rather than to derive particle number concentration.

1.3 Dissertation outline

The rest of this dissertation is organized as follows. Chapter 2 of this dissertation uses the ISDAC and IDEAS-2011 data to answer the following questions:

1) How does the ratio of particle concentrations between standard and modified 2DCs in different size ranges vary as a function of particle size, shape, number concentration, temperature, true air speed, aircraft attitude angles, and angle of attack?

2) How does the ratio of particle concentrations in different size ranges from a 2DC change when using artifact removal algorithms as a function of particle size, shape, concentration, habit, temperature, true air speed, aircraft attitude angles, and angle of attack?

Chapter 2 also shows the algorithm that was developed to remove shattered artifacts from 2DC data and has minor modifications from a paper that is in press for the *Journal of Atmospheric and Oceanic Technology*.

Chapter 3 computes potential biases in the bulk microphysical and single scattering properties β, *IWC*, ω, g, v_m derived from 2DCs due to shattering by intercomparing β, *IWC*, ω, g. v_m from a 2DC with standard tips to those same parameters derived from a 2DC with modified tips, and to those derived from data corrected with shattering artifact removal algorithms. The

goal is to provide estimates of potential biases in microphysical parameters that are used in parameterization schemes and remote sensing schemes derived from data uncorrected for shattering. Chapter 3 is adapted from a paper published in the *Journal of Atmospheric and Oceanic Technology*.

Chapter 4 of this dissertation uses the techniques developed in Chapter 2 to determine how N, β, IWC, and curve fits to $N(D)$ observed in the mid-latitude cirrus during SPARTICUS vary as a function of environmental conditions. This material will ultimately be included in a paper to be submitted to the *Journal of Geophysical Research*.

Chapter 5 discusses the principal conclusions and provides recommendations for future studies determining the relationship between cirrus microphysical properties and environmental parameters. The appendix of this dissertation contains the code used to process the optical array probe data as well as information on how to use this code.

CHAPTER 1 FIGURES AND TABLES

Table 1.1. List of commonly used instrumentation for measuring ice particle size distributions

Name	**Type**	**Size range**	**Can apply Field et al. (2003) shattering correction?**
Forward scattering spectrometer probe (FSSP)	Forward scattering probe	2 – 100 μm	Only to "Fast" FSSP.
Cloud and Aerosol Spectometer (CAS)	Forward scattering probe	2 – 50 μm	No
Cloud Droplet Probe (CDP)	Forward scattering probe	2 – 50 μm	"Fast" CDP - open path design
2D Cloud Probe (2DC)	Optical array probe	Depends on model	Yes
2D Stereo Probe (2DS)	Optical array probe	Depends on model	Yes
Cloud Imaging Probe (CIP)	Optical array probe	Depends on model	Yes
2D Precipitation Probe (2DP)	Optical array probe	Depends on model	Yes
Precipitation Imaging Probe (PIP)	Optical array probe	Depends on model	Yes
High Volume Precipitation Sampler (HVPS)	Optical array probe	Depends on model	Yes

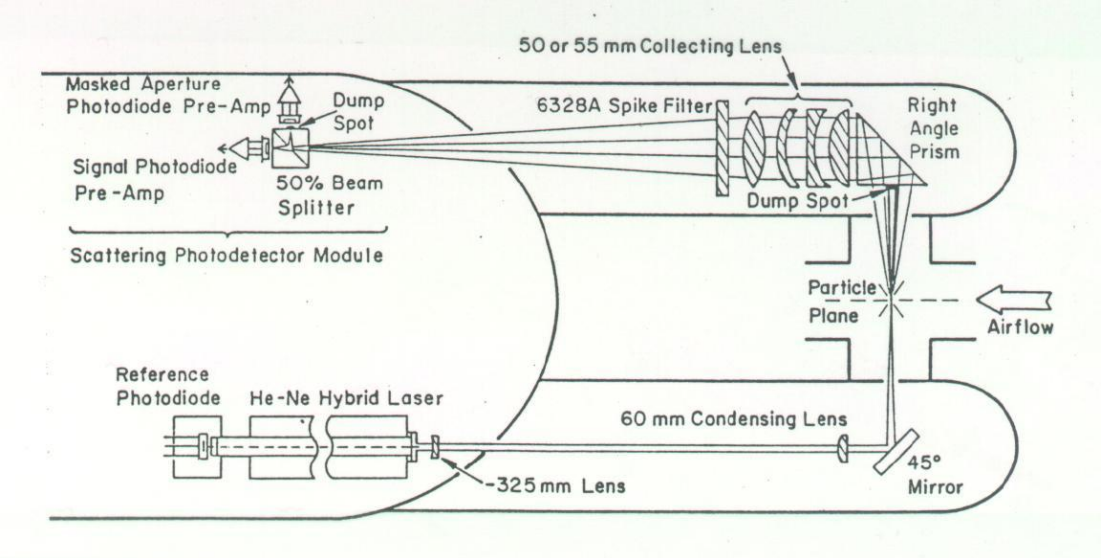


Figure 1.1. Schematic diagram of a forward scattering spectrometer probe. [Source: https://www.eol.ucar.edu/raf/Bulletins/B24/fssp100.html].

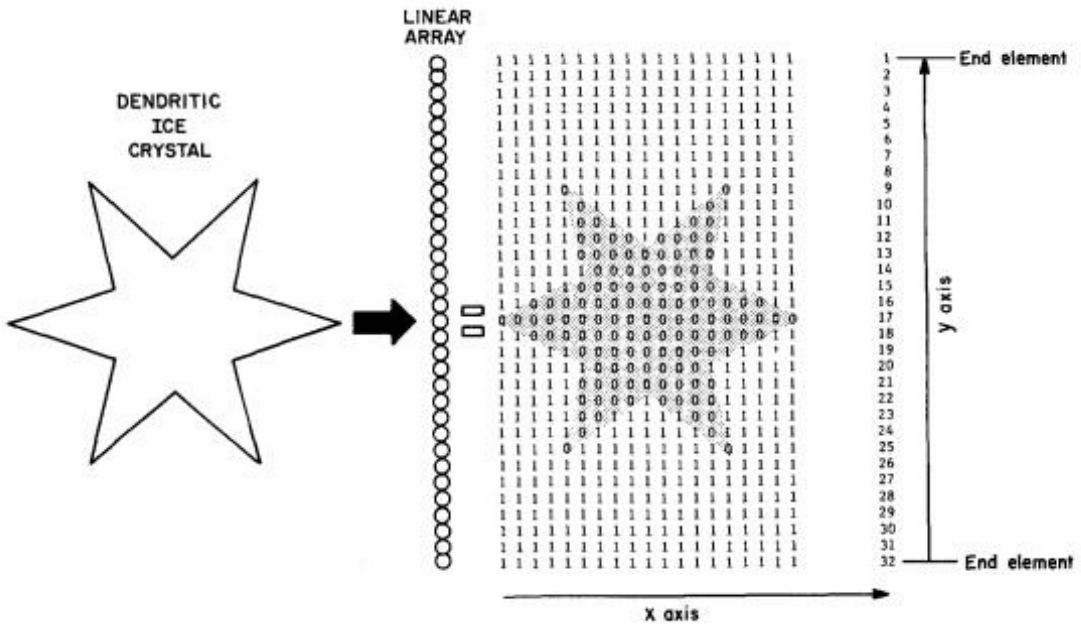


Figure 1.2. Heymsfield and Parish (1978)'s conceptual diagram of a particle passing over the photodiode array of an optical array probe. The resulting image recorded by the probe is shown on the right.

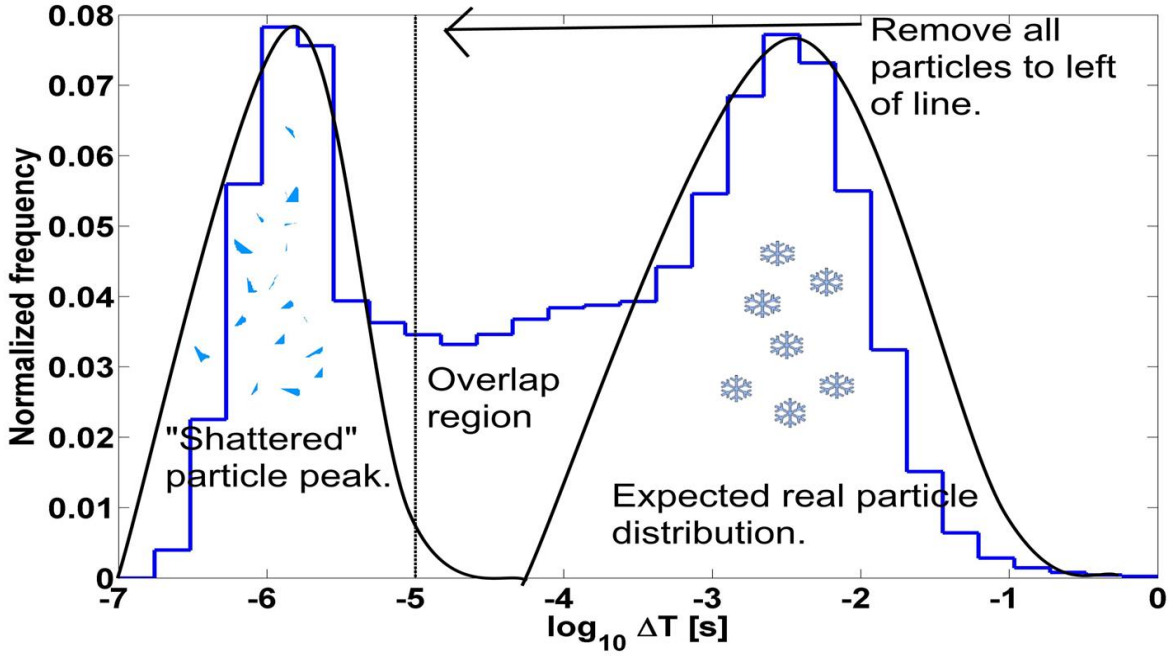


Figure 1.3. Conceptual diagram of the algorithm used to remove shattered artifacts from forward scattering and optical array probe data. Blue line shows the normalized frequency distribution of log Δt for a field experiment. The black curves show a fit of a bimodal distribution to the data. In this example, all particles with $\Delta t < 10^{-5}$ s are considered shattered artifacts.

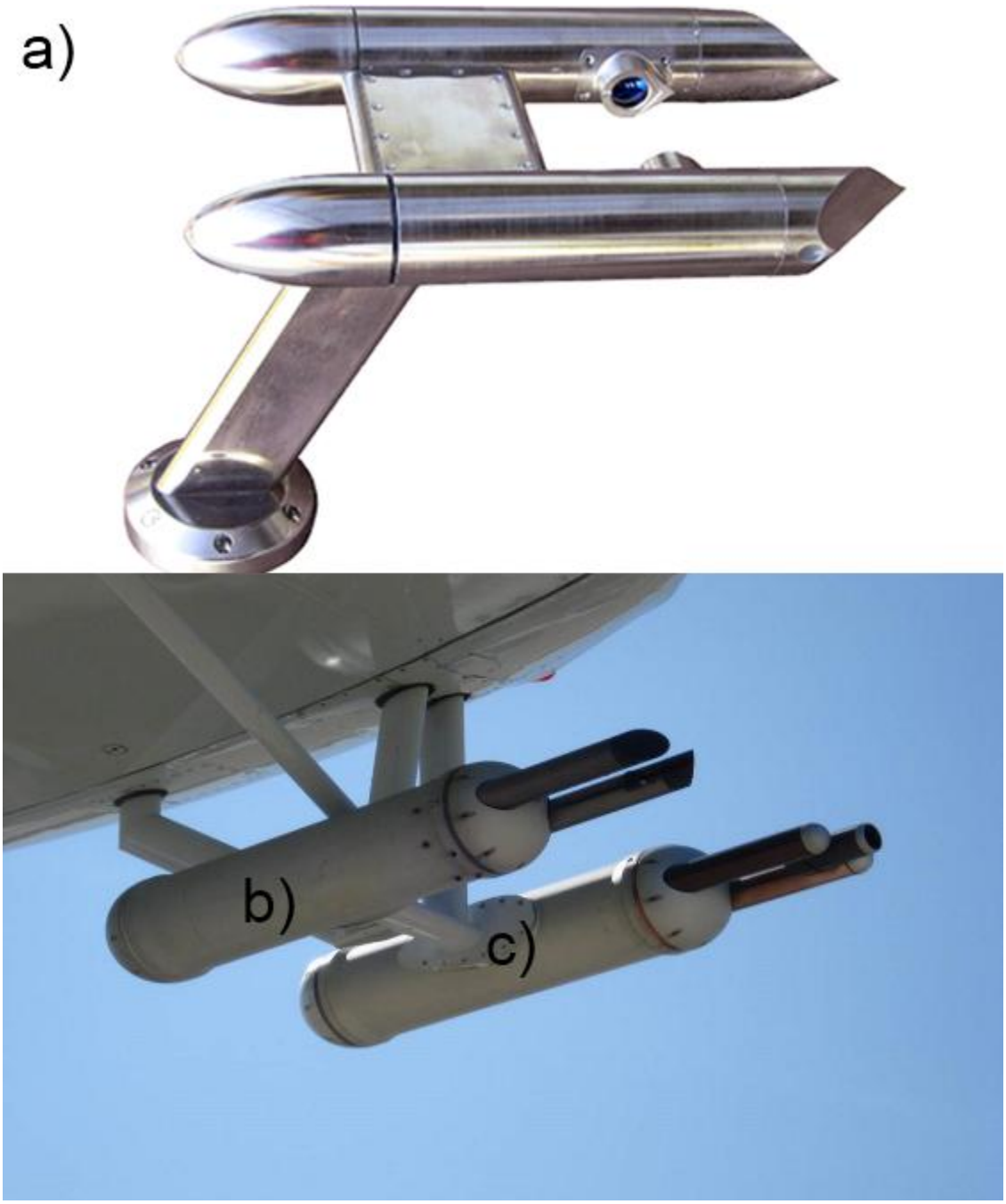


Figure 1.4. Photograph of a CDP **(a**), an unshrouded FSSP **(b),** and a shrouded FSSP **(c).** Top photograph taken by Droplet Measurement Technologies, Inc. [http://www.dropletmeasurement.com/]. Bottom photograph from https://www.eol.ucar.edu/raf/Bulletins/B24/fssp3.jpg.

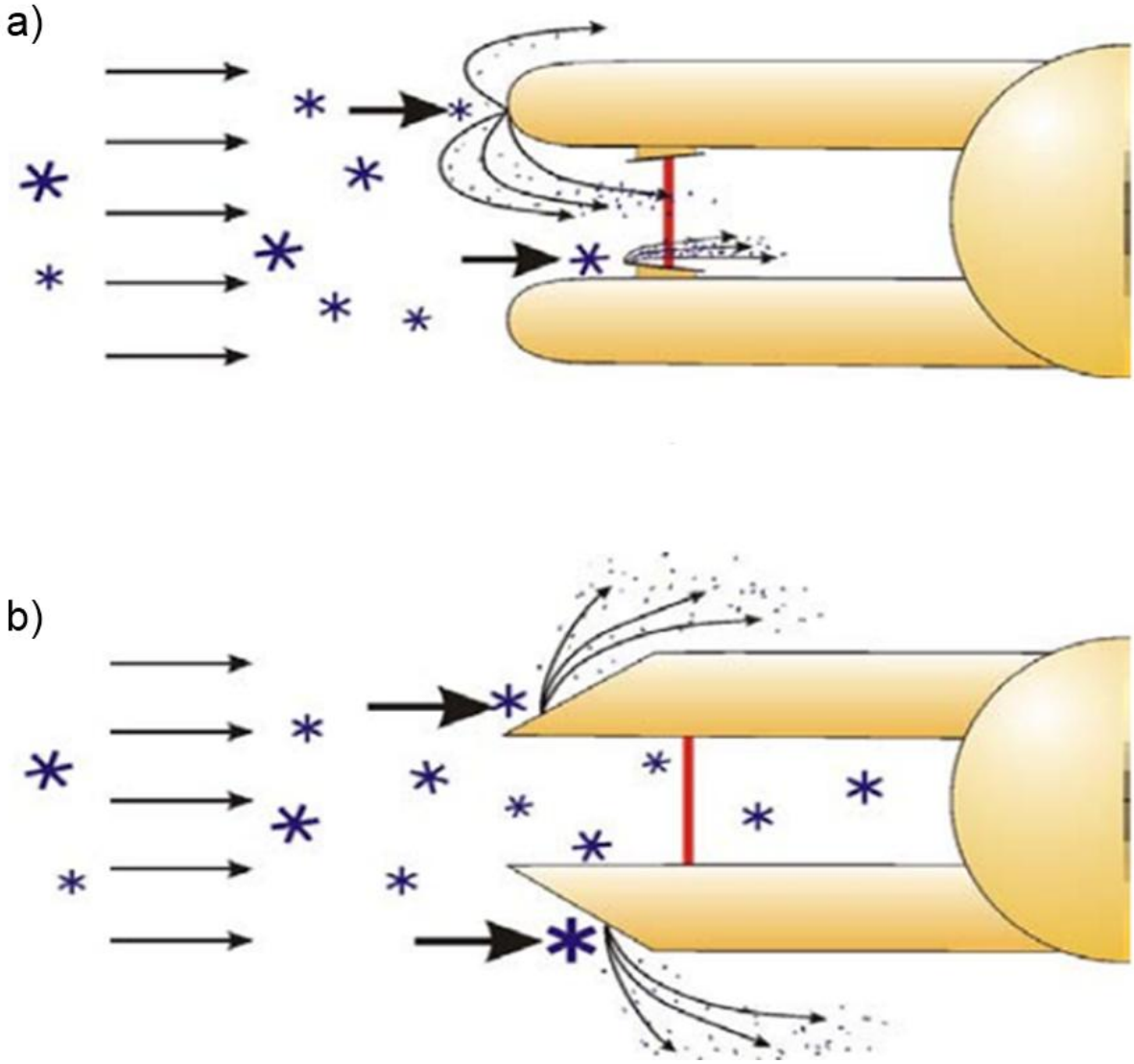


Figure 1.5. Conceptual diagram of a 2DC with standard tips **(a)** and with modified tips **(b)** sampling ice clouds, with the direction of aircraft flight to the left. Fewer shattered artifacts enter the sample volume in **(b)** than do in **(a).** Figure from Korolev and Isaac (2005)**.**



Figure 1.6. 2DC with standard tips **(left)** mounted next to the 2DC with shatter-mitigating tips **(right)** on the NRC-Convair-580 during ISDAC**.**

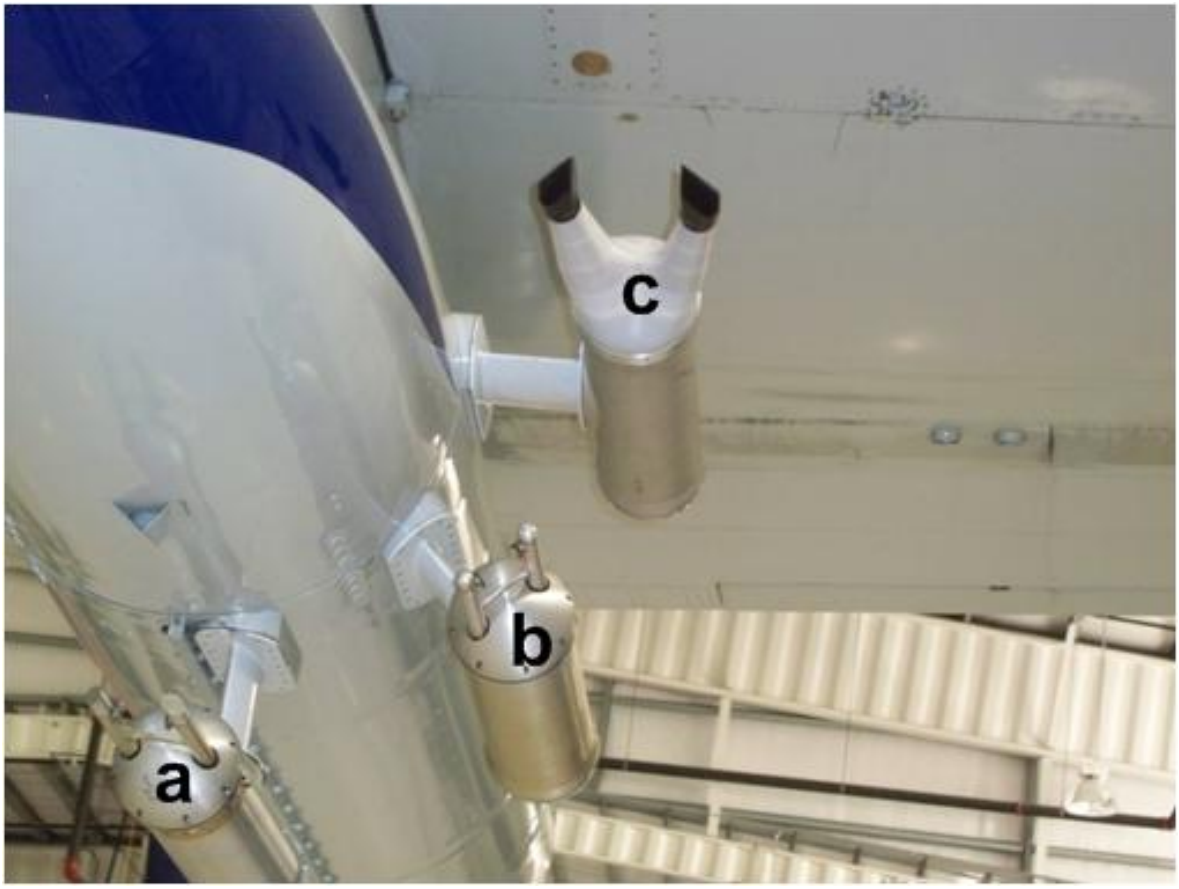


Figure 1.7. 2DC with shatter-mitigating tips **(a)** mounted next to the 2DC with standardtips **(b)** and the Holographic Detector for Clouds **(c)** on the NSF/NCAR C-130 during IDEAS-2011.



Figure 1.8. The 2DP, 2DS, Cloud Particle Imager (CPI) and Fast FSSP mounted on the SPEC Learjet during SPARTICUS. [image source: http://asr.science.energy.gov/meetings/stm/posters/view?id=22]

CHAPTER 2

AN ASSESSMENT OF THE IMPACT OF ANTI-SHATTERING TIPS AND ARTIFACT REMOVAL TECHNIQUES ON CLOUD ICE SIZE DISTRIBUTIONS MEASURED BY THE 2D CLOUD PROBE

Copyright Notice

This chapter is a paper in the *Journal of Atmospheric and Oceanic Technology* which is a journal of the American Meteorological Society (Jackson et al. 2014). The differences in this chapter from Jackson et al. (2014) only include changes to the formatting of the tables and figures. Jackson et al. (2014) is © Copyright 2014 American Meteorological Society (AMS). Permission to use figures, tables, and brief excerpts from this work in scientific and educational works is hereby granted provided that the source is acknowledged. Any use of material in this work that is determined to be "fair use" under Section 107 of the U.S. Copyright Act September 2010 Page 2 or that satisfies the conditions specified in Section 108 of the U.S. Copyright Act (17 USC §108, as revised by P.L. 94-553) does not require the AMS's permission. Republication, systematic reproduction, posting in electronic form, such as on a web site or in a searchable database, or other uses of this material, except as exempted by the above statement, requires written permission or a license from the AMS. Additional details are provided in the AMS Copyright Policy, available on the AMS Web site located at (http://www.ametsoc.org/) or from the AMS at 617-227-2425 or copyrights@ametsoc.org.

2.1. Introduction

Ice clouds cover around 30% of the planet (Wylie et al. 2005), contribute significantly to the diabatic heating of the upper troposphere (Ramaswamy and Ramanathan 1987) and may play

a role in regulating sea surface temperatures in the Tropics (Ramanathan and Collins 1991). They emit infrared radiation at lower temperatures than the surface, providing a longwave warming effect (Hartmann et al. 1992; Chen et al 2000; Stephens 2005; Boudala et al. 2007) that competes with the shortwave cooling effect, the balance of which depends highly on ice particle properties such as size and shape (e.g. Heymsfield and Milosevich 1991; Stephens et al. 1990; Stackhouse and Stephens 1991; Zender and Kiehl 1994; Zhang et al. 1999; McFarquhar et al. 2003; Bouldala et al. 2007). The ice particle mass-weighted terminal velocity, which also depends on ice crystal properties, controls the gravitational settling and abundance of cirrus (Jacob and Klein 1999) and is an important control of cloud fraction and ice water path simulated by general circulation models (GCMs) (Mitchell et al. 2008; Sanderson et al. 2008).

Mixed-phase clouds in all geographic regions also play important roles in global climate feedbacks and shaping the Earth's radiative budget. Midlevel mixed-phase clouds cover around 22% of the planet (Warren et al. 1986) and occur at all latitudes (Yoshida et al. 2010). In the Arctic, mixed-phase clouds have a critical role in a complex feedback involving the sea ice, clouds, aerosols, and the atmosphere (Curry et al. 1993; 1995), where rapid changes in temperature and sea ice coverage attributable to climate change are occurring (IPCC 2014). The persistence of arctic mixed-phase clouds is driven by a balance between cloud-top radiative cooling, ice sedimentation rates, latent heating, and synoptic scale and surface forcing (Harrington et al. 1999; Harrington and Olsson 2001; Avramov and Harrington 2010; Morrison et al. 2011; 2012; Solomon et al. 2011), the details of which depend on the ice particle properties. Therefore, accurate knowledge of ice and mixed-phase cloud properties are needed for process oriented understanding as well as development of parameterization schemes for models.

The uncertainty in ice crystal concentrations translates into large uncertainties in sedimentation and single scattering properties which affects model evolution. For example, Mitchell et al. (2008) showed that increasing the fraction of ice crystals with $D < 150$ μm compared to the total number concentration induced an upper tropospheric warming of 3 K and a total cloud forcing of -5 W m^2 in the Tropics through an increase of cirrus cloud coverage in a global model simulation relative to a control case. On the other hand, Boudala et al.'s (2007) simulation including contributions of crystals with $D < 150$ μm had a net radiative forcing of 2.4 W m^{-2} greater than that of a control case where such contributions were excluded. McFarquhar et al. (2003) showed that decreasing the ice crystal effective radius in a parameterization scheme affected the vertical profile of radiative forcing which in turn impacted low cloud cover, and hence longwave radiative forcing. Knowledge of ice crystal concentrations is not only needed for model parameterizations, but also for evaluation of nucleation schemes for mixed-phase clouds (Fridlind et al. 2007; Yang et al. 2013a), and remote sensing retrievals (e.g. Yang et al. 2013b).

In situ observations of the number distribution function *N(D)* are conventionally derived from two classes of cloud probes covering four orders of magnitude in particle size: forward scattering probes such as the Forward Scattering Spectrometer Probe (FSSP), Cloud Droplet Probe (CDP) and Cloud and Aerosol Spectrometer (CAS); and optical array probes such as the Two Dimensional Cloud Probe (2DC), Cloud Imaging Probe (CIP), Two Dimensional Stereo Probe (2DS), Precipitation Imaging Probe (PIP), Two-Dimensional Precipitation Probe (2DP), and High Volume Precipitation Sampler (HVPS). Korolev et al. (2011) photographed ice crystals bouncing and shattering on the 2DC's hemispheric tips and on a FSSP's inlet, showing that shattered artifacts could be swept into the probe sample volume contributing to *N(D)* for $D <$ 500 μm. Several other studies hypothesized that the *N(D)* measured by forward scattering and

optical array probes is contaminated by the presence of such small crystals generated by the shattering of larger crystals (Cooper 1978; Gardiner and Hallett 1985; Gayet et al. 1996; Field et al. 2003; 2006; Korolev and Isaac 2005; Heymsfield 2007; McFarquhar et al. 2007b; 2011; Jensen et al. 2009; Zhao et al. 2011; Lawson 2011; Febvre et al. 2012; Korolev et al. 2013a; 2013b).

The impact of ice crystal shattering on *N(D)* from 2DCs is investigated in this paper. Following Korolev et al. (2011, 2013b), two strategies are used to reduce the contributions of shattered artifacts to calculated ice crystal concentrations:

1) Algorithms based on the time between which particles are detected in the sample volume (Field et al. 2003; 2006) and based on the numbers, sizes and gaps between fragments in a single image recorded by the probe (Korolev and Isaac 2005) are used to remove shattered artifacts;
2) 2DC tips modified to deflect most artifacts generated by the shattering of large ice crystals away from the sample volume (Korolev and Isaac 2005; Korolev et al. 2011; Lawson 2011; Korolev et al. 2013a; 2013b) are used.

Previous studies differ on whether the use of algorithms or modified tips is more effective at removing shattered artifacts. Korolev et al. (2011) concluded that the modified tips removed more shattered artifacts than algorithms for the 2DC and CIP, but Lawson (2011) concluded that algorithms were more effective for the 2DS. Typically, a combination of the two approaches removes the most shattered artifacts (McFarquhar et al. 2011). The number of shattered fragments have been hypothesized to depend on a variety of factors such as particle size, shape, concentration, degree of riming, temperature *T*, true air speed *TAS*, aircraft attitude angle, and angle of attack (Vidaurre and Hallett 2009; Korolev et al. 2011; Korolev et al. 2013a,b). The

dependence of the shattering on concentration and size were characterized in Korolev et al. (2013b). In this study, data from the Indirect and Semi-Direct Aerosol Campaign (ISDAC) and the Instrumentation and Development in Airborne Science 2011 (IDEAS-2011) project in which 2DCs with both standard and modified tips were installed on the same aircraft are used to answer the following questions:

1) How does the ratio of particle concentrations between standard and modified 2DCs in different size ranges vary as a function of particle size, shape, concentration, temperature, true air speed, aircraft attitude angles, and angle of attack?

2) How does the ratio of particle concentrations in different size ranges from a 2DC using artifact removal algorithms and not using artifact removal algorithms vary as a function of particle size, shape, concentration, habit, temperature, true air speed, aircraft attitude angles, and angle of attack?

This extends previous studies in that data from 2 additional projects using particle probes with different size resolutions and time responses are used to determine the degree to which conventional probes are contaminated by particle shattering, and hence assess the degree to which prior data may be correctable. Further, a simplified model is developed to explain how the number of shattered artifacts depends on cloud and environmental parameters.

During ISDAC, the National Research Council (NRC) of Canada Convair-580 flew in the vicinity of Barrow and Fairbanks, Alaska in April 2008. The NRC Convair-580 flight on 30 April 2008 had two 2DCs installed, one with standard and one with modified tips. During IDEAS-2011, the National Science Foundation (NSF)/National Center for Atmospheric Research (NCAR) C-130 flew over Colorado and Wyoming during October and November 2011. Two

flights, 25 October and 1 November 2011 in deep precipitating systems generating snowfall over southeast Wyoming, are used because they provided the most data (~5 hours) from a standard 2DC and a 2DC with modified tips. Both data sets contain data in ice and mixed-phase clouds, with sampling of a wide a range of particle sizes and habits, for a variety of true air speed from 80 to 150 m s^{-1}, angles of attack from 0° to 7°, roll angles ~0°, and pitch angles from 0° to 8° .

The remainder of the chapter is organized as follows. Section 2.2 highlights the instrumentation used, sampling strategies, and data collection and processing methods. Section 2.3 describes the algorithm used to identify shattered artifacts. The effects of the modified tips and shattering removal algorithms on the measurements in different size ranges are discussed in Section 2.4. The principal findings of the study are summarized in Section 2.5.

2.2. Methodology

2.2.1. Instrumentation

Table 2.1 lists the instruments installed on the two aircraft from which data were used in this study. On each aircraft, the 2DC probes with standard and modified tips were mounted adjacent to each other, as shown in Figure 2.1, to minimize the impact of cloud inhomogeneties over larger spatial scales on probe intercomparisons. On the C-130, the 2DC with modified tips was mounted at a 45 degree angle relative to the 2DC with standard tips. This difference should not have a big impact on the measurement of *N(D)* because only particles with large aspect ratios, like columns, should fall with preferential orientation (McFarquhar et al. 1999). The 2DCs on the C-130 were mounted at different locations on a pod whereas the 2DCs on the NRC Convair-580 were installed on pylons under a wing. Because the different locations of the 2DCs on a pod during IDEAS-2011 could have affected the nature of the shattering, it is important that

data from both projects be used to look at common trends. The 2DC data were processed using the algorithms described in Section 2.2.2 that were originally developed at NCAR and subsequently modified at the University of Illinois. Due to the poorly defined depth of field and errors related to digitizing of binary images of 2DCs for small ice crystals (Korolev et al. 1998; Baumgardner and Korolev 1997), particles with $D < 125$ μm are not considered in this study.

A unique comparison of 2DC size distributions against those measured by the HOLOgraphic Detector in Clouds (HOLODEC) instrument is conducted here. The HOLODEC captures a 3D snapshot of up to 20 cm^3 volumes of cloud air at a rate of 3.3 Hz. Recorded holograms are numerically reconstructed, yielding the size, shape and 3D location of cloud particles with 6 μm < D < 1 mm in the sample volume. More details on the holographic technique and HOLODEC are available elsewhere (Fugal et al. 2004; Fugal and Shaw 2009; Fugal et al. 2009; Spuler and Fugal 2011). The 3D nature of the HOLODEC data allows for identification of shattering by analyzing spatial statistics of particle sizes and locations. Particles generated by the shattering of large crystals on probe tips appear in holograms as clusters of particles with smaller variability in inter-particle separation distances than expected from Poisson statistics, and are located primarily within the first few centimeters from the edges of the sample volume. This method can be considered a spatial analog of anti-shattering algorithms that remove ice contamination through interarrival time statistics, such as that employed for the 2DC data described in Section 2.3 of this paper. By analyzing the 3D particle positions in each hologram, individual holograms which contain shattering events can be identified and excluded from the analysis.

Other state and aircraft parameters placed the 2DC in the appropriate context. A pitot-static system measured airspeed, pitch, roll and attack angles while a Rosemount sensor

measured angle of attack. The true airspeed and angle of attack at the 2DCs on the C-130 could been altered by the nearby wings and pod. However, since the 2DC images were not elongated in the time direction, the probe was likely recording the images at the correct clock speed. Therefore, it can be assumed that particles arrive at the probe at approximately the true airspeed.

2.2.2. 2DC Data Processing technique

The software used to process the 2DC data includes corrections to the sample volume for the size dependence of the depth of field (Baumgardner and Korolev 1997). The interarrival time Δt for each particle was the time between the arrival of the current and previously detected particle. The maximum dimension of a particle D was defined by the maximum length in any direction, not just parallel or perpendicular to the diode array. If a particle was touching the edge of the photodiode array, D and the sample area SA were computed using the image reconstruction technique of Heymsfield and Parish (1978). SA varies from 102 mm^2 for D = 125 μm to 234 mm^2 for D = 3.2 mm for the 64 photodiode 2DC.

The University of Illinois software sorts each particle into nine habit categories following Holroyd (1987). In order to correct for the oversizing of hollow spherical particles due to diffraction, the correction algorithm of Korolev et al. (2007) adjusts D for particles classified as spheres. Diffraction effects are also possible for nonspherical particles, but there are no existing algorithms to correct for such effects.

Each particle is accepted or rejected based on a series of criteria designed to remove spurious stuck bits, splash artifacts, blank records, and streaker particles. With A_p representing the projected area of the particle, L the length of the particle in the direction perpendicular to the diode array, and W the length of the particle in the direction parallel to the diode array, a particle

is rejected if (i) $L/W > 5$ (6 if particle touches edge) to remove streakers and stuck bits with large aspect ratios, (ii) if $A_p/((\pi/4)D^2) < 0.2$ to remove particles such as stuck bits and streakers with a low number of shaded photodiodes relative to a shaded circle with diameter D.

Korolev and Isaac (2005) concluded that the presence of three or more separated images in a single identified particle were likely shattering events because the 2DC only detects multiple particles as a single particle when there are at most 2 slices between particles. This corresponds to a Δt of at most 6.2×10^{-7} s at an airspeed of 100 m s^{-1}. Single particles with three or more fragments typically have a low percentage of shaded area A/LW. Images with $A/LW < 0.25$ were thus classified as shattered particles following Lawson (2011). This criterion classified 3.7% of all particles imaged by the 2DC with standard tips and 2.1% of the particles from the 2DC with modified tips as shattered artifacts during ISDAC and IDEAS-2011.

2.2.3. Derivation of median mass diameter

To estimate the mass distribution function $M(D)$, information about the relationship between crystal mass m and D, which depends on ice crystal habit, is required. Following Jackson et al. (2012), two different techniques were used for estimating $M(D)$. $M(D)$ was derived as

$$\mathrm{M(D)} = \sum_j \sum_k f_h(D_j)\alpha_h D_j^{\beta_k} N(D_j)\Delta D_j \qquad (1)$$

where $f_h(D_j)$ is the fraction of crystals in the bin centered at D_j having crystal habit h, α_h and β_h are habit-dependent coefficients listed in Table 2.2 that define the mass of an individual crystal $m = \alpha_h D_j^{\beta h}$, and $N(D_j)$ is the number distribution function for bin j with midpoint D_j and width ΔD_j. For 30 April 2008 and 1 November 2011 the 2.3 µm resolution CPI images were used to determine $f_h(D_j)$ using the habit classification scheme of Um and McFarquhar (2011). Because

the CPI has a smaller sample volume than the 2DC, the averaging period required to obtain a statistically significant sample (Hallet 2003) is larger, and hence the habit distributions were applied to each of the 2DC size distributions occurring within the CPI averaging period. The 3V-CPI was not installed on the NSF/NCAR C-130 on 25 October 2011, so $f_h(D_j)$ were derived from the 2DC data using the Holroyd (1987) technique for that day. The $M(D)$ was then used to derive the median mass diameter D_{mm}.

2.2.4. Overview of flights

Data from three flights were used in this study. On 25 October 2011, a 1000 hPa low pressure system moved south through southern Wyoming into Colorado and produced a deep precipitating stratocumulus that persisted over Cheyenne, WY for ~12 hr. On 1 November 2011, a 1012 hPa low pressure system moved south through southern Wyoming into Colorado and produced another deep precipitating system that persisted over Cheyenne, WY for ~12 hr. On both sorties the NSF/NCAR C-130 penetrated the deformation zone of the system over Cheyenne, WY in an ascending spiral pattern from a height of 2 km to a height of 7 km for 1 hour followed by a descending spiral pattern for 30 min from a height of 7 km to a height of 2 km at airspeeds from 90 to 150 m s^{-1} and sampling temperatures 0℃ to -30℃.

On 30 April 2008, cold (< 0℃) air from a 1036 hPa high pressure system over the North Pole encountered a warm (> 0℃) airmass from a 1006 hPa low pressure center located about 500 km east of Fairbanks, AK. This produced a cold front that provided forcing for a precipitating multi-layer mixed-phase stratocumulus cloud system over Nenana and Fairbanks. The NRC Convair-580 ascended from the ground at Fairbanks to 7 km above Nenana for 20 min, and then performed a descending spiral from a height of 7 km to a height of 1.5 km for 15 min followed

by a 15 min ascending spiral from a height of 1.5 km to a height of 7 km through the cloud system before returning to Fairbanks. The NRC Convair-580 sampled from 0.5 km (-5℃) to 7 km (-40℃) at airspeeds from 50 to 120 m s^{-1}.

No data on these 3 flights were collected in exclusively liquid-phase conditions. A phase identification scheme (McFarquhar et al. 2007; Jackson et al. 2012) using information from a Rosemount icing detector, the shape of forward scattering spectrometer probe size distributions and manual inspection of 2DC/CPI imagery divided all 1-s averaged measurements in clouds into mixed- and ice-phase conditions.

2.3. Interarrival time analysis

This section provides the basis for defining the threshold particle interarrival time Δt below which all particles are identified as shattered artifacts. The dependencies on the frequency distribution of Δt are also explored using a Monte Carlo simulation.

To remove the greatest number of particles from the shattered particle mode while removing the fewest from the natural particle mode at larger Δt, Field et al. (2006) calculated the best fit of a normalized frequency distribution $\frac{dP(\Delta t)}{d\ln\Delta t}$ to a bimodal Poisson probability density function given by

$$\frac{dP(\Delta t)}{d\ln\Delta t} = (1-A)\frac{\Delta t}{\tau_1}\exp\left(-\frac{\Delta t}{\tau_1}\right) + (A)\frac{\Delta t}{\tau_2}\exp\left(-\frac{\Delta t}{\tau_2}\right) \quad (2)$$

where τ_1, τ_2 represent the Δt for the natural and shattered particle modes respectively, and A represents the relative contribution of the shattered particle mode to $\frac{dP(\Delta t)}{d\ln\Delta t}$. Field et al. (2006) classified particles with $\Delta t_s < 2\tau_2$ as shattered artifacts in order to remove 90% of the particles

from the second mode. The initial particle arriving before this identified artifact, which frequently had $\Delta t_s > 2\tau_2$, was also classified as a shattered artifact.

Normalized frequency distributions of Δt from the standard and modified 2DCs for all three flights are shown in Figure 2.2 and Figure 2.3 for ice-phase and mixed-phase conditions, respectively. All frequency distributions exhibit two modes: the first mode at larger Δt represents natural particles and the second mode at smaller Δt represents shattered artifacts (Field et al. 2003; 2006). In order to account for natural particles that may be removed by excluding particles with $\Delta t_s < 2\tau_2$, the number concentrations are multiplied by $\frac{1}{2e^{-\tau_2/\tau_1}-1}$ (Field et al. 2006). Fits to the normalized frequency distributions for ice- and mixed-phase conditions sampled during each flight were performed using Eq. (2) with the results depicted in Figures 2 and 3. For the mixed-phase conditions encountered on all three sorties, $\frac{dP(\Delta t)}{d\ln\Delta t}$ is still bimodal. The A for the 2DC with standard tips varies from 0.20 to 0.29 while A from the 2DC with modified tips varies from 0.00 to 0.17 mixed phase conditions. Meanwhile, A from the 2DC with standard tips varies from 0.20 and 0.29 and A from the 2DC with modified tips varies from 0.06 to 0.17 in ice phase conditions. This suggests shattered artifacts are likely still present in the mixed-phase conditions due to the presence of ice.

The τ_1 and τ_2 were, in general, larger for ISDAC than for IDEAS-2011. In order to assess the differences in τ_1, τ_2, A, and $\frac{dP(\Delta t)}{d\ln\Delta t}$ between projects, Monte Carlo simulations of particle shattering events were performed. For these simulations, particles with the median maximum dimension of 650 μm recorded by the 2DC with modified tips with concentration $N_{natural}$ were assumed to arrive at a sample area SA of 125 mm^2 for the 64 diode 2DC and of 74.6 mm^2 for the 32 diode 2DC. The particles were travelling at a true airspeed TAS and were randomly

distributed according to Poisson statistics. Particles also arrived according to Poisson statistics at the location of the probe tips, which had a cross sectional area *TA* of 92 mm^2. Upon contact, the particles were assumed to shatter into a train of *n* fragments that were distributed according to Poisson statistics across some length *l* in the direction of the flight as demonstrated in Figure 4. The probability of a single fragment entering the sample volume is given by *K* and was assumed to be independent of the probability of any other fragment entering the volume. Thirty second simulations with varying $N_{natural}$, *TAS*, *l* and *n* were run 100 different times to create 100 different versions of $\frac{dP(\Delta t)}{d\ln\Delta t}$. The τ_1, τ_2, *A* were then determined from the best fit of Equation (2) to the average of the 100 versions of $\frac{dP(\Delta t)}{d\ln\Delta t}$.

The sensitivity of τ_1, τ_2, and *A* to $N_{natural}$, *l*, *n*, and *TAS* were explored in these idealized simulations and provide guidance on what factors determine the locations of and relative importance of modes corresponding to natural and shattered particles. The results are shown in Figure 2.5, with the coefficients of the linear best fit lines, Pearson coefficient *R*, and the significance level of the correlation coefficient p_r according to an *F*-test listed in Table 3. For each panel in Figure 2.5, one variable was allowed to vary, while the other variables were kept constant at the control values listed in the caption of Figure 2.5. The τ_1, τ_2, and *A* are assumed to be correlated in a statistically significant sense and dependent on $N_{natural}$, *l*, *n*, or *TAS* when $p_r <$ 0.05 and when τ_1, τ_2, and *A* vary by more than the width of the 95% confidence interval. The range of $N_{natural}$ and *TAS* used in the simulations was selected from the ranges observed during ISDAC and IDEAS-2011. The *l* varies from 0 to 10 cm as Korolev et al. (2011) stated that *l* does not typically exceed 10 cm. The formvar replicator data of Vidaurre and Hallett (2009) was used to define the range of *n* values from 0 to 200. From Table 2.3 and Figures 5a, d and g it is seen that τ_1 is inversely correlated with and dependent on $N_{natural}$, *TAS*, but not with *l* or *n*. In addition,

τ_1 does not change when K decreases from 5% to 0.5%. The decrease of τ_1 with $N_{natural}$ is consistent with ISDAC and IDEAS-2011 observations because particles arrived in the sample volume more frequently when there were more of them; similarly, increases in *TAS* lead to more frequent arrival of particles. On the other hand, τ_1 is independent of the number or spacing between shattered particles or how many shattered particles enter the sample volume because it is describing only the mode of naturally occurring particles.

In Table 2.3, τ_2 is significantly correlated with and dependent on $N_{natural}$ and l, and inversely correlated with and dependent on *TAS* and n. Artifacts arrive in the sample volume more frequently if either the number of artifacts or airspeed increases, or if the same number of artifacts is spread over a smaller distance. The τ_2 is also smaller when K is larger because a larger fraction of particles enter the sample volume. Although A is sometimes correlated with $N_{natural}$, l, and *TAS* with $p_r < 0.05$, A does not depend on these variables because it is approximately constant. A is correlated with n because when more fragments are generated by a collision there are more particles to contribute to the mode of $\frac{dP(\Delta t)}{d \ln \Delta t}$ corresponding to shattered artifacts. The lack of dependence of A on *TAS* is probably not realistic because the number of fragments generated by a collision may increase with increasing *TAS*; however, there were no data to determine a relationship between n and *TAS* and hence no such dependence was assumed.

These simulations show that τ_1 is correlated with $N_{natural}$, *TAS* while τ_2 also depends on l, and n as well as $N_{natural}$ and *TAS*. However, the n, k, and l could not be identified from the 2DC observations. Therefore, to place the idealized simulations in context of the ISDAC and IDEAS-2011 observations, simulations where performed where parameters related to particle shattering, (i.e. n, k, and l) were kept constant while parameters related to how natural particles arrive in the

sample volume were varied corresponding to each 1 s averaged *TAS* and $N_{natural}$ measured by the 2DCs with modified tips processed using the shattered artifact removal algorithms. Figure 2.6 shows $\frac{dP(\Delta t)}{d\ln\Delta t}$ generated from all the simulations for ISDAC and IDEAS-2011 separately. As in the observations shown in Figure 2.2, the τ_1 of 0.015 s for ISDAC is greater than the τ_1 of 0.0053 s for IDEAS-2011. Further, Figure 2.6 shows that there is reasonable similarity in the $\frac{dP(\Delta t)}{d\ln\Delta t}$ obtained from the simulations and those obtained from Poisson statistics using the given τ_1, τ_2 and *A* in Eq. (2) from the simulations. But, the simulated $\frac{dP(\Delta t)}{d\ln\Delta t}$ is wider than that generated by Eq. (2) for the mode at τ_1. Since τ_1 is related to *TAS* and $N_{natural}$, this is likely caused by the superposition of many different natural particle modes of $\frac{dP(\Delta t)}{d\ln\Delta t}$ from periods with different *TAS* and $N_{natural}$ periods during ISDAC and IDEAS-2011. This, and the correlation between τ_1 and $N_{natural}$, and τ_1 and *TAS*, show that differences in τ_1 between ISDAC and IDEAS-2011 can be explained by differences in how often natural particles are arriving at the probe sample volume.

In Figure 2.6, τ_2 for ISDAC is 1.1 x 10^{-4} s, higher than τ_2 of 9.1 x 10^{-5} s for IDEAS-2011. The *A* from the 2DC with modified tips is 0.20 for ISDAC and 0.14 for IDEAS-2011, whereas the *A* from the 2DC with standard tips is 0.20 and 0.28 for ISDAC and IDEAS-2011, respectively. Therefore, the changes in simulated τ_2 and *A* between ISDAC and IDEAS-2011 are smaller than, but consistent with, changes in the observed τ_2 and *A*. Therefore, even though τ_2 is related to *TAS* and $N_{natural}$, the fact that simulated changes are smaller than observed changes in τ_2 and *A* show that the variability in *n*, *k*, and *l* from changes in ice crystal habit, size, density, impact velocity, angle of impact, temperature, and angle of attack (Korolev et al. 2011; 2013b) are likely greater factors in determining τ_2 and *A* than changes in *TAS* and $N_{natural}$. In addition, the different mounting locations of the 2DCs during ISDAC and IDEAS-2011 during ISDAC and

IDEAS-2011 could have affected the shattered artifacts and contributed to the variations in magnitude and location of the shattered particle mode for ISDAC and IDEAS-2011. The fact that the simulations cannot reproduce the differences in τ_2 and *A* during ISDAC and IDEAS-2011 shows that further understanding on how particle size, habit, air speed, angle of attack, and probe configuration affect *n*, *k*, and *l* are required.

The observation of bimodal interarrival time distributions also could occur if there was a plausible natural mechanism through which ice crystals could form such clusters. One potential explanation for such particle clustering would be the inertial response of particles within a turbulent flow, sometimes referred to as preferential concentration (Siebert et al. 2010). Careful laboratory measurements of the inertial clustering of water droplets in homogeneous isotropic turbulence (Saw et al. 2012) showed that the correlation function increases smoothly and monotonically with decreasing length scale. Figure 2.4 from Saw et al. (2012) shows that the correlations at $r/r_K > 30$ (where r is the droplet separation distance and r_K is the Kolmogorov microscale, approximately 1 mm for typical cloud turbulence) are due to turbulent mixing and the correlations at scales less than $r/r_K \approx 30$ are due to inertial clustering. These measurements were for weakly inertial particles with relatively minimal gravitational settling, so ice crystals would be expected to behave somewhat differently. In particular, ice particles are typically more sparse in tropospheric clouds than are water drops, so it may take long times for them to come together due to turbulence. Further, larger differences in terminal fall speeds of ice crystals may prohibit the maintenance of such crystals even if they form. Thus, the explanation of natural clustering seems unlikely for the quick interarrival times of ice crystals.

2.4. Uncertainty in *N(D)* due to shattered artifacts

2.4.1. Assessment of algorithms and modified tips

In the remainder of this paper, subscripts *s* (*m*) are used to denote standard (modified) tips, and superscripts *a* (*na*) denote that algorithms were (not) used to derive number concentration *N*, from the 2DC probes. For example, N_s^{na} is the number concentration of particles from the 2DC with standard tips without the use of algorithms. The mean and range of the ratio of the number concentration from the 2DC using standard tips, N_s, to that from the 2DC using modified tips, N_m, in differing size ranges is plotted as a function of median mass diameter D_{mm} derived from the 2DC with modified tips for ice phase conditions during IDEAS-2011 in Figure 7 and during ISDAC in Figure 2.8. The mean and ranges were determined from all 10-s averaged SDs within the 0.15 mm D_{mm} bin on the horizontal axis. The ISDAC data are cut off at a smaller D_{mm} due to the use of a 32 photodiode 2DC instead of the 64 photodiode 2DC used during IDEAS-2011. Even with application of reconstruction techniques, only particles with $D < 1.6$ mm could be detected with any statistical significance during ISDAC. Hence, D_{mm} may have been underestimated for the ISDAC data. Figures 2.7a, b, 2.8a, and b show that $N_s^a/N_m^a > 2$ for $125 < D < 300\ \mu m$ and $300 < D < 500\ \mu m$ for $D_{mm} > 1.6$ mm meaning that algorithms do not remove all shattered particles. This is consistent with Korolev et al. (2011, 2013b)'s findings from the Airborne Icing Evaluation Experiment.

During IDEAS-2011, the Pearson correlation coefficient *R* between N_s^a/N_m^a of $125 < D < 300\ \mu m$ with D_{mm} was 0.47, and between N_s^{na}/N_m^{na} at $300 < D < 500\ \mu m$ and D_{mm} was 0.42. For ISDAC, these correlation coefficients were 0.39 and 0.30 respectively The increase of N_s^{na}/N_m^{na} with D_{mm} is consistent with previous studies that showed the likelihood of a particle shattering on

the 2DC hemispheric tips is related to particle size (Korolev and Isaac 2005), particle mass and impact velocity (Vidaurre and Hallett 2009). Another significant finding is that $N_s^a/N_m^a < 2$ for $D > 500\ \mu m$ indicating that the shattered artifacts are smaller than $500\ \mu m$ for the two diverse datasets examined. However, since there were no environments with little to no shattering potential sampled, nor were there any flights where 2 2DCs with identical tips flown on the same aircraft, a comparison to determine the possible effects of the differences in 2DC electronics and mounting position on N_s/N_m was not performed, yet these factors can potentially have an effect on N_s/N_m. Despite these differences, a similar threshold for shattered artifacts as in other projects using probes with different response times and resolutions (Korolev et al. 2011; 2013a). Therefore, this 500 µm threshold seems to apply to most 2DC probes and over the types of ice crystals observed in field projects thus far.

Figure 2.9 examines the effect of the artifact removal algorithms on concentrations in different size ranges by plotting the ratio of concentrations determined using algorithms N^a to those calculated without use of algorithms N^{na} as a function of D_{mm} for probes with both standard and modified tips. N_s^{na}/Ns^a for $125 < D < 300$ µm and $300 < D < 500$ µm increases from 1 to 10 as D_{mm} increases during IDEAS-2011. Increases with D_{mm} were also noted for ISDAC, with, for example, N_s^a/N_m^a consistently greater than 2 for $125 < D < 300$ µm. This shows that the use of algorithms alone did not remove all shattered artifacts for particles $125 < D < 500$ µm, consistent with Korolev et al.'s (2011) findings. This indicates that the use of both shattered artifact removal algorithms and modified tips is important for removing shattered artifacts in 2DC data.

The relative importance of the probe tips and shatter removal algorithms can be compared by examining Fig. 2.7 and Fig. 2.9. For IDEAS-2011, the impact of the shatter removal algorithms with standard probe tips is seen in Fig. 2.9 as N_s^{na}/N_s^a at $125 < D < 300$ µm

and $300 < D < 500$ µm vary from 1 to 10, and 1 to 5 respectively. In Fig. 2.7 the modified tips remove a similar fraction of particles when algorithms are not applied first with N_s^{na}/N_m^{na} at $125 < D < 300$ µm and at $300 < D < 500$ µm varying from 1 to 10 and 1 to 5 respectively. Further, the differences between concentrations determined by applying and not applying algorithms to data collected by probes with modified tips, and determined from the standard and modified probes after algorithms are first applied are similar. Thus, for the IDEAS-2011 data the modified tips removed a similar number of artifacts as the shatter removal algorithms.

However, a different finding is obtained when inspecting the ISDAC data. In Figures 2.9c, d, N_s^{na}/N_s^{a} at $125 < D < 300$ µm and $300 < D < 500$ µm vary from 1 to 3 while N_s^{na}/N_s^{a} at $125 < D < 300$ µm and $300 < D < 500$ µm vary from 1 to 7 in Figure 2.8, indicating that the tips removed more particles than did the algorithms. The reason for the difference in findings between IDEAS-2011 and ISDAC is not known, but may be due to the different response times of probes affecting how well the 2DCs detect small particles or may be due to different distributions of particles or mounting locations of the probes in the two projects. Given the differences between IDEAS-2011 and ISDAC and given that probes with different response times, designs, configurations, and resolutions are still in use, the important conclusion is that both techniques must be applied to remove the majority of shattered artifacts for IDEAS-2011 and ISDAC.

Figures 2.10a and b show the mean $N_s^{na}(D)$, $N_s^{a}(D)$, $N_m^{na}(D)$, $N_m^{a}(D)$ for all ice phase conditions during IDEAS-2011 and ISDAC respectively. The bootstrap technique (Efron and Tibishrani 1993, McFarquhar and Heymsfield 1997) generates alternate versions of the average SDs by randomly drawing, with replacement, from the population of SDs. The standard deviations in $N(D)$ from the alternate average SDs from the bootstrap technique were then

displayed as error bars. These error bars show that $N_s^{na}(D)$ is greater than $N_s^{a}(D)$, $N_m^{na}(D)$, $N_m^{a}(D)$ in a statistically significant manner for $D < 400$ μm. This indicates that the shattered artifacts most likely have $D < 400$ μm for IDEAS-2011 and ISDAC.

2.4.2. Comparison with N(D) measured by HOLODEC

The analysis in Section 2.4.1 showed that the use of modified tips and algorithms combined removed more shattered artifacts than the use of either technique alone. However, it is not known whether the two methods combined remove all artifacts. As shown in Korolev et al. (2013b) they may also remove naturally occurring crystals. To further investigate the fidelity of the size distributions derived from the 2DC, the 2DC size distributions (SDs) were compared against those measured by the HOLODEC because its unique 3D nature gives an alternate viewpoint for detecting and eliminating shattered artifacts, and hence offers an independent confirmation on the extent to which the modified tips and algorithms remove shattered particles.

Figure 2.10c shows *N(D)* measured by the 2DCs with and without modified tips and by the HOLODEC. The HOLODEC has Korolev-style anti-shattering tips installed which may account for the minor difference between the calculated *N(D)* for $D > 125$ μm when shatter detection algorithms are and are not applied to *N(D)* from the HOLODEC. Both $N(125 < D < 300\ \mu m)$ from the HOLODEC and from the 2DC modified tips both have values around 5×10^{-3} L^{-1} μm^{-1} in Figure 2.10c. It is emphasized that the 2DC and HOLODEC instruments are fundamentally different measurement methods (focused imaging versus digital reconstruction of an interference pattern, respectively), with distinct spatial sampling methodology (continuous sampling of single particles versus instantaneous sampling of volumes containing multiple particles, respectively) and distinct identification of shattering events (temporal statistics versus

spatial statistics, respectively). *N(125 < D < 300 μm)* from the 2DC with standard tips is around 3 x 10^{-2} L^{-1} μm^{-1}, or about 6 times higher than *N(D)* from both the HOLODEC and 2DC with modified tips. Thus, this further demonstrates that the modified tips should be used in combination with shattered artifact removal algorithms for the cloud conditions sampled on 25 October 2011. Only particles with D > 125 μm from the HOLODEC probe are included in Figure 10c. Because there are few, if any, probes that provide accurate measurements of the numbers of particles with D < 125 μm, such particles may also make large and unknown impacts on quantifying the effects of shattering on *N(D)*. Further, due to the limited amount of HOLODEC data available for this study, more data collected in projects with a 2DC with standard tips, a 2DC with modified tips, and a HOLODEC should be collected for a wide range of aircraft, probe configurations, and cloud conditions to test the generality of the conclusion presented here.

2.4.3. Dependence of shattering on other parameters

Although Section 2.4.1 showed that $N_s^{na}/N_m^{na} > 1$ for $D < 500\ \mu m$ with N_s^{na}/N_m^{na} increasing with $D_{mm,}$ there was still substantial spread in N_s^{na}/N_m^{na} for the same $D_{mm,}$ showing other factors influence the amount of shattering. To identify the cause of this spread, the correlation between N_s^{na}/N_m^{na} for $125 < D < 300\ \mu m$ and $300 < D < 500\ \mu m$ with true air speed, angle of attack, pitch angle, roll angle, airspeed, and temperature *T* was examined. Algorithms to remove shattered artifacts were not used in this analysis to more easily identify the presence of shattered artifacts.

Table 2.4 shows that N_s^{na}/N_m^{na} for $125 < D < 300\ \mu m$ and $300 < D < 500\ \mu m$ both had a correlation coefficient of $R \approx 0.5$ with *T*. The correlation coefficients of N_s^{na}/N_m^{na} for $125 < D <$

300 μm and *300 < D < 500 μm* with airspeed ($R \approx -0.3$), roll angle ($R \approx -0.2$), and angle of attack ($R \approx -0.1$) were weaker. No angle of attack data were available on 30 April 2008 during ISDAC. The likelihood of shattering is expected to be proportional to the particle's impact velocity (Vidaurre and Hallett 2009), and the likelihood of shattered particles entering the sample volume is expected to increase when the drag force (pressure) is lower (Korolev et al. 2013a). Therefore, the correlation coefficients of ~0.6 between N_s^{na}/N_m^{na} and pressure, and the weak correlation coefficient of $R \approx -0.3$ between N_s^{na}/N_m^{na} at *125 < D < 300 μm* and *300 < D < 500 μm* with airspeed at first seem inconsistent with expectations. However, the correlation coefficient between airspeed and *T* was -0.51 and, between airspeed and pressure was 0.97 as the aircraft flew faster at higher altitudes and lower pressures. Further, D_{mm} were smaller at lower temperatures and pressures, hence explaining the weaker correlations. The counterintuitive lack of correlations with airspeed, angle of attack and pitch attitude show that particle type is perhaps the dominant factor in influencing the amount of shattering rather than aircraft parameters. Therefore a larger dataset stratified by particle type is needed to further address the causes of the scatter illustrated in Section 2.4.1.

Table 2.5 shows the correlations calculated over time periods when $D_{mm} < 1.6$ mm and $D_{mm} > 1.6$ mm separately in order to account for the dependence of N_s^{na}/N_m^{na} at *125 < D < 300 μm* and *300 < D < 500 μm* on D_{mm}. The correlation coefficient between N_s^{na}/N_m^{na} at *125 < D < 300 μm* and *300 < D < 500 μm* and *T* and pressure are still around ~0.5 for both $D_{mm} < 1.6$ mm and $D_{mm} > 1.6$ mm in Table 2.5.

The N_s^{na}/N_m^{na} in two different size ranges as a function of D_{mm} for T > and < -8℃ for data collected during IDEAS-2011 and ISDAC is shown in Figure 2.11. The N_s^{na}/N_m^{na} at *125 < D < 300 μm* and *300 < D < 500 μm* is greater when *T* > -8℃ (the median *T* observed),

according to a Mann-Whitney U Test at a 99% significance level for D_{mm} > 1600 μm for IDEAS-2011. However, they were similar for T > -8℃ and T < -8℃ during ISDAC. It is hard to interpret shattering solely in terms of T as particle growth habits are determined by a multitude of factors and because crystals are typically not observed at the T where they grow. Figures 2.11e and f show representative images of ice particles obtained by the CPI with D > 1 mm from the 1 November 2011 flight for T < -8℃ and T > -8℃ . At T < -8℃ , rimed particles were observed 11% of the time in ice phase conditions, while they were observed 62% of the time in ice phase conditions at T > -8℃ . This suggests the dependence on T for the IDEAS-2011 data might actually originate from a dependence on riming. The lack of dependence on T for ISDAC is consistent with this result because rimed particles were observed only 0.03% of the time for all temperatures sampled during that project

To quantitatively determine the dependence of N_s^{na}/N_m^{na} on the degree of riming, Figure 2.12 shows N_s^{na}/N_m^{na} in two different size ranges as a function of D_{mm} and separated according to whether particles were identified as graupel or dendrites following Holroyd (1987), namely whether the perimeter divided by area of particles with D > 0.5 mm S was < 0.013 μm^{-1} versus when S > 0.013 μm^{-1}. Particles with D < 500 μm were not used in the calculation to avoid contamination from shattered artifacts. The medians of N_s^{na}/N_m^{na} for 125 < D < 300 μm and 300 < D < 500 μm are higher in graupel at a 99% significance level according to a Mann-Whitney U test for 1.6 mm < D_{mm} < 2.2 mm during IDEAS-2011, and were the same for D_{mm} < 1.6 mm during IDEAS-2011 and during ISDAC. To confirm the increased shattering when particles were identified as graupel, Figure 2.13 shows greater contributions of the shattered particle mode to $\frac{dP(\Delta t)}{d\ln\Delta t}$ for these time periods. In fact the difference between A from the 2DC with standard and modified tips when particles were identified as graupel is 0.46 compared to 0.20 when particles

were identified as dendrites. Thus it is hypothesized that graupel particles shatter into a greater number of fragments due to either their increased mass or more fragile structure. The lack of a relationship between N_s^{na}/N_m^{na} and T or between N_s^{na}/N_m^{na} and S on 30 April 2008 is consistent with the lack of riming observed on this sortie.

Due to the uncertainty in the concentrations of particles with $D > 1.6$ mm during ISDAC from the use of the 32 photodiode 2DC, particles with 1.6 mm < D < 3.2 mm were not considered in the calculation of D_{mm} for ISDAC. Since particle mass scales approximately with the square of D (e.g. Brown and Francis 1995), D_{mm} is affected by the absence of large particles in the ISDAC calculations. To test the sensitivity of excluding particles with $D > 1.6$ mm, Figure 16 shows N_s^{na}/N_m^{na} in two different size ranges as a function of D_{mm} for IDEAS-2011 divided according to whether particles were identified as graupel or dendrites, with only particles with $0.125 < D < 1.6$ mm considered in the calculation of $D_{mm.}$ Figure 2.14 shows that the medians of N_s^{na}/N_m^{na} for $125 < D < 300$ μm and for $300 < D < 500$ μm are still higher for $1225 < D_{mm} <$ 1500 μm compared to when particles were identified as graupel at a 99% significance level. This shows that the dependence of N_s^{na}/N_m^{na} on particle habit is not sensitive to exclusion of particles with $D > 1.6$ mm.

2.5. Conclusions

In situ aircraft measurements of ice crystal number distribution function *N(D)* were acquired by a 2DC probe with standard tips, mounted immediately adjacent to a 2DC probe with tips modified to reduce the number of shattered fragments entering the sample volume on the NRC Convair-580 during ISDAC and on the NSF/NCAR C-130 during IDEAS-2011. The impact of the tips and the artifact removal algorithms (Korolev and Isaac 2005; Field et al 2006)

on the derivation of ice crystal concentrations N in different size ranges was assessed as a function of cloud sampling conditions and aircraft parameters in this study. A collocated HOLODEC probe used during IDEAS-2011 also identifies and eliminates shattered particles because the artifacts appear as clusters in the holograms; insight into the 2DC shattering problem is obtained through comparison against the HOLODEC's size distributions. A series of Monte Carlo simulations with varying ice crystal concentrations, numbers of shattered particles, length of train of shattered fragments, and probabilities of shattered particles entering the sample volume were also conducted and used to interpret the measured distribution of particle interarrival times by the 2DCs. The principal conclusions of this study are as follows:

1. The differences in the interarrival time modes corresponding to natural particles from the 2DCs were likely due to the different number concentrations of natural particles observed, different 2DC sample volumes, and different airspeeds flown between ISDAC and IDEAS-2011. But, the differences between projects in the interarrival time modes corresponding to shattered particles were likely due to the different ways the particles shattered on the 2DC probe tips. In addition, differences in the number of artifacts the 2DCs could have observed due to observed differences in natural number concentration, 2DC sample volumes, and airspeed could have contributed to the variations in the modes.
2. The ranges of ratios between the number concentration from the 2DC with standard tips to the number concentration from the 2DC with modified tips without the use of algorithms N_s^{na}/N_m^{na} at $125 < D < 300$ µm and $300 < D < 500$ µm were 1 to 10 and 1 to 5 respectively, and were correlated with D_{mm} with a correlation coefficient of ~0.5, suggesting that shattered artifacts most likely have $D < 500$ µm. As similar conclusions have been reached by Korolev et al. (2011; 2013b), this study extends the generality of

this result to a different aircraft in a wider range of temperatures than those encountered in Korolev et al. (2011; 2013b).

3. For IDEAS-2011, the difference between number concentration from the 2DC with standard tips without use of shattered artifact removal algorithms to that with use of algorithms N_s^{na}/N_s^{a} at $125 < D < 300$ μm and $300 < D < 500$ μm varied from 1 to 10 and 1 to 5 respectively, equal to the N_s^{na}/N_m^{na} of 1 to 10 and 1 to 5 in these same size ranges. This indicates that the modified tips removed as many shattered artifacts as the use of anti-shattering algorithm. For ISDAC, N_s^{na}/N_s^{a} at $125 < D < 300$ μm and $300 < D < 500$ μm varied from 1 to 7 while N_s^{na}/N_m^{na} at $125 < D < 300$ μm and $300 < D < 500$ μm varied from 1 to 3, indicating that algorithms removed fewer shattered artifacts than modified tips.

4. The *N(125 < D < 300 μm)* for the HOLODEC and for the 2DC with modified tips had values of around 5×10^{-3} L^{-1} μm^{-1}. Further, there was a factor of 6 difference in *N(D)* between the HOLODEC and from the 2DC with standard tips. This demonstrates that the modified tips used in combination with shattered artifact removal algorithms are removing artifacts with $125 < D < 300$ μm during IDEAS-2011. Therefore, both modified tips and anti-shattering algorithms should be applied to correct 2DC particle size distributions.

5. The N_s^{na}/N_m^{na} is larger and a greater number of particles classified as shattered artifacts when particles of $D > 500$ μm were identified as graupel and with $D_{mm} > 1.6$ mm during IDEAS-2011. This suggests graupel particles shatter into a greater number of fragments than dendrites of equivalent maximum dimension due to either their increased mass or their shape.

The comparisons made in this paper estimate the uncertainty due to shattered artifacts on *N(D)* from 2DCs in the cloud conditions sampled in three sorties during ISDAC and IDEAS-2011 and for the specific probes used. The response time of the 2DC used during IDEAS-2011 was lower than that from the 2DC used during ISDAC due to the upgraded electronics of the 64 diode 2DC, increasing the probability of detection of particles $D < 500$ μm. However, due to the limited subset of conditions encountered in this study, it is important to note that the results of this paper cannot necessarily be generalized to any other 2DC probe, cloud, probe installation, aircraft, or project. Despite these differences and limitations, some very common trends in the data that have been also observed in previous studies have been noted in this paper. Therefore, the differences between quantities calculated between probes with standard and modified tips, and processed with and without artifact removal algorithms provide guidance on the uncertainties that shattering induces on *N(D)* for cloud conditions commonly encountered in field projects using commonly used instrumentation.

It is important that these investigations be conducted for a variety of probes with different response times to understand the impact of shattering in a wider variety of conditions. Future studies should concentrate on acquiring data in ice clouds at colder temperatures (e.g. < -40°C), in a wider range of aircraft operating parameters (e.g. higher true air speeds), in a wider range of crystal habits (e.g. more rimed particles and graupel), using different probe installations (e.g. pairs of standard 2DCs with different configurations) and with a wider variety of probes (e.g. two-dimensional stereo probe, cloud imaging probe) with different designs of probe tips for deflecting shattered particles from the probe sample volume (e.g. Korolev et al. 2013a).

CHAPTER 2 FIGURES AND TABLES

Table 2.1. List of instrumentation from IDEAS-2011 and ISDAC.

Instrument	**Quantity**	**Use during ISDAC?**	**Use during IDEAS-2011?**
2DC with standard tips	*N(25 < D < 1600 μm)* (IDEAS-2011) *N(25 < D < 800 μm)* (ISDAC)	Yes	Yes
2DC with modified tips	*N(25 < D < 1600 μm)* (IDEAS-2011) *N(25 < D < 800 μm)* (ISDAC)	Yes – Korolev et al. (2011) design	Yes – 45 degree angle
CPI (Cloud Particle Imager)	High resolution images	Yes	Yes 01 Nov only
Rosement Icing Detector (RICE)	Presence of supercooled water	Yes	Yes
Cloud Droplet Probe (CDP)	*N(2 < D < 50 μm)*	No	Yes
Forward Scattering Spectrometer Probe (FSSP)	*N(2 < D < 50 μm)*	Yes	No

Table 2.1 (cont.)

Pitot-static system	Airspeed, pitch, roll angle, angle of attack	Yes	Yes
Rosemount sensor	Ambient air temperature	Yes	Yes

Table 2.2. List of *m-D* relationships used in Equation (1).

Habit	m-D relationships	α [g cm$^{-3-\beta}$]	β
Tiny	Brown and Francis (1995)	0.00294	1.9
Oriented	Brown and Francis (1995)	0.00294	1.9
Linear	Mitchell (1996) "Hexagonal columns"	0.00166 (125 < *D* < 300 μm) 0.000907 (*D* > 300 μm)	1.91 (125 < *D* < 300 μm) 1.74 (*D* > 300 μm)
Aggregate	Brown and Francis (1995)	0.00294	1.9
Irregular	Brown and Francis (1995)	0.00294	1.9
Graupel	Mitchell (1996) "Lump graupel"	0.049	2.8
Dendrite	Mitchell (1996) "Broad branched crystal"	0.000516	1.80
Plate	Mitchell (1996) "Hexagonal plate"	0.00739	2.45

Table 2.3. Numerical values of best fits of τ_1, τ_2, and A to N_n, *TAS*, and l for similatuions when k = 5% (columns 1,2) and k = 0.5% (columns 3,4). Significant correlations indicated by $p_r < 0.05$ and a statistically significant change are given in bold.

Fit displayed	*K = 5%*	*R*, p_r, (*k = 5%*)	*k = 0.5%*	*R*, p_r (*k = 0.5%)*
Best fit of τ_1 vs. *n*	$\tau_1 = -1.2*10^{-6}n + 0.076$	-0.26, 0.43	$\tau_1 = 4.1*10^{-6}n + 0.076$	0.61, 0.05
Best fit of τ_1 vs. *TAS*	**$\tau_1 = -0.00091TAS + 0.17$**	**-0.95, 0.00**	**$\tau_1 = -0.0095TAS + 0.18$**	**-0.96, 0.00**
Best fit of τ_1 vs. *l*	**$\tau_1 = -0.0002l + 0.0735$**	**-0.51, 0.02**	**$\tau_1 = -0.0002l + 0.0735$**	**-0.47, 0.04**
Best fit of τ_1 vs. N_n	**$\tau_1 = 0.026\exp(-0.07N_n)$**	**-0.90, 0.00**	**$\tau_1 = 0.027\exp(-0.07N_n)$**	**-0.90, 0.00**
Best fit of τ_2 vs. *n*	**$\tau_2 = -4.4 * 10^{-7}n + 6.7*10^{-5}$**	**-0.97, 0.00**	**$\tau_2 = -4.4 * 10^{-7}n + 1.2*10^{-4}$**	**-0.74, 0.01**
Best fit of τ_2 vs. *TAS*	**$\tau_2 = -9.4*10^{-4}TAS + 1.8$**	**-0.95, 0.00**	**$\tau_2 = 1.5*10^{-5}TAS + 2.9 \times 10^{-4}$**	**-0.96, 0.00**
Best fit of τ_1 vs. *l*	**$\tau_2 = 7.51 * 10^{-4}l + 0.0735$**	**-0.87, 0.00**	**$\tau_2 = 7.55 * 10^{-4}l + 0.027$**	**-0.86, 0.00**
Best fit of τ_1 vs. N_n	**$\tau_2 = -3.9 \times 10^{-7}N_n + 6.5 \times 10^{-5}$**	**-0.73, 0.01**	**$\tau_2 = -7 \times 10^{-6}N_n + 1.3*10^{-4}$**	**-0.80, 0.01**

Table 2.3. (cont.)

Best fit of A vs. n	$\boldsymbol{A = 0.0023n + 0.62}$	**0.64, 0.04**	$\boldsymbol{A = 0.0033n + 0.26}$	**0.88, 0.00**
Best fit of A vs. TAS	$A = 4.1\text{x } 10^{-5}TAS + 0.68$	-0.32, 0.34	$A = 8.7 \text{ x } 10^{-5}TAS + 0.19$	0.09, 0.80
Best fit of A vs. l	$A = -0.0008l + 0.689$	-0.87, 0.00	$A = -.0034l + 0.201$	-0.49, 0.03
Best fit of τ_1 vs. N_n	$A = -0.006N_n + 0.68$	-0.99, 0.00	$A = -0.0012N_n + 0.16$	-0.97, 0.00

Table 2.4. Correlation coefficient between N_s/N_m and specified parameter. Correlations above 0.03 are significant at a 95% significance level.

Parameter	$N_s/N_m(125 < D < 300\ \mu m)$	$N_s/N_m(300 < D < 500\ \mu m)$
Pitch angle	0.00	0.01
Roll angle	-0.13	-0.22
Temperature	0.50	0.50
Angle of attack	-0.09	-0.13
True air speed	-0.31	-0.32
Pressure	0.61	0.64

Table 2.5. Correlation coefficient between N_s/N_m and specified parameter when $D_{mm} < 1.6$ mm (left value) and when $D_{mm} > 1.6$ mm (right value). Correlations above 0.03 are significant at a 95% significance level.

Parameter	$N_s/N_m(125 < D < 300\ \mu m)$	$N_s/N_m(300 < D < 500\ \mu m)$
Pitch angle	-0.26, 0.06	-0.22, 0.07
Roll angle	0.09, -0.15	-0.15, -0.28
Temperature	0.55, 0.51	0.46, 0.56
Angle of attack	-0.34, 0.09	-0.28, 0.08
True air speed	-0.09, 0.34	-0.10, -0.40
Pressure	0.58, 0.53	0.50, 0.62

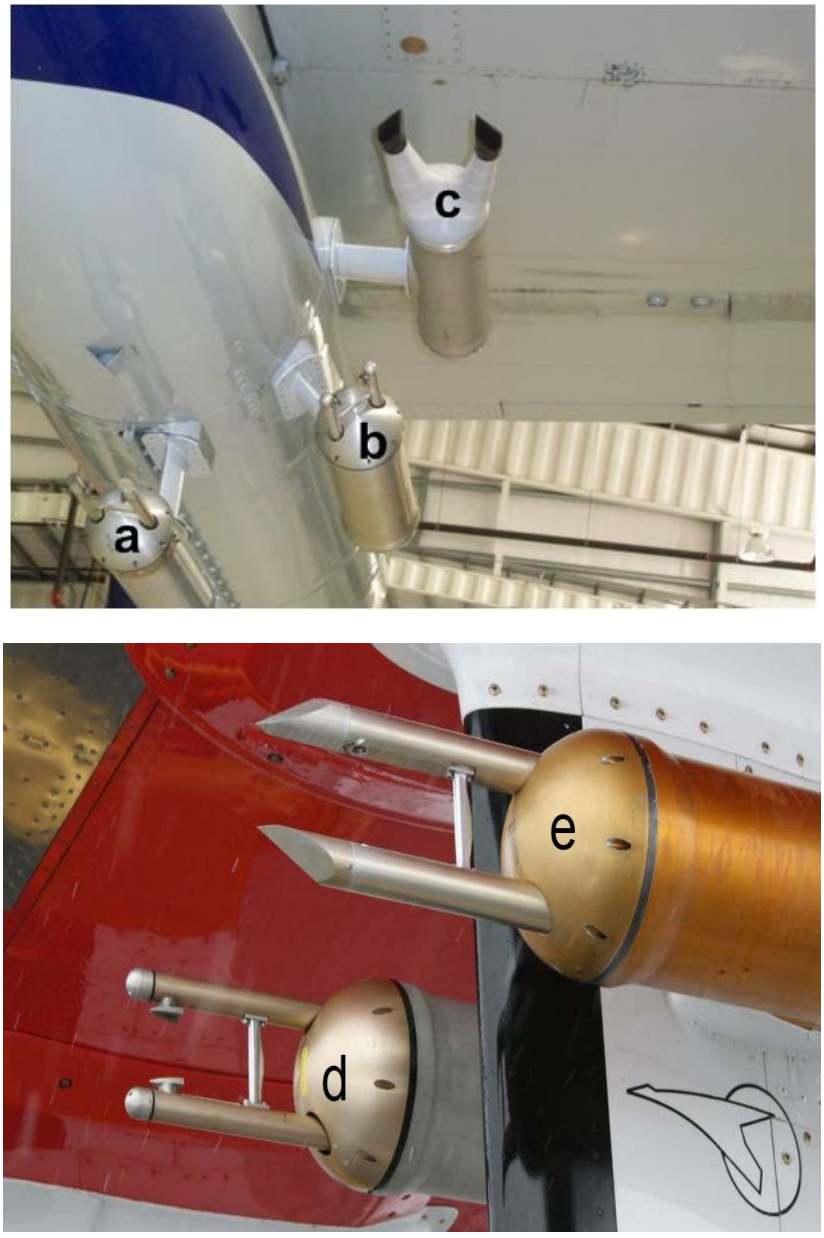


Figure 2.1. (a) Photograph of the 2DC with modified tips, (b) 2DC with standard hemispheric tips and (c) the HOLODEC as installed on the left pod of the NCAR C-130 during IDEAS-2011 (photo credit: Jeff Stith). (d) Photograph of the 2DC with standard tips and **(e)** modified tips installed on the left pod of the NRC-Convair 580 (photo credit: Alexei Korolev) during ISDAC.

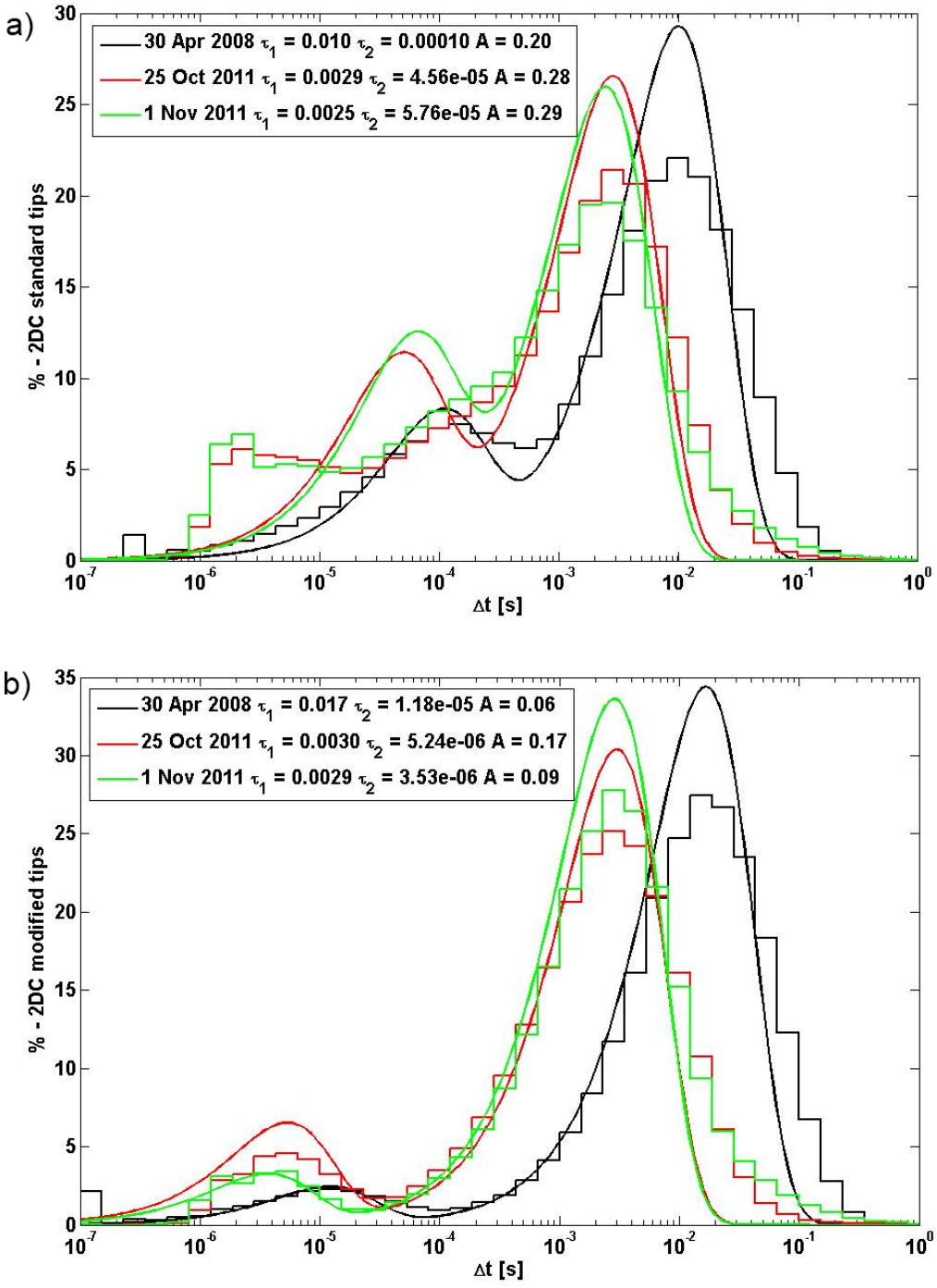


Figure 2.2. $\frac{dP(\Delta t)}{d \ln \Delta t}$ (stair step lines) and best fits (curves) for the sorties from the standard 2DC (a) and modified 2DC (b) for all ice phase time periods on 30 April 2008, 25 October 2011, and 1 November 2011.

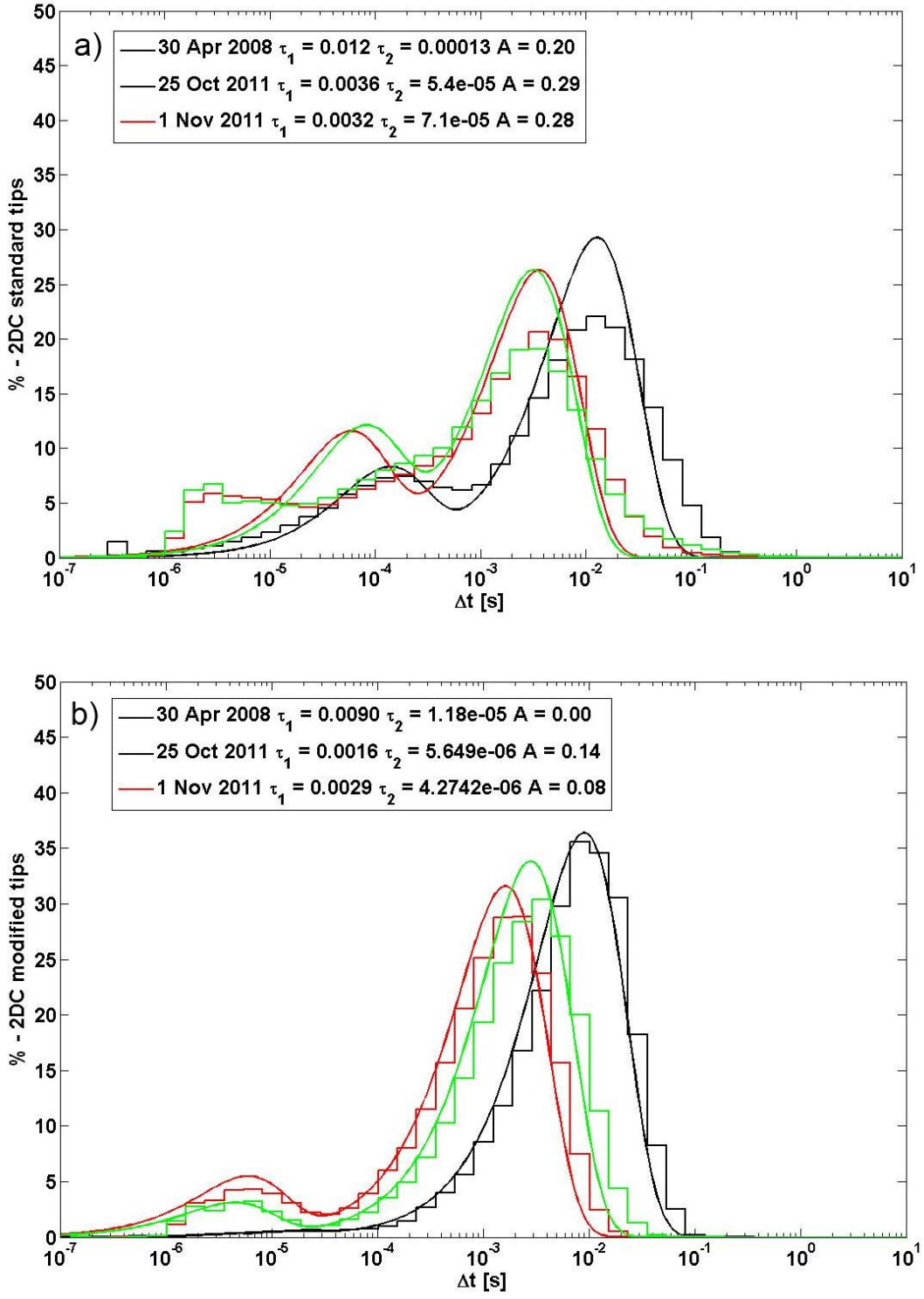


Figure 2.3. As in Figure 2.2(a,b) but for all mixed phase time periods.

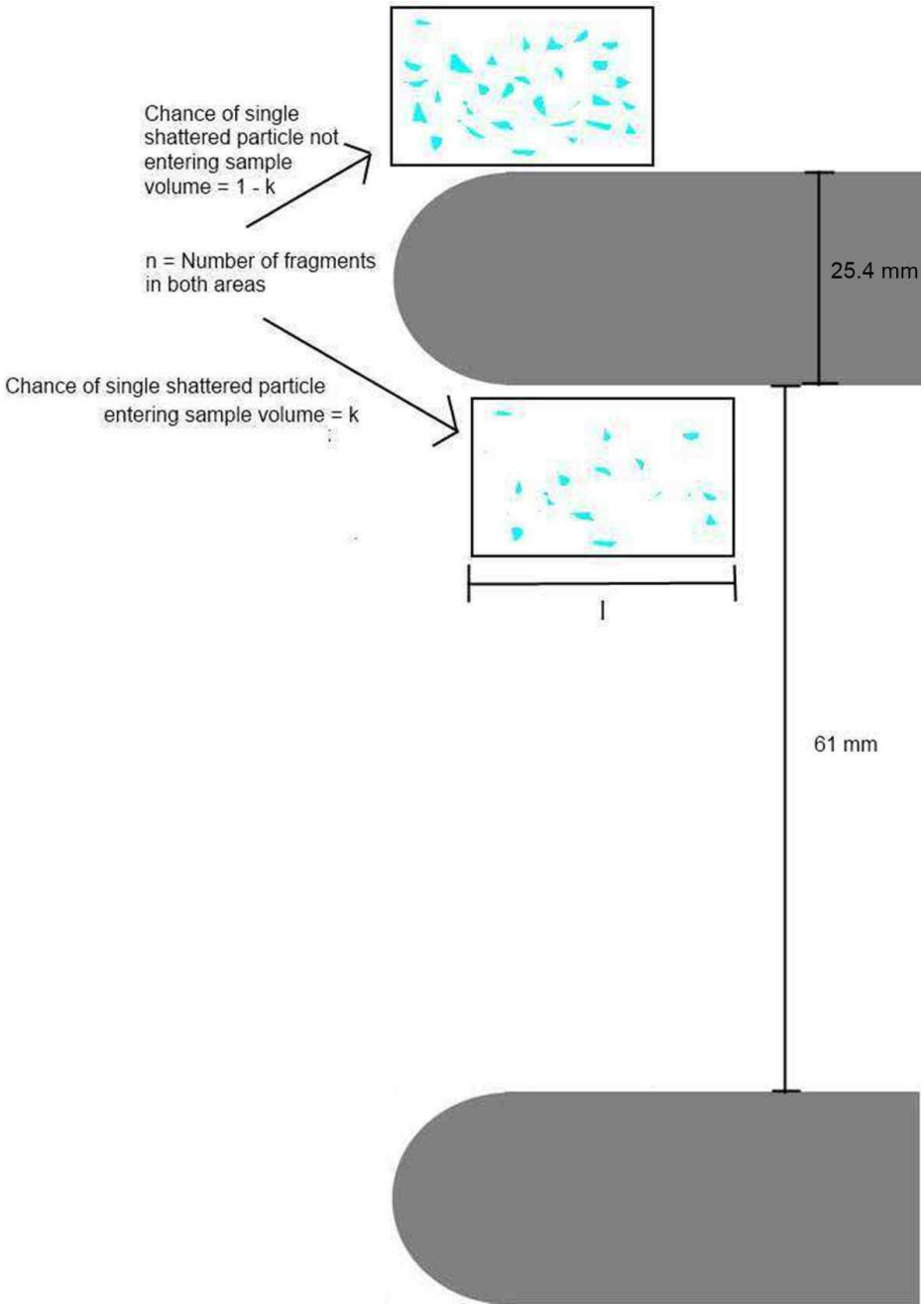


Figure 2.4. Conceptual diagram demonstrating simulation of particle shattering on a probe tip. Sample volume is below probe tip in this diagram. *l* = length of train of fragments in direction of flight.

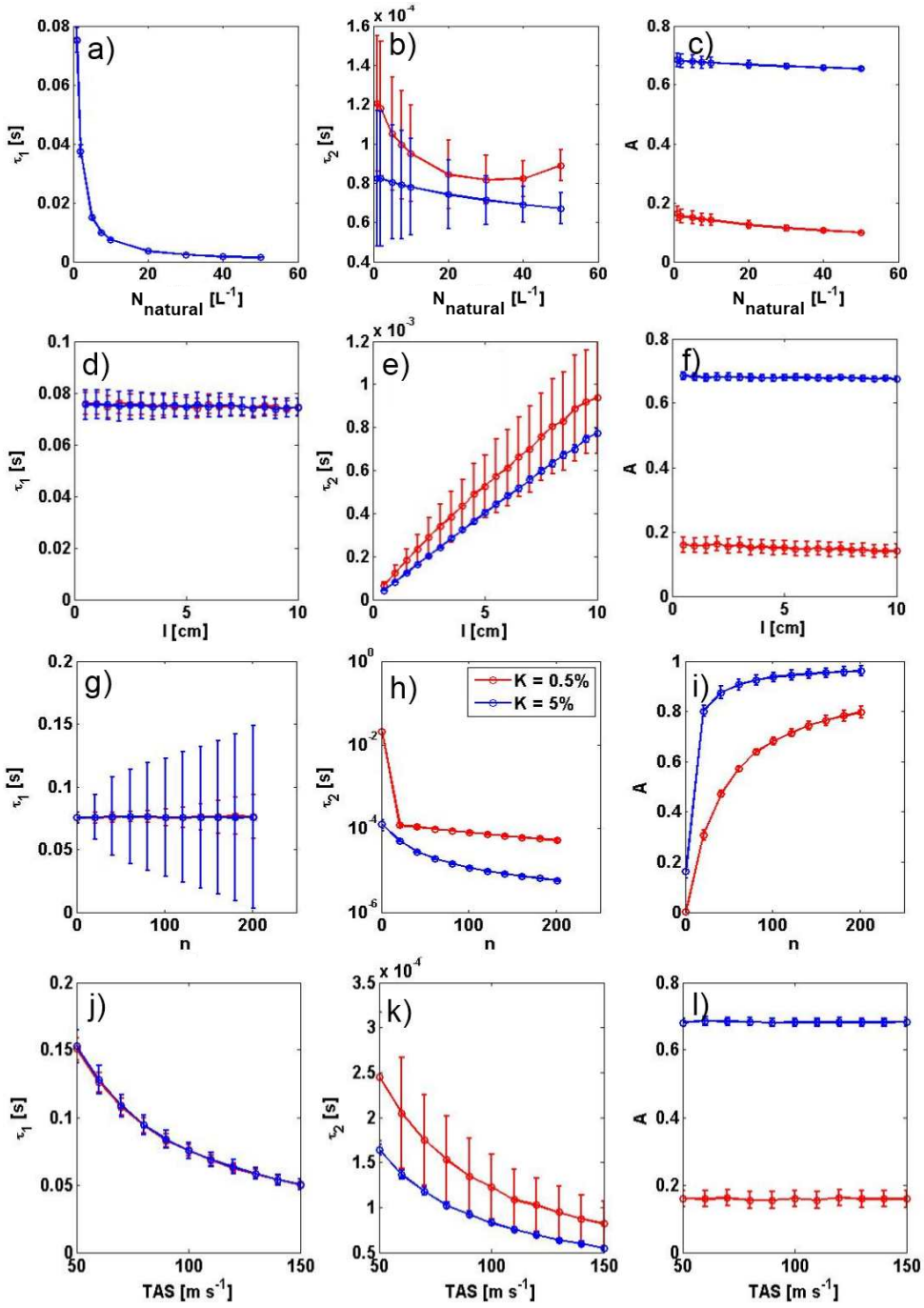


Figure 2.5. Best fit parameters τ_1, τ_2, and A to Equation (2) of the mean $P(\Delta t)$ of all ensemble simulations described in Section 3. 95% confidence intervals of τ_1, τ_2, and A are shown as errorbars as a function of $N_{natural}$, TAS, l, and n. The control values of all parameters in each panel are l = 1 cm, N_n = 1 L^{-1}, TAS = 100 m s^{-1}, k = 5%, and n = 10.

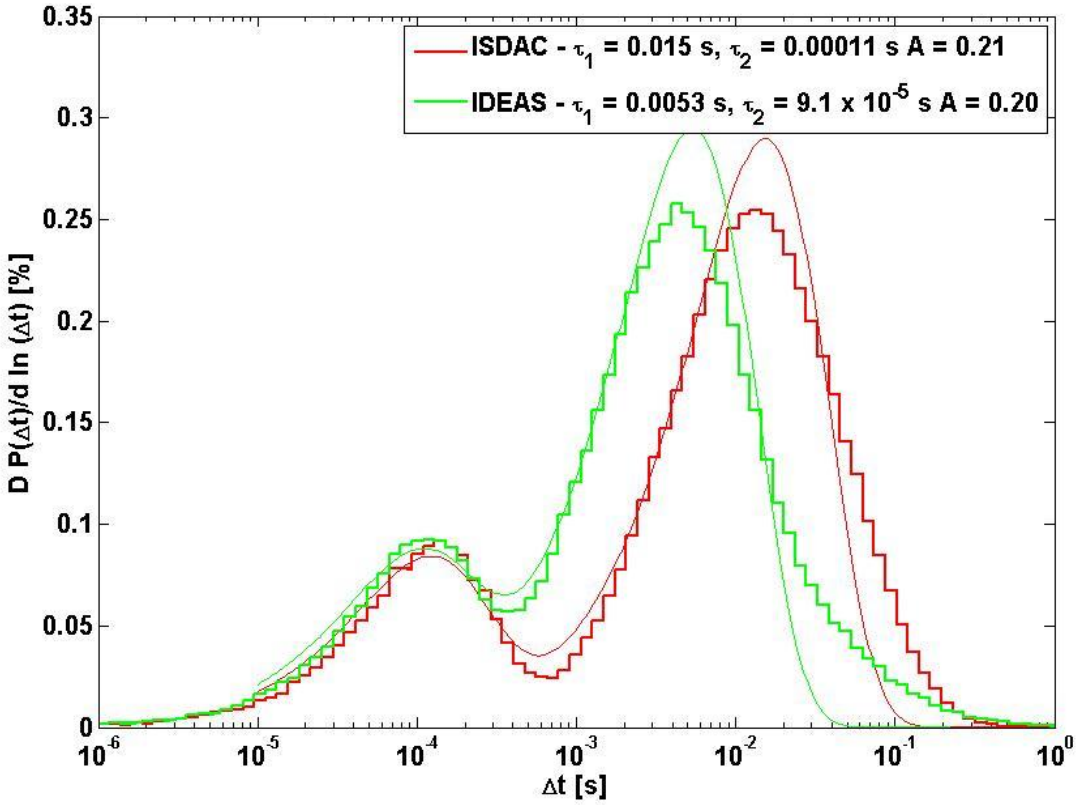


Figure 2.6. $\frac{dP(\Delta t)}{d \ln \Delta t}$ generated by running the Monte Carlo simulation for each 1 Hz value of *TAS*, and $N_{natural}$ from the 2DC number concentration after shattered artifact removal algorithms are applied and modified tips are used. Thin line is the expected $\frac{dP(\Delta t)}{d \ln \Delta t}$ derived from Poisson statistics (fit of observations to Equation (2)) for the given τ_1, τ_2, and A. For ISDAC, 32 photodiodes were assumed, and 64 for IDEAS-2011. For these simulations, k = 0.5%, and n = 10.

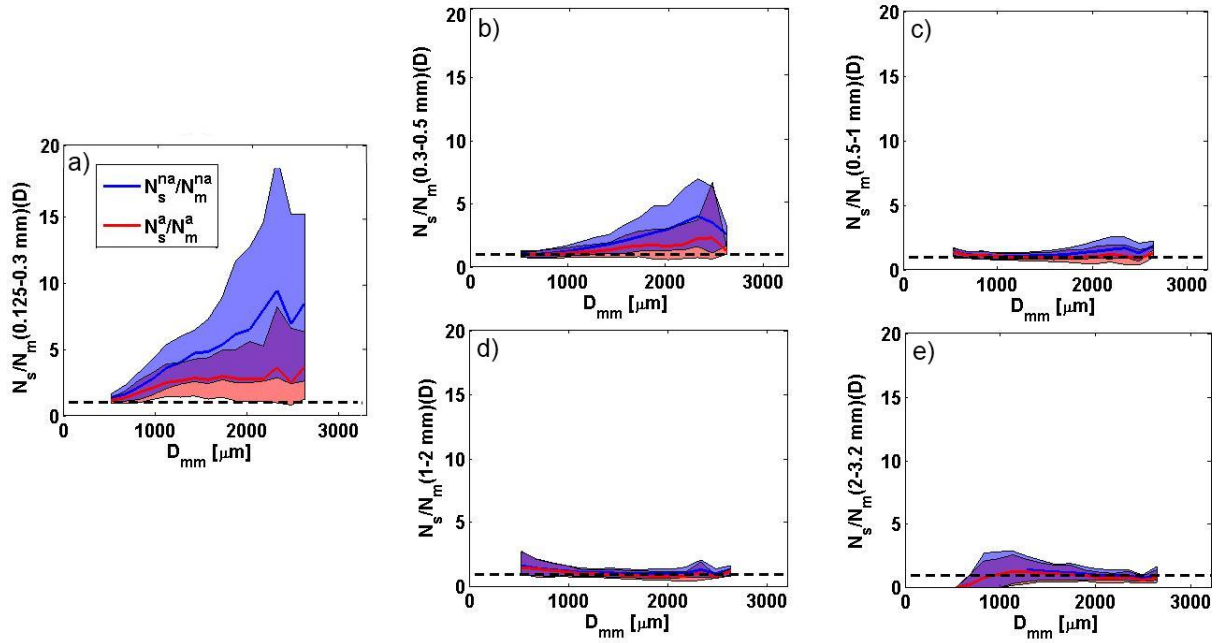


Figure 2.7. N_s^{na}/N_m^{na} and N_s^{a}/N_m^{a} for **(a)** 125 < D < 300 µm, **(b)** 300 < D < 500 µm, **(c)** 500 < D < 1000 µm, **(d)** 1000 < D < 2000 µm and **(e)** D > 2000 µm as a function of D_{mm} for both sorties during IDEAS-2011. Solid lines denote means, width of shading denote 10th and 90th percentiles. Blue shading and lines denote N_s^{na}/N_m^{na}. Red shading and line denote N_s^{a}/N_m^{a}.

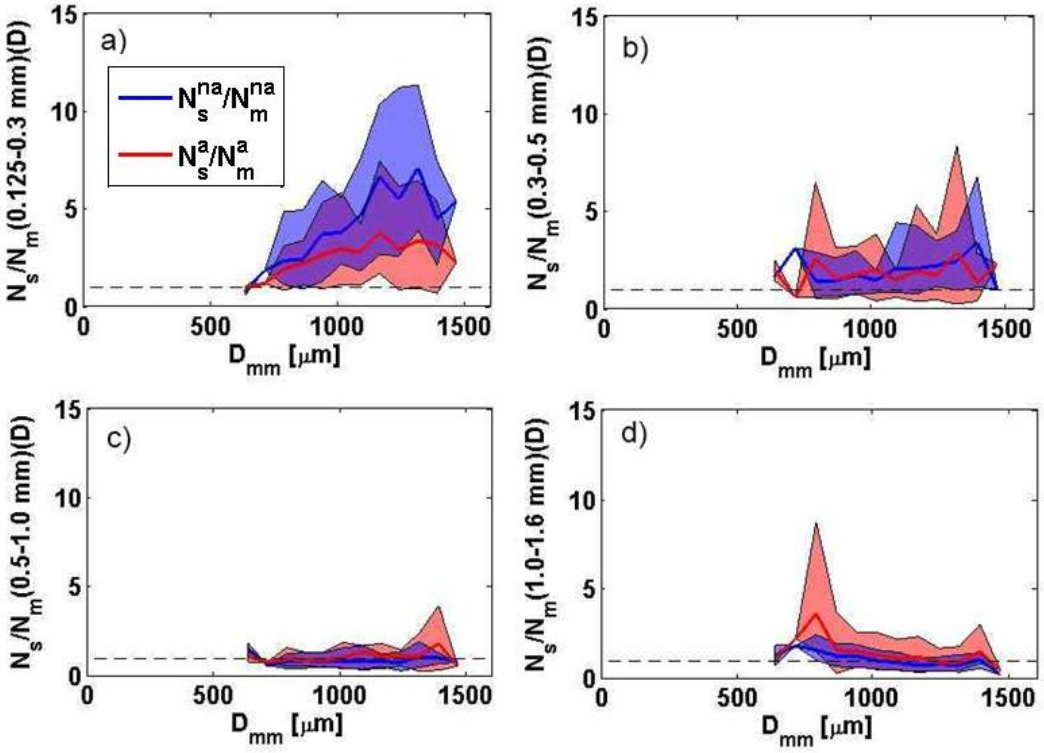


Figure 2.8. (a,b,c,d) As in Figure 2.7a, b, c, and d, but for ISDAC.

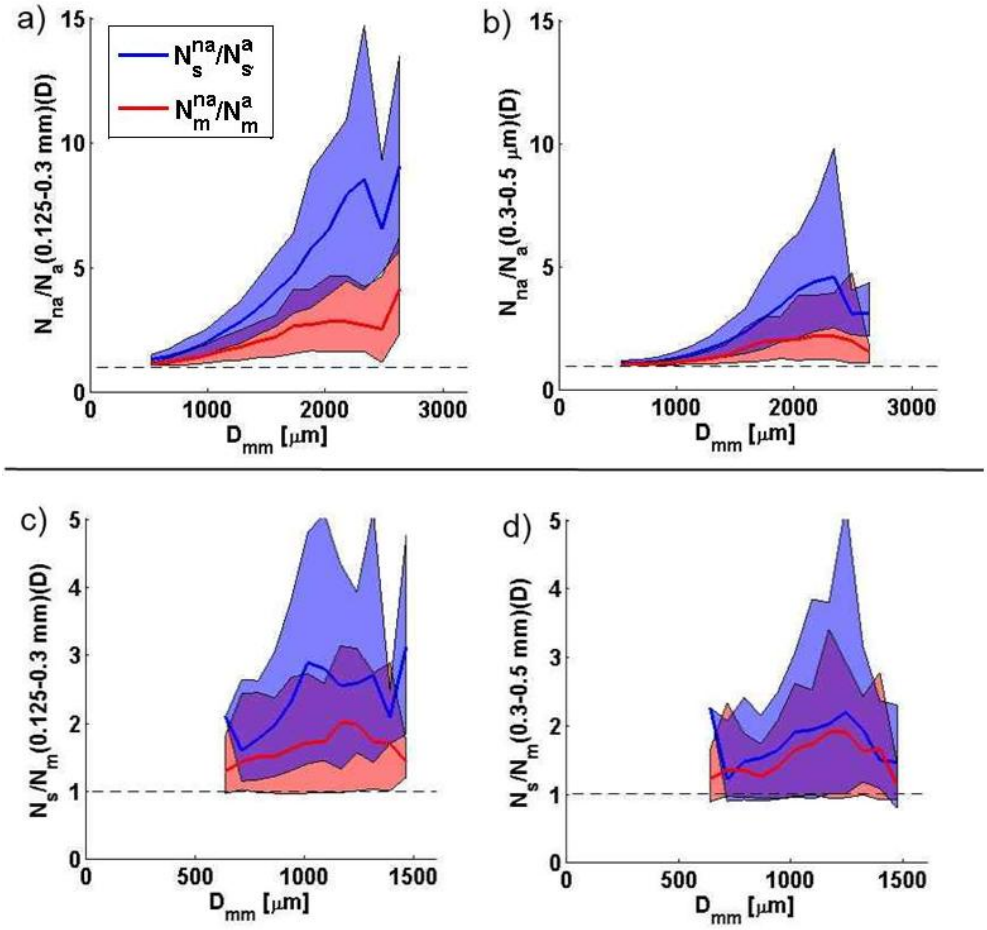


Figure 2.9. N_s^{na}/N_s^a and N_m^{na}/N_m^a for **(a)** 125 < D < 300 μm **(b)** 300 < D < 500 μm, as a function of D_{mm} for both sorties during IDEAS-2011. Solid lines denote means, width of shading denote the 10th and 90th percentiles. Blue shading and lines denote N_s^{na}/N_s^a. Red shading and line denote N_m^{na}/N_m^a. **(c,d)** as in **(a,b)** but for ISDAC.

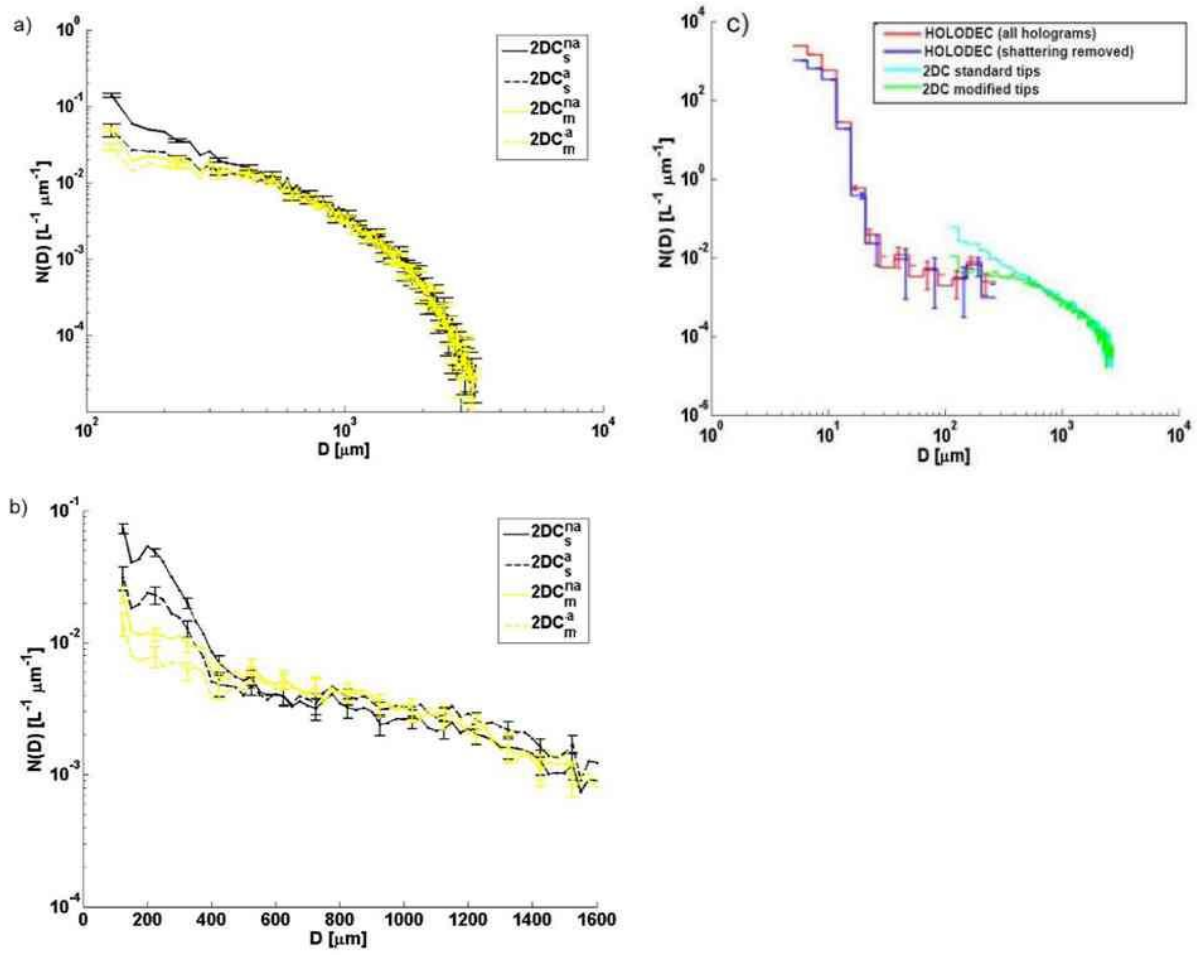


Figure 2.10. (a) Mean *N(D)* from the 2DC with standard and modified tips 25 October 2011 and 1 November 11. **(b)** Mean *N(D)* from the 2DC with standard and modified tips for 30 April 2008. Errorbars indicate standard deviations produced by the bootstrap technique. (**c)** Size distributions for IDEAS-2011 RF03, 19:21:30 through 19:26:41 comparing the HOLODEC and standard and modified 2DC instruments.

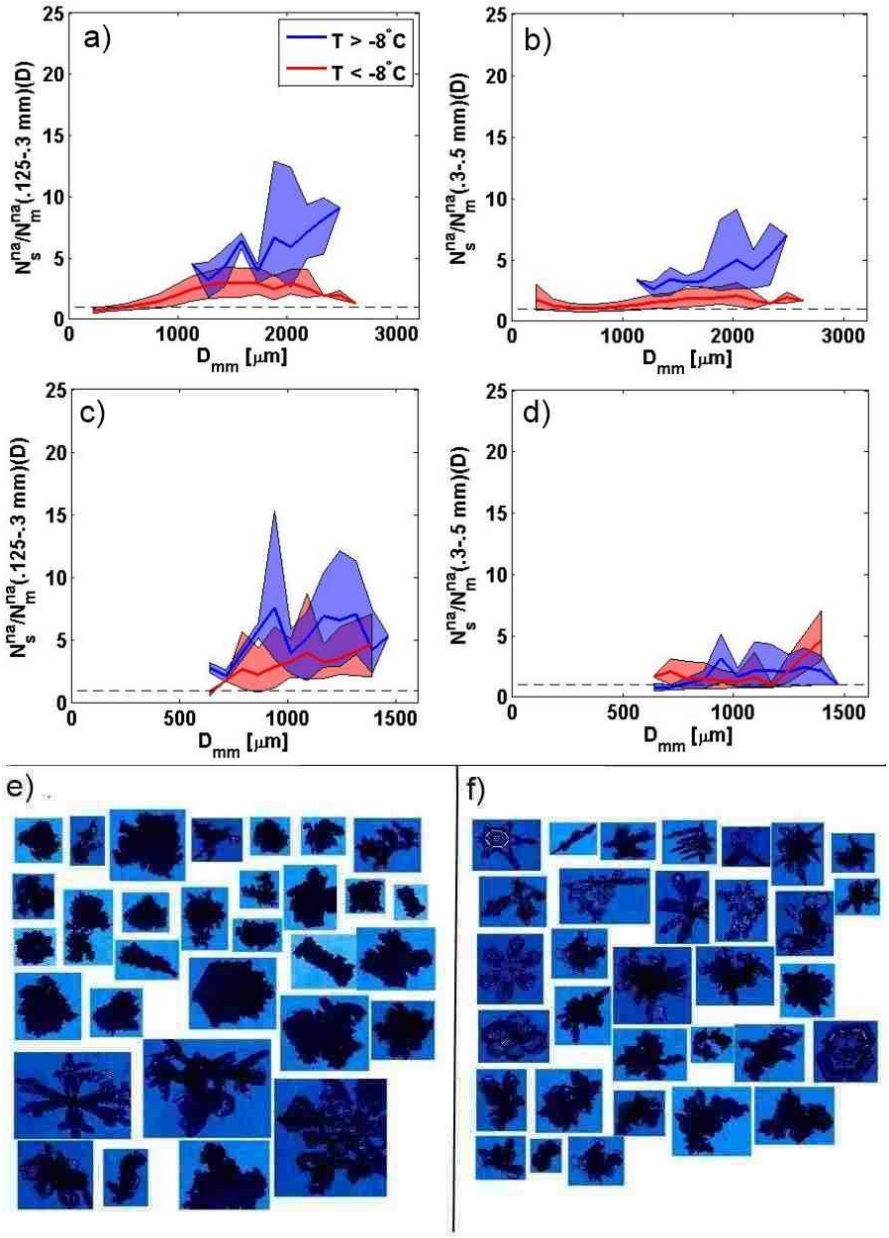


Figure 2.11. N_s^{na}/N_m^{na} for 125 μm < *D* < 300 μm **(a)** and 300 μm < *D* < 500 μm **(b)** as a function of D_{mm} for both sorties during IDEAS-2011. Solid lines denote means, width of shading 10th and 90th percentiles. **(c)** as in **(a)**, but for ISDAC. **(d)** as **(b)**, but for ISDAC. No artifact removal algorithms applied. Red shading and lines calculated using time periods when T > -8°C, and red lines calculated using time periods when T < -8°C. (**e)** Representative CPI images of particles of D > 1 mm on 1 November 2011 when T > -8°C. **(f)** As **(e)**, but for T < -8°C.

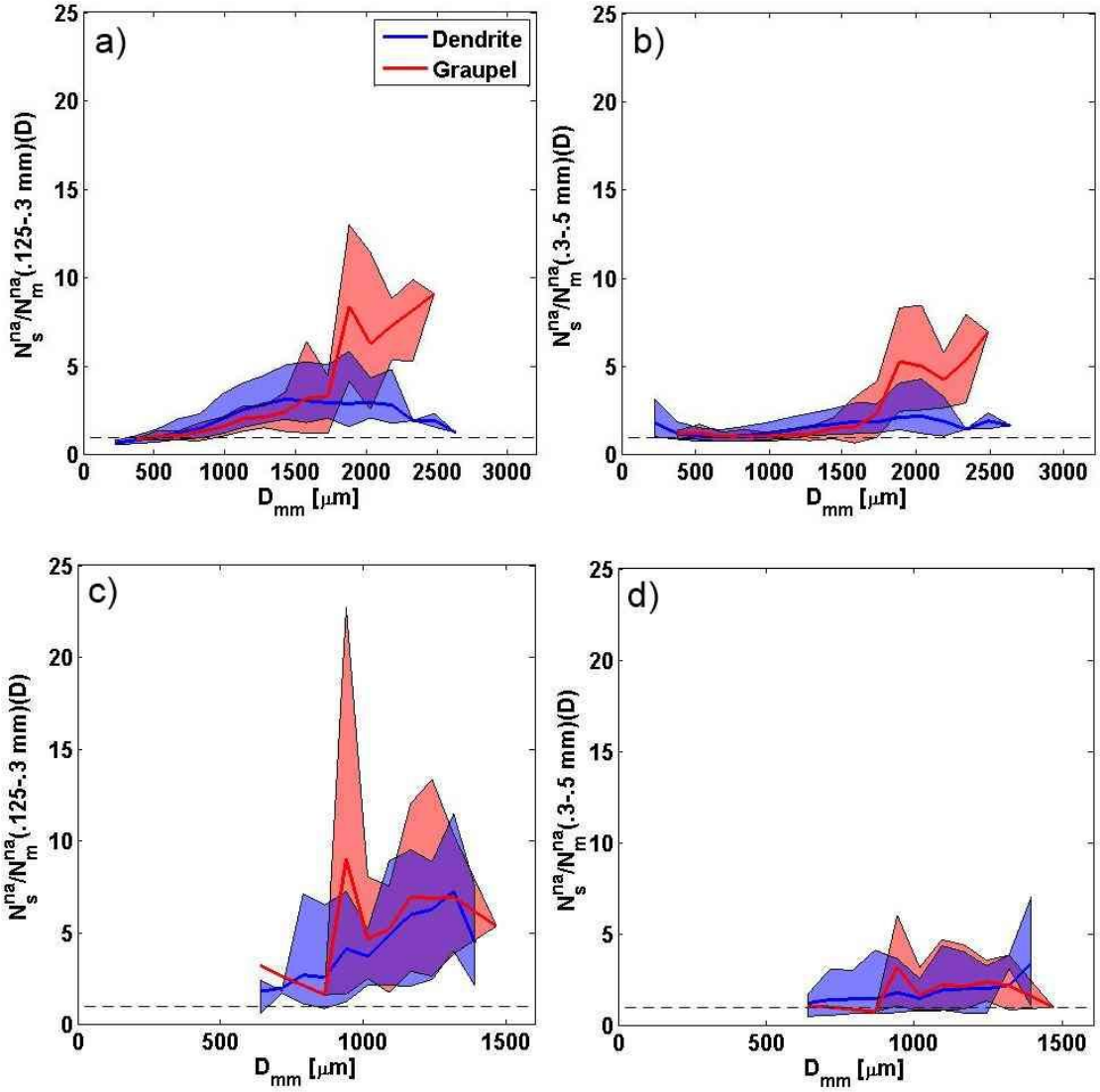


Figure 2.12. N_s^{na}/N_m^{na} for 125 μm < D < 300 μm **(a)** and 300 μm < D < 500 μm **(b)** as a function of D_{mm} for both sorties during IDEAS-2011. Solid lines denote means, width of shading denote 10th and 90th percentiles. No artifact removal algorithms applied. Blue shading and lines calculated using time periods when particles of D > 500 μm were identified as dendrites and red shading and lines when particles of D > 500 μm were identified as graupel. **(c)** as in **(a)** but for ISDAC. **(d)** as in **(b)** but for ISDAC.

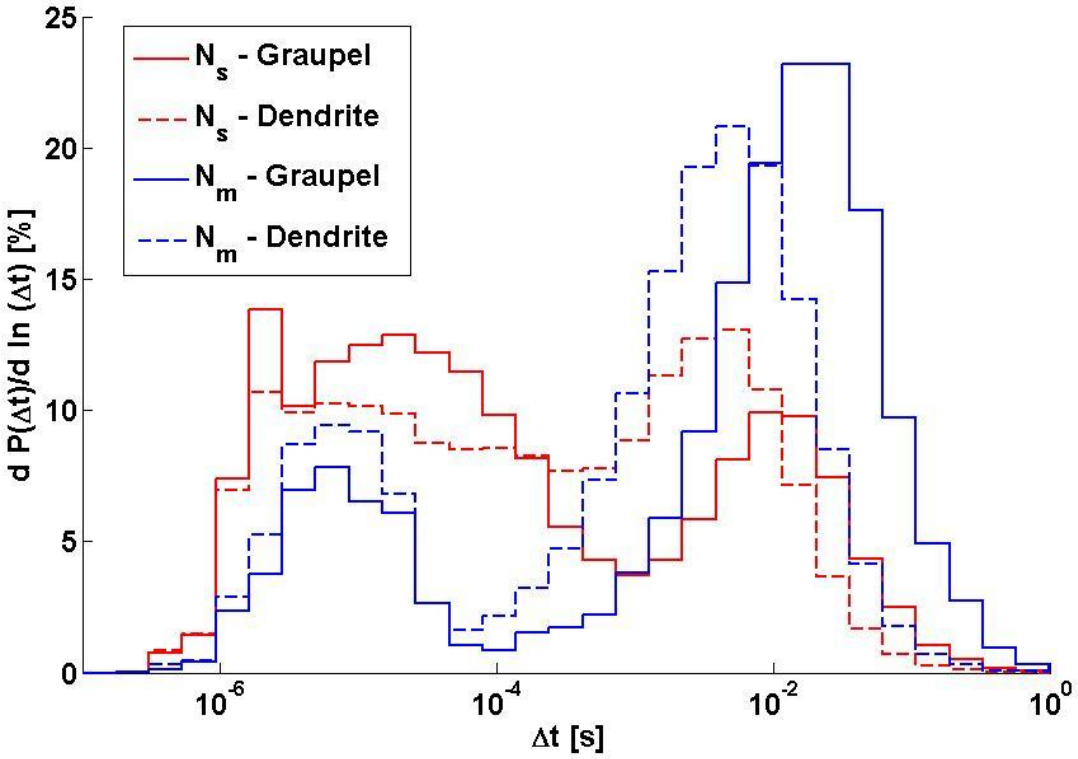


Figure 2.13. $\frac{dP(\Delta t)}{d \ln \Delta t}$ from the 2DC with standard tips (N_s) and modified tips (N_m) during IDEAS-2011 for time periods when D_{mm} > 1.6 mm. Solid lines calculated using time periods when particles of D > 500 μm were identified as graupel and dashed lines when particles of D > 500 μm were identified as dendrites.

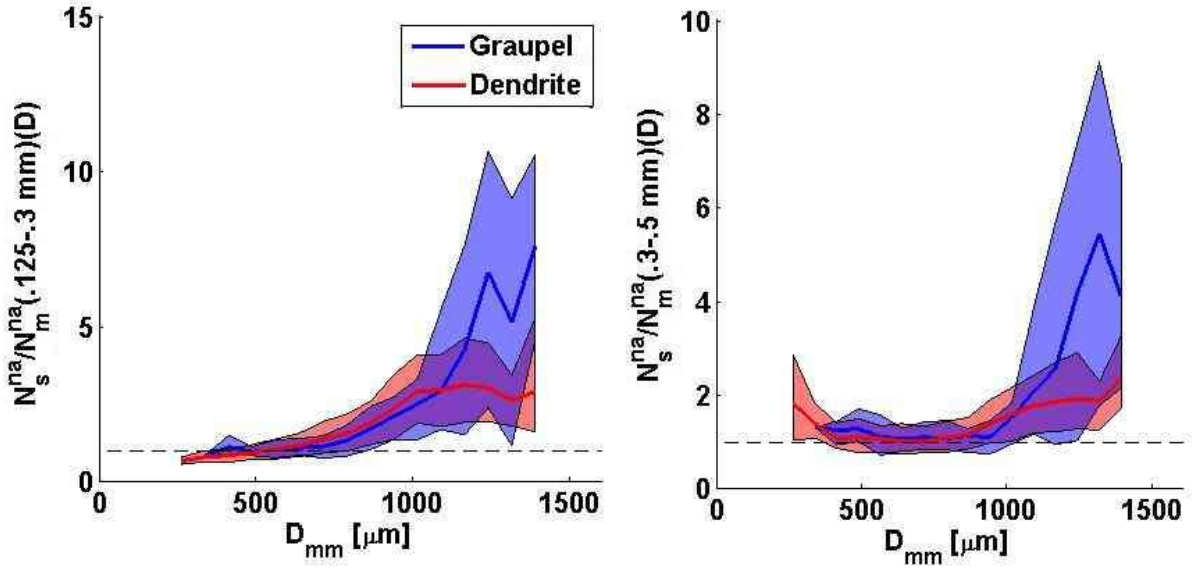


Figure 2.14. As Figure 2.12 **(a,b)** except only particles 0.125 mm < D < 1.6 mm are considered in calculating D_{mm}.

CHAPTER 3

AN ASSESSMENT OF THE IMPACT OF ANTI-SHATTERING TIPS AND ARTIFACT REMOVAL TECHNIQUES ON BULK CLOUD ICE MICROPHYSICAL AND OPTICAL PROPERTIES MEASURED BY THE 2D CLOUD PROBE.

Copyright Notice

This chapter is a paper in the *Journal of Atmospheric and Oceanic Technology* which is a journal of the American Meteorological Society (Jackson and McFarquhar 2014). The differences in this chapter from Jackson and McFarquhar (2014) only include changes to the formatting of the tables and figures. Jackson and McFarquhar (2014) is © Copyright 2014 American Meteorological Society (AMS). Permission to use figures, tables, and brief excerpts from this work in scientific and educational works is hereby granted provided that the source is acknowledged. Any use of material in this work that is determined to be "fair use" under Section 107 of the U.S. Copyright Act September 2010 Page 2 or that satisfies the conditions specified in Section 108 of the U.S. Copyright Act (17 USC §108, as revised by P.L. 94-553) does not require the AMS's permission. Republication, systematic reproduction, posting in electronic form, such as on a web site or in a searchable database, or other uses of this material, except as exempted by the above statement, requires written permission or a license from the AMS. Additional details are provided in the AMS Copyright Policy, available on the AMS Web site located at (http://www.ametsoc.org/) or from the AMS at 617-227-2425 or copyrights@ametsoc.org.

3.1. Introduction

In situ aircraft observations of the ice crystal number distribution function *N(D)* as a function of maximum dimension *D* are conventionally derived from in situ aircraft probes. However, as shown by Gardiner and Hallett (1985), Gayet et al. (1996), Field et al. (2003, 2006), Korolev and Isaac (2005), Heymsfield (2007), McFarquhar et al. (2007b, 2011), Jensen et al. (2009), Zhao et al. (2011), Lawson (2011), Febvre et al. (2012), Korolev et al. (2011, 2013a, 2013b) and Jackson et al. (2014), the shattering of large ice crystals on the tips or inlets of such probes can cause significant overestimates in *N(D).* The impact of the shattering on *N(D)* is especially problematic for crystals with $D < 500$ μm and in estimates of the total number concentration, N_T. Thus, any parameterization or use of such data requiring accurate estimates of N_T may provide biased or misleading results. However, many schemes also make use of estimates of quantities that are derived from higher order moments of *N(D)*, and the degree to which these quantities are biased by shattering is not well established. Examples of such quantities include the ice water content *IWC*, ice crystal effective radius r_e, visible extinction β, mass weighted terminal velocity v_m, and median diameter weighed by number (D_m) and mass (D_{mm}). Wavelength dependent properties such as single scatter albedo ω_o and asymmetry parameter g may also be biased by shattering. All of these properties can be derived or estimated from in situ aircraft observations of *N(D)*.

These quantities are necessary for the development of model parameterizations of sedimentation and radiative effects, development and evaluation of remote sensing retrieval schemes of bulk microphysical parameters, and for understanding processes occurring in ice and mixed phase clouds. For example, ω_o, g used in satellite retrieval algorithms are parameterized in terms of r_e derived from in situ observations (Baum et al. 2005a; 2005b; 2007; 2011). In situ estimates of *IWC*, D_{mm}, and v_m are used in the evaluation of Doppler radar retrievals of ice clouds

(i.e. Dong and Mace 2006) and to determine ice mass flux. The representation of scattering and absorption of solar and infrared radiation used in general circulation models depends on knowledge of β, ω_o, and g as a function r_e (i.e. Fu 1996), which is again based on measurements of *N(D).* Table 3.1 summarizes a list of applications for which estimates of bulk parameters based on in situ observations have been used, and provides references for previous studies using bulk properties derived from in situ measurements of *N(D)*.

Uncertainties in *N(D)* can translate into uncertainties in representations of sedimentation and single scattering properties in models which in turn affects model evolution. For example, Mitchell et al. (2008)'s global model simulations showed that increasing the number of ice crystals with $D < 150$ μm induced an upper tropospheric warming of 3 K and a total cloud forcing of -5 W m^2 in the Tropics through an increase of cirrus cloud coverage compared to a control simulation. On the other hand, Boudala et al.'s (2007) simulation including contributions of crystals with $D < 150\ \mu m$ had a net radiative forcing of 2.4 W m^{-2} greater than that of a simulation where such contributions were excluded. Further, McFarquhar et al. (2003) showed that decreasing r_e in a parameterization scheme affected the vertical profile of radiative forcing which in turn impacted low cloud cover, changing the shortwave radiative forcing by up to 25 W m^{-2}. Similar studies need to be conducted where the assumed uncertainties in bulk parameters are more closely tied to uncertainties associated with observations.

The impact of ice crystal shattering on *IWC,* β, v_m, r_e, D_m, D_{mm}, ω_0, g is investigated in this paper. Following Korolev et al. (2013b) and Jackson et al. (2014), the impact of two different methodologies used to reduce the role of shattered artifacts on the calculated bulk parameters is examined:

1) Algorithms based on the time between which particles are detected in the sample volume (Field et al. 2003; 2006) and based on the numbers, sizes and gaps between fragments in a single image recorded by probe (Korolev and Isaac 2005) that identify shattered artifacts;
2) 2DC tips modified to deflect most artifacts generated by the shattering of large ice crystals away from the sample volume (Korolev and Isaac 2005; Korolev et al. 2011; Lawson 2011).

Although previous studies have investigated the relative importance of algorithms and modified tips on *N(D)* and N_T, they have not thoroughly examined the effects of algorithms or modified tips on the derivation of v_m, r_e, D_m, D_{mm}, ω_0, and g. In this study, data from flights in which 2DCs with both standard and modified tips were installed on the same aircraft are used to determine the contributions of particles identified as shattered artifacts to the *IWC,* β, v_m, r_e, D_m, D_{mm}, ω_0, and g derived from 2DC data. This is investigated using measurements of particles with 125 μm < D < 3.2 mm acquired by 2DCs installed on National Research Council (NRC) Convair-580 during ISDAC and the National Science Foundation (NSF)/National Center for Atmospheric Research (NCAR) C-130 during IDEAS-2011. As described by Jackson et al. (2014), three sorties were conducted in two deep precipitating storms during IDEAS-2011, and in a multi-layer mixed phase stratocumulus during ISDAC. Further details on the meteorology and aircraft maneuvers used are given in Section 2.3.2 of this dissertation.

The remainder of the chapter is organized as follows. Section 3.2 highlights the instrumentation used and describes how *IWC,* β, v_m, r_e, D_m, D_{mm}, ω_0, and g were derived from *N(D)*. The effect of the modified tips and shattering removal algorithms on the derivation of

IWC, β, v_m, r_e, D_m, D_{mm}, ω_0, and g is discussed in Section 3.3. The principal findings of the study are summarized in Section 3.4.

3.2. Methodology

3.2.1. Instrumentation and data processing

Details of the instrumentation on the NRC Convair-580 and NSF/NCAR C-130 and the methods used to derive *N(D)* from the 2DC SDs with and without the use of shattered artifact removal algorithms are the same as in Chapter 2. Therefore, this section will only detail the calculation of the bulk parameters.

A number of different bulk parameters were computed from *N(D)*. The area ratio A_r is the cross sectional area of a particle divided by the area of a circle with the same *D* (McFarquhar and Heymsfield 1996). It was used in the calculation of β given by

$$\beta = \sum_j \sum_k Q_{ext}(D_{j,} A_{r_k}) A_p N(D_j, A_{r_k}) \Delta D_j \qquad (2)$$

where $Q_{ext}(D_{j,} A_{r_k})$ is the extinction efficiency, A_p is the cross sectional area of the particle, and $N(D_j, A_{r_k})$ the ice crystal number distribution function for bin *j* with maximum dimension midpoint D_j, width ΔD_j, and area ratio midpoint A_{r_k}. For particles with $D > 125$ μm at visible wavelengths, $Q_{ext}(D_{j,} A_{r_k}) \approx 2$ since geometric optics applies (Um and McFarquhar 2007). Thus, $\beta = 2A_c$, where A_c is the total cross sectional area estimated from the two-dimensional images of particles measured by the probe.

To estimate *IWC*, information about the relationship between crystal mass *m* and *D*, which depends on ice crystal habit, is required. Following Jackson et al. (2012), two different techniques were used for estimating *IWC*. One method, CPI-mD, derived $IWC_{CPI\text{-}mD}$ using

$$IWC_{CPI-mD} = \sum_j \sum_k f_k(D_j)\alpha_k D_j^{\beta_k} N(D_j)\Delta D_j \quad (3)$$

where $f_k(D_j)$ is the fraction of crystals in the bin centered at D_j having crystal habit k, α_k and β_k are habit-dependent coefficients listed in Table 3.2 that define the mass of an individual crystal $m = \alpha_k D_j^{\beta k}$,and $N(D_j)$ is the number distribution function for bin j with midpoint D_j and width ΔD_j. Only contributions from crystals with $A_r > 0.2$ were included. For 30 April 2008 and 1 November 2011 the 2.3 μm resolution CPI images were used to determine $f_k(D_j)$ using the habit classification scheme of Um and McFarquhar (2011). Because the CPI has a smaller sample volume than the 2DC or CIP, the averaging period of 60 s required to obtain a statistically significant sample was larger than the 10 s period required for the 2DC, and hence the habit distributions were applied to each of the 2DC/CIP size distributions occurring within the CPI averaging period. The 3V-CPI was not installed on the NSF/NCAR C-130 on 25 October 2011, so $f_k(D_j)$ were derived from the 2DC data using the Holroyd (1987) technique for that day.

A second method for computing *IWC*, denoted BL06, used the cross sectional mass-area relation of Baker and Lawson (2006) to determine the mass of each particle from the area measured by the optical array probes. If the mass calculated by this method was greater than that of an ice sphere with the same maximum dimension, the ice sphere mass was used instead. IWC_{BL06} is given by:

$$IWC_{BL06} = \sum_j \sum_k a\left(\frac{\pi}{4} D_j^2 A_{r_k}\right)^{1.218} N(D_j, A_{r_k})\,\Delta D_j \quad (4)$$

where $a = 0.115$ mg mm$^{-2.436}$.

D_m and D_{mm} are defined as the median number and median mass dimension respectively. Due to the uncertainty in the concentrations of particles with $D > 1.6$ mm during ISDAC from the use of the 32 photodiode 2DC, particles with $1.6 < D < 3.2$ mm were not considered in the

calculation of D_{mm} and *IWC* for ISDAC, but were for IDEAS-2011. Since particle mass scales approximately with the square of *D* (e.g., Brown and Francis 1995), D_{mm} and *IWC* were somewhat affected by the absence of large particles in the ISDAC calculations.

There are multiple definitions of r_e for distributions of ice crystals (McFarquhar and Heymsfield 1998). Fu (1996)'s definition of r_e given by

$$r_e = \frac{\sqrt{3}}{3\rho_i}\frac{IWC}{A_c} \tag{5}$$

was used here because the scattering properties of ice crystals are closely linked to the ratio of mass to cross sectional area (McFarquhar and Heymsfield 1998). Following McFarquhar and Black (2004) and others, the mass-weighted fall speed v_m was determined as

$$v_m = \frac{\sum_j \sum_k \gamma_k D_j^{\delta_k} \alpha_k D_j^{\beta_k} f_k(D_j) N(D_j) \Delta D_j}{\sum_j \sum_k \alpha_k D_j^{\beta_k} f_k(D_j) N(D_j) \Delta D_j} \tag{6}$$

where γ_k and δ_k are habit-dependent coefficients from empirical fits of ice crystal fall speed *v* to D_j as $v = \alpha_k D_j^{\beta k}$ from Loctaelli and Hobbs (1974) or Mitchell (1996).

The narrowband single-scatter albedo ω_0 and asymmetry parameter *g* were derived following McFarquhar et al. (2002) and Baum et al. (2005) as

$$\omega_0(\lambda_i) = \frac{\sum_j \sum_k \omega_{0ik}(\lambda_i, D_j) \sigma_{ik}(D_j) f_k(D_j) N(D_j) \Delta D_j}{\sum_j \sum_k \sigma_{ik}(D_j) f_k(D_j) N(D_j) \Delta D_j} \tag{7}$$

and

$$g(\lambda_i) = \frac{\sum_j \sum_k g_{ik}(\lambda_i, D_j) \sigma_{ik}(D_j) f_k(D_j) N(D_j) \Delta D_j}{\sum_j \sum_k \sigma_{ik}(D_j) f_k(D_j) N(D_j) \Delta D_j} \tag{8}$$

where $g_{ik}(\lambda_i, D_j)$ is the asymmetry parameter, $\omega_{0_{ik}}(\lambda_i, D_j)$ the single scatter albedo and $\sigma_{ik}(\lambda_{ik}, D_j)$ the scattering cross section of a particle with habit *k* and maximum dimension D_j at wavelength λ_i. Libraries of $g_k(\lambda_i, D_j)$, $\omega_{0_k}(\lambda_i, D_j)$ and $\sigma_k(\lambda_i, D_j)$ for nine idealized shapes (droxtals, hexagonal plate, hollow and solid columns, 4 and 6 bullet rosettes, and smooth and rough aggregates) from Yang et al. (2000; 2003) were used to calculate these properties for 56 solar wavelengths. For infrared wavelengths, $g_{ik}(\lambda_i, D_j)$, $\omega_{0_{ik}}(\lambda_i, D_j)$ and $\sigma_{ik}(\lambda_i, D_j)$ were taken from Yang et al. (2005)'s libraries for eight idealized shapes (aggregates, bullet rosettes, droxtals, hollow and solid columns, plate, and spheroid) and 49 wavenumbers from 100 cm^{-1} to 3250 cm^{-1}. The Yang et al. (2000; 2005) libraries did not always include the same habits used in the CPI and 2DC habit classification schemes. In this case, idealized habits whose shapes were closest to the morphological features of observed crystals were used. Table 3.2 lists the mapping between the CPI/Holroyd shapes and the corresponding Yang et al. (2000; 2005) shapes.

3.3. Contributions of shattered particles to bulk parameters

In this section, the differences in bulk microphysical parameters derived from 2DCs with standard and modified tips and processed with and without shattered artifact removal algorithms are compared. Data from both IDEAS-2011 and ISDAC are used. Subscripts *s* (*m*) are used to denote when standard (modified) tips were used, and superscripts *a* (*na*) denotes when algorithms were (not) used to derive β, *IWC*, v_m, r_e, D_m, and D_{mm} from the 2DC probes. For example, β_s^a is the visible extinction from the 2DC with standard tips calculated using shattered artifact removal algorithms. A list of symbols, subscripts, and superscripts used is provided in Table 3.3.

Figure 3.1 shows the mean cumulative area distribution from the 2DC $A_s^a(D)$, $A_s^{na}(D)$, $A_m^a(D)$, and $A_m^{na}(D)$ for ISDAC and IDEAS-2011. Jackson et al. (2014) showed that on average shattered artifacts contributed 71% to $N(D)$ for particles with $125 < D < 300$ µm and 63% for particles $300 < D < 500$ µm during ISDAC and IDEAS-2011 and that their contribution was higher in the presence of graupel. In Figs. 3.1a, and 3.1b, $A_s^{na}(D = 500\ \mu m)$ is 0.1 mm^2 L^{-1} (0.05 mm^2 L^{-1}) higher than $A_s^a(D = 500\ \mu m)$ for ISDAC (IDEAS-2011). Even though this means shattered artifacts contributed at least 15% (5%) to the cumulative area distribution at 500 μm, the fractional amount shattered artifacts contributed to A_c is much less than to N_T. In fact, particles with $D < 500\ \mu m$ contributed only 25% (10%) to A_c whereas they contributed 79% (59%) to N_T during ISDAC (IDEAS-2011). In the remainder of this section, it is shown that shattered artifacts make smaller contributions to bulk cloud and single scattering properties that are dominated by the higher moments.

It should also be noted that $A_s^a(D) < A_m^a(D)$ for $D > 1.2$ mm in Fig. 3.1b. At first, this seems counterintuitive. But, the sampling of different particle populations by the two distinct probes could easily cause this difference. For a 10 second average, the error due to statistical sampling, proportional to the square root of the actual number of particles measured, was 15.9% on average compared to the 7.5% difference between $A_s^a(D = 3.2\ mm)$ and $A_m^a(D = 3.2\ mm)$ in Figure 3.1a. For particles with $D < 500\ \mu m$, $A_s^a(D = 500\ \mu m)$ is 0.05 mm^2 L^{-1} higher than $A_m^{na}(D = 500\ \mu m)$ for ISDAC and equal to $A_m^{na}(D = 500\ \mu m)$ for IDEAS-2011, which is consistent with the modified tips removing at least as many particles as the algorithms. It is also possible that the accidental removal of large, real particles by the shattered artifact removal algorithm is causing this difference.

Figure 3.2 shows β_s^{a}, β_s^{na}, and β_m^{na} as functions of β_m^{a}. There is a strong (R = 0.99) correlation between all variables. β_s^{na} is 1.14 times higher than β_m^{na}, while β_s^{na} is 1.27 times higher than β_s^{a} and β_s^{na} is 1.22 times higher than β_m^{a}. These differences indicate that β was lower on average when derived from SDs measured by the standard tips and processed using the shatter detecting algorithms than for SDs measured by the probe with modified tips. However, as explained above, this result is not inconsistent with Jackson et al. (2014)'s claim that the tips are more effective than the algorithms at removing artifacts because the contribution of the artifacts to β are smaller than the differences associated with statistical samples of particles from the same population.

Figure 3.3a shows scatter plots of IWC_s^{a}, IWC_s^{na}, and IWC_m^{na}, as functions of IWC_m^{a} derived using the 'BL06' method. There is a strong (R = 0.99) correlation between all variables. IWC_s^{na} is 25.7% higher than IWC_s^{a} while IWC_s^{na} is 1.12 times higher than IWC_m^{na}. IWC_s^{na} is 1.22 times higher than IWC_m^{a}. As for the trends in β noted in Fig. 3.2, the comparisons of *IWC* at first glance seem surprising in that *IWC* is lower for SDs measured by the standard probes with the artifact algorithms applied than for the SDs measured by the modified probes without algorithms. However, as explained above, the differences between *IWCs* is mainly associated with varying numbers of large crystals with D > 1.2 mm measured by the two probes since the concentrations of these large particles have the greatest uncertainty due to counting statistics (McFarquhar et al. 2007c). Since crystals with D > 500 μm are not likely shattered artifacts, the trends in Fig. 3.3a are not inconsistent with Jackson et al (2014)'s finding that modified tips are more effective than algorithms at removing shattered artifacts.

Another uncertainty in calculating *IWC* is the poorly known relationship between particle mass and D. Figure 3.3b shows a scatter plot of IWC_m^{a} derived using the 'CPI-mD' method as a

function of IWC_m^a derived using the 'BL06' method. IWC_s^a derived using the 'CPI-mD' method as a function of IWC_s^a derived using the 'BL06' method is also shown. The slope of the best fit of IWC_m^a derived using the 'CPI-mD' method to IWC_m^a derived using the 'BL06' method is 0.47 and the R is only 0.78. Thus, there is a greater amount of scatter and more discrepancy in the values in Fig. 3.3b than in Fig. 3.3a. Thus, the contribution of particles identified as shattered artifacts to *IWC* is less than the uncertainty in *IWC* associated with the use of varying *m*-D relations for the conditions sampled during ISDAC and IDEAS-2011, showing shattered artifacts are not the largest source of uncertainty in determining *IWC*.

Figure 3.4 shows scatter plots of $D_{m\text{-}m}^{na}$, $D_{m\text{-}s}^{na}$, and $D_{m\text{-}s}^{a}$ as functions of $D_{m\text{-}m}^{a}$. On average, $D_{m\text{-}s}^{na}$ is 53% of $D_{m\text{-}m}^{na}$. Meanwhile, $D_{m\text{-}s}^{na}$ is 71% of $D_{m\text{-}s}^{a}$, and $D_{m\text{-}s}^{na}$ is 43% of $D_{m\text{-}m,}^{a}$ on average. This shows that the use of modified tips changes D_m more than the use of shattered artifact removal algorithms. The lack of correlation (R = 0.16) and up to a factor of 5 difference between $D_{m\text{-}s}^{a}$ and $D_{m\text{-}m}^{a}$ in Fig. 3.4 show that it is critical that both modified tips and shattered artifact algorithms are used when deriving D_m from 2DCs. Thus, parameterizations of D_m derived from 2DC data in ice phase conditions such as those of Kristjansson et al. (2000) and Ivanova et al. (2001) may need to be revised using data from 2DCs with modified tips and shattered artifact removal algorithms. It is not surprising that D_m is more sensitive to shattered artifacts and less sensitive to larger particles than β or *IWC* as it is weighted more towards lower moments of *N(D)* and consequently by smaller particles.

Figures 3.5a shows $D_{mm\text{-}s}^{a}$, $D_{mm\text{-}s}^{na}$, and $D_{mm\text{-}m}^{na}$ as functions of $D_{mm\text{-}m}^{a}$, calculated using mass derived from the 'BL06' method. $D_{mm\text{-}s}^{na}$ is 99% of $D_{mm\text{-}s}^{a}$ while $D_{mm\text{-}s}^{na}$ is 98% of $D_{mm\text{-}m}^{na}$ on average. To test the sensitivity of D_{mm} to the *m-D* relationship, Figure 3.5b shows a scatter plot of $D_{mm\text{-}m}^{a}$ derived using the 'CPI-mD' method as a function of $D_{mm\text{-}m}^{a}$ derived using the

'BL06' method. The slopes of the best fit of $D_{mm\text{-}m}{}^{a}$ calculated using the CPI-mD method to $D_{mm\text{-}m}{}^{a}$ using the 'BL06' method of 0.75 is lower than 0.95 for the fit of $D_{mm\text{-}s}{}^{a}$ to $D_{mm\text{-}m}{}^{a}$ shown in Fig. 3.5a. The R values of 0.83 for this best fit are lower than all of the *R* values of 0.89 or greater for the fits in Fig. 3.5a, indicating a greater amount of scatter in Fig. 3.5b than in Fig. 3.5a. Therefore, changing mass calculation techniques impacts D_{mm} more than does applying shattered artifact algorithms or using modified tips. Thus, for conditions sampled during ISDAC and IDEAS-2011, the impact of shattering is not the largest uncertainty for the calculation of D_{mm}.

Figure 3.6a shows $\nu_{m\text{-}s}{}^{a}$, $\nu_{m\text{-}s}{}^{na}$, and $\nu_{m\text{-}m}{}^{na}$ as functions of $\nu_{m\text{-}m}{}^{a}$, calculated using mass derived from the 'BL06' method. $\nu_{m\text{-}s}{}^{na}$ is 96% of $\nu_{m\text{-}s}{}^{a}$ while $\nu_{m\text{-}s}{}^{na}$ is 94% of $\nu_{m\text{-}m}{}^{na}$ on average. To test the sensitivity of ν_m on the to the *m-D* relationship, Figure 3.6b shows a scatter plot of $\nu_{m\text{-}m}{}^{a}$ derived using the 'CPI-mD' method as a function of $\nu_{m\text{-}m}{}^{a}$ derived using the 'BL06' method. The $\nu_{m\text{-}m}{}^{a}$ derived using the 'BL06' method is 97% of $\nu_{m\text{-}m}{}^{a}$ derived using the 'CPI-mD' method on average. The $\nu_{m\text{-}s}{}^{a}$ derived using the 'BL06' method is 100% of $\nu_{m\text{-}s}{}^{a}$ derived using the 'CPI-mD' method on average. Therefore the use of modified tips, algorithms, and a different *m-D* relationship all change ν_m by similar amounts for the ISDAC and IDEAS-2011 data, with none of the changes invoking more than a 5% difference in ν_m on average.

Figure 3.7a shows $r_{e\text{-}s}{}^{a}$, $r_{e\text{-}s}{}^{na}$, and $r_{e\text{-}m}{}^{na}$, as functions of $r_{e\text{-}m}{}^{a}$, calculated using mass derived from the 'BL06' method. $r_{e\text{-}s}{}^{na}$ is 98% of $r_{e\text{-}s}{}^{a}$ and $r_{e\text{-}s}{}^{na}$ is 97% of $r_{e\text{-}m}{}^{na}$ on average. Therefore, modified tips and algorithms change r_e by less than 5% for the ISDAC and IDEAS-2011 data. Figure 3.7b shows a scatter plot of $r_{e\text{-}m}{}^{a}$ derived using the 'CPI-mD' method as a function of $r_{e\text{-}m}{}^{a}$ derived using the 'BL06' method. The $r_{e\text{-}m}{}^{a}$ derived using the 'BL06' method is 74% of $r_{e\text{-}m}{}^{a}$ derived using the 'CPI-mD' method on average. In addition, *R* for the best fit of $r_{e\text{-}}$

$_m{}^a$ derived using the 'BL06' method to $r_{e\text{-}m}{}^a$ derived using the 'CPI-mD' method is equal to 0.10 suggesting there is essentially no correlation between the two calculations of $r_{e\text{-}m}{}^a$. Because r_e is proportional to IWC/β, the scatter in Fig. 3.7b is related to the scatter in *IWC* in Fig. 3.3b caused by uncertainties in the conversion of two-dimensional particle images into three-dimension volumes and mass. Therefore, changing mass calculation techniques has more of an impact on r_e than does the application of shattered artifact algorithms or use of modified tips. Finally, it should be noted that although no true values of D_m, D_{mm}, r_e, or v_m are known because shattered artifacts may still make contributions even with the use of algorithms and modified tips, the preceding analysis gives some indication of the expected uncertainties on these quantities due to shattering.

Figures 3.8 and 3.9 show the mean and standard deviation of $\omega_{0s}{}^{na}$, $\omega_{0s}{}^{na}$, $\omega_{0m}{}^a$, $\omega_{0m}{}^{na}$, $g_s{}^a$, $g_s{}^{na}$, $g_m{}^a$, and $g_m{}^{na}$ as functions of wavelength and wavenumber. Figures 3.8a and 3.9a show that $\omega_{0s}{}^a$, $\omega_{0s}{}^{na}$, $\omega_{0m}{}^{na}$, are all greater than $\omega_{0m}{}^a$ by at most 0.005. In addition, $g_s{}^a$, $g_s{}^{na}$, and $g_m{}^a$ are all greater than $g_m{}^a$ by at most 0.01 in Figure 3.8b and less than $g_m{}^a$ by at most 0.01 in Figure 3.9b. In Eq. (7) and Eq. (8), ω_0 and g are weighted by the scattering cross section which is related to the particle cross sectional area. Therefore, ω_0 and g are weighted towards higher order moments of $N(D)$ and larger particles, so it is not surprising that shattered artifacts contribute fractionally less to ω_0 and g, on the order of 1%, than they do to number concentration and D_m. These differences are much less than the uncertainties associated with dominant crystal habits or choice of size distribution (Macke et al. 1998; McFarquhar et al. 1999; Um and McFarquhar 2007) or presence of surface roughness (Yang et al. 2013). Further, the uncertainties in these single scattering quantities derived for the cloud conditions sampled on these 3 sorties during ISDAC

and IDEAS-2011 are less than the certainty with which they need to be known from a climate perspective (Volgelmann and Ackermann 1995; McFarquhar et al. 2002).

3.4. Discussion and conclusions

In situ aircraft measurements of ice crystal number distribution function *N(D)* were acquired by a 2DC with standard tips immediately adjacent to a 2DC with tips modified to reduce the number of shattered fragments entering the sample volume on the NRC Convair-580 during ISDAC and on the NSF/NCAR C-130 during IDEAS-2011. Extending the study of Jackson et al. (2014), the impact of the tips and artifact removal algorithms (Korolev and Isaac 2005; Field et al 2006) on a number of bulk cloud properties such as bulk extinction β, ice water content *IWC*, asymmetry parameter g, single scatter albedo ω_0, median diameter D_m, median mass diameter D_{mm}, effective radius r_e, and mass weighted fall speed v_m, derived from the number distribution function *N(D)* was examined. The principal conclusions of this study are as follows:

1. The use of modified tips reduced β from 2DC data by 15% while the use of shattered artifact removal algorithms reduced β by 25%. Shattered artifacts contributed up to 15% to the total cross sectional area. The β derived from the 2DC with standard tips and use of shattered artifact removal algorithms was lower than that from modified tips with the use of shattered artifact removal algorithms most likely due to differences in statistical sampling of particles with $D > 1.2$ mm by the different probes.
2. *IWC*, r_e, D_{mm} changed by less than 20% when either shattered artifact removal algorithms or modified tips were used. These changes were less than the up to

60% change in *IWC*, r_e, and D_{mm} when changing the assumed relationship between mass and *D*. The large amount of scatter in the relationship between *IWC*, r_e, D_{mm} calculated from mass derived from habit-dependent *m-D* relationships versus those calculated from mass from the Baker and Lawson (2006) method shows that the error in *IWC* r_e, and D_{mm} induced by shattering is not the primary uncertainty in deriving these quantities.

3. $v_{m,}$ changed by less than 20% when either shattered artifact removal algorithms or modified tips were used and when changing the assumed relationship between mass and *D*.
4. D_m increased by an average of 43% when modified tips were used in place of standard tips. When only shattered artifact removal algorithms were used, the increase was 30%. There was also a large amount of scatter in the relationship between D_m from the standard tips and D_m from the modified tips whether or not algorithms were used, indicating that both the algorithms and modified tips must be used in deriving D_m for the conditions encountered during ISDAC and IDEAS-2011.
5. The use of modified tips and shattered artifact removal algorithms changed *g* and ω_0 by at most 0.01 and 0.005 respectively, less than how well they currently need to be known for climate parameterizations.

The comparisons made in this paper can be used as an estimate of the bias due to shattered artifacts in bulk parameters derived from 2DCs in the cloud conditions sampled during one sortie of ISDAC and two sorties of IDEAS-2011. Despite the differences in the electronics of the 2DCs used and the sampling conditions encountered during IDEAS-2011 and ISDAC,

there are common trends between both projects. Therefore, the determined differences between quantities calculated between probes with standard and modified tips, and processed with and without artifact removal algorithms provides some guidance on the bias that shattering induces on various bulk cloud properties. However, the results of this study cannot necessarily be generalized to determine the contribution of shattered artifact on bulk parameters any 2DC used in any cloud condition.

The results of this study provide guidance on interpreting biases in data collected prior to the development of the modified tips and processed before the development of shatter detecting algorithms. In general, biases induced by shattering on quantities dominated by higher order moments (e.g., β, *IWC*, D_{mm}, v_m) are less than those in quantities dominated by lower order moments (e.g. N and D_m). This implies that historical observations of and parameterizations using β, g, *IWC*, D_{mm}, v_m derived from particle size distributions with $N(D > 125\ \mu m)$ from 2DCs with standard tips might still be acceptable for use provided the differences noted here are less than the accuracy at which this quantities need to be known. However, historical observations and parameterizations of N and D_m derived from 2DCs with standard tips are likely to be contaminated by the large amounts of shattered artifacts indicated here and in Jackson et al. (2014), and therefore caution is advised in the use of these parameters. Future studies should concentrate on conducting similar comparisons with data acquired in ice clouds at colder temperatures (e.g. < -40°C), in a wider range of aircraft operating parameters (e.g. higher true air speeds), in a wider range of crystal habits (e.g. more rimed particles and graupel) with a wider variety of probes (e.g. 2D Stereo Probe), and with different designs of probe tips for deflecting shattered particles from the probe sample volume (e.g. Korolev et al. 2013a).

CHAPTER 3 FIGURES AND TABLES

Table 3.1. Examples of products developed using in situ aircraft observations of ice clouds.

Product	References
Parameterizations of $N(D)$	Heymsfield and Platt (1984), Ivanova et al. (2001), Boudala et al. (2002), Field and Heymsfield (2003), Field et al. (2007)
Parameterizations of r_e	Fu (1996), McFarquhar (2001), McFarquhar et al. (2003), Boudala and Isaac (2006), Liou et al. (2008), Schuman et al. (2011)
Parameterizations of $D_{(mm)}$	Delanoë et al. (2005)
Parameterizations of single scattering properties	Takano and Liou (1989); Ebert and Curry (1992); Fu and Liou (1993), Fu (1996); Kristjánsson et al. (2000), McFarquhar et al. (2002), Naisiri et al. (2002) Baum et al. (2005a, 2005b, 2007; 2010)
Empirical fits of mass m	Brown and Francis (1995), Heymsfield et. al. (2002b), Heymsfield et al. (2002a; 2002b), Heymsfield et al. (2004), Heymsfield (2007), Heymsfield et al. (2010)
Empirical fits of area a	Heymsfield and Miloshevich (2003), Lawson et al. (2010)

Table 3.1 (cont.)

Parameterizations of v_m	Heymsfield (2003a; 2003b), Heymsfield (2007), Schmitt and Heymsfield (2009), Mitchell et al. (2011)
Passive remote sensing retrieval techniques	Takano and Liou (1989), Baum et al. (2000), Rolland et al. (2000), King et al. (2004), Mitchell et al. (2010)
Active remote sensing retrieval techniques	Atlas et al. (1995), Donovan and van Lammeren (2001), Mace et al. (2002), Dong and Mace (2006), Hogan et al. (2006), Delanoë et al. (2007), Austin et al. (2009), Delanoë and Hogan (2008), Kulie and Bennartz (2009)
Evaluation of remote sensing retrievals	Francis et al. (1998), Matrosov et al. (2002), Wang and Sassen (2002a; 2002b), van Zadelhoff et al. (2007), Protat et al. (2007), Heymsfield et al. (2008), Pokharel and Vali (2011)

Table 3.2. List of *m-D, v-D* relationships, and assumed ice crystal model used in Equations (3, 5, 6, 7, 8).

Habit	m-D relationships	v-D relationship	Yang et al. (2000; 2003) shape model	Yang et al. (2005) shape model
Tiny	Brown and Francis (1995)	Locatelli and Hobbs (1974)	Droxtal	Droxtal
Oriented	Brown and Francis (1995)	Locatelli and Hobbs (1974)	Solid column	Solid column
Linear	Mitchell (1996) "Hexagonal columns"	Mitchell (1996) "Hexagonal columns"	Solid column	Solid column
Aggregate	Brown and Francis (1995)	Locatelli and Hobbs (1974)	Rough aggregate	Rough aggregate
Irregular	Brown and Francis (1995)	Locatelli and Hobbs (1974)	Rough aggregate	Rough aggregate
Graupel	Mitchell (1996) "Lump graupel"	Mitchell (1996) "Lump Graupel"	Droxtal	Spheroid
Dendrite	Mitchell (1996) "Broad branched crystal"	Mitchell (1996) "Broad branched crystal"	Dendrite	Aggregate
Plate	Mitchell (1996) "Hexagonal plate"	Mitchell (1996) "Hexagonal plate"	Plate	Plate

Table 3.3. List of symbols, superscripts, and subscripts used in Chapter 3

Maximum dimension	D
Number distribution function	N(D)
visible extinction	β
ice water content	IWC
ice crystal effective radius	r_e
mass weighted fall speed	v_m
single scatter albedo	ω_o
asymmetry parameter	g
median diameter weighed by number	D_m
median diameter weighed by mass	D_{mm}
Total cross sectional area	A_c
Area ratio	A_r
Extinction efficiency	Q_{ext}
Bulk density of ice	ρ_i
Interarrival time	Δt
Probability density function of Δt	$\frac{dP(\Delta t)}{d \ln \Delta t}$
Contribution of shattered artifacts to $\frac{dP(\Delta t)}{d \ln \Delta t}$	S
Empirical fit coefficients of *m, v to* D	$\alpha, \beta, \gamma, \delta$
standard tips used	s
modified tips used	m

Table 3.3. (cont.)

shattered artifact removal algorithms used	*a*
shattered artifact removal algorithms not used	*na*
Mass calculated using Equation (3)	*CPI-mD*
Mass calculated using Equation (4)	*BL06*

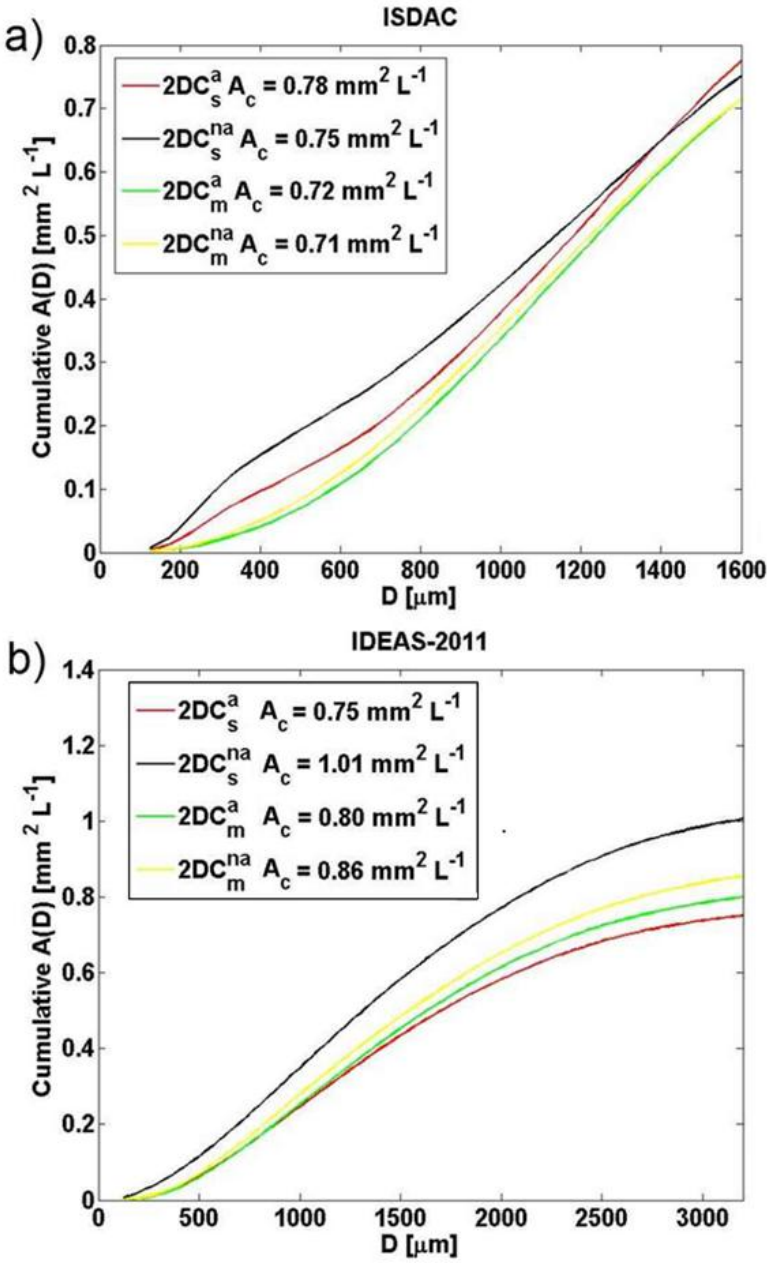


Figure 3.1. Mean cumulative area distribution *A(D)* from the 2DCs during **(a)** ISDAC and **(b)** IDEAS-2011. Subscripts *s* (*m*) denote the use of standard (modified) tips, and superscripts *a* (*na*) when algorithms were (not) applied as indicated in legend. Mean A_c distribution indicated in legend. $A_s^{na}(D = 500\ \mu m)$ is 0.1 mm^2 L^{-2} (0.05 mm^2 L^{-2}) higher than $A_s^{a}(D = 500\ \mu m)$ for ISDAC (IDEAS-2011).

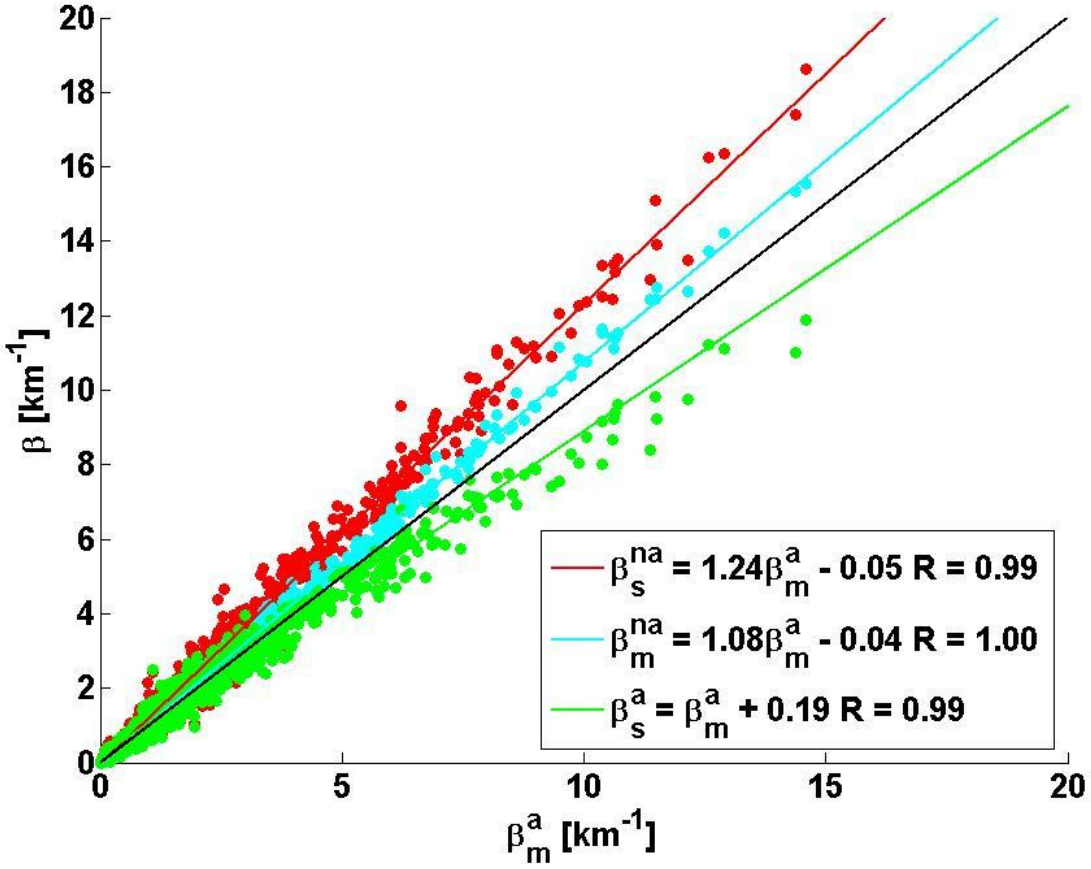


Figure 3.2. Scatter plot of β_s^{na}, β_s^{a}, and β_m^{na} as a function of β_m^{a}. Solid lines are best fit of quantity on *y* axis to quantity on *x* axis, with coefficients of fit and regression coefficient indicated in legend. Black line is 1:1. In general, $\beta_s^{na} > \beta_m^{na} > \beta_m^{a} > \beta_s^{a}$.

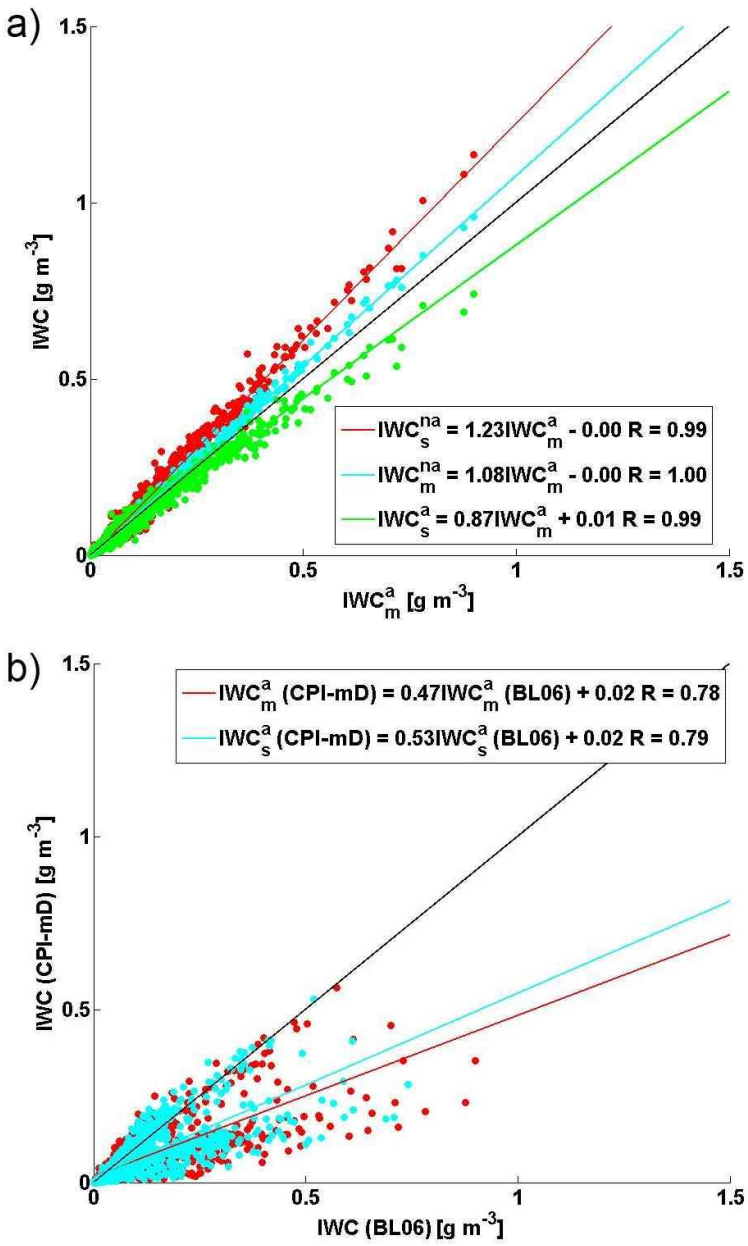


Figure 3.3. (a) As in Fig. 3.2, except for IWC_s^{na}, IWC_s^{a}, and IWC_m^{na} as a function of IWC_m^{a}. 'BL06' method was used to calculate *IWC*. **(b)** Scatter plot of IWC_m^{a} and IWC_s^{a} calculated using the 'CPI-mD' method as a function of IWC_m^{a} and IWC_s^{a} calculated using the 'BL06' method. Black line is 1:1. In general, $IWC_s^{na} > IWC_m^{na} > IWC_m^{a} > IWC_s^{a}$.

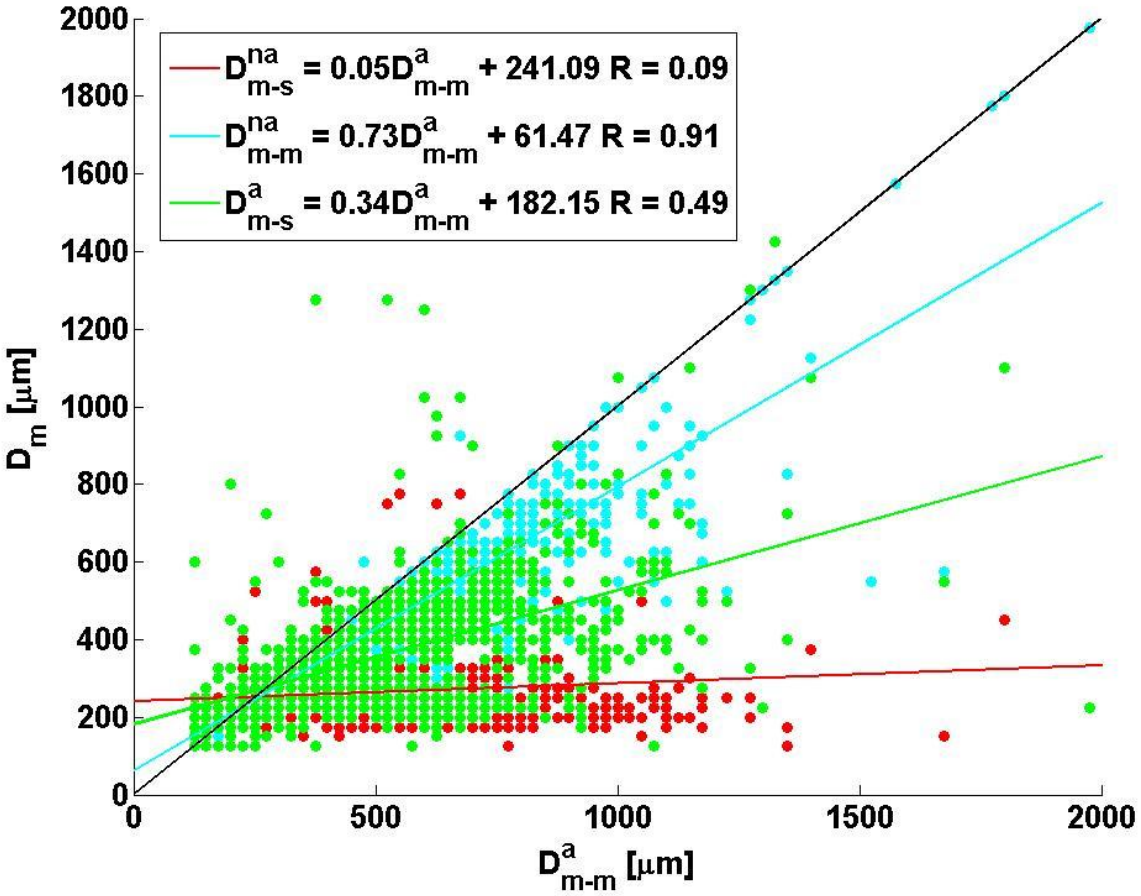


Figure 3.4. As in Fig. 3.2, except for $D_{m\text{-}m}^{\ na}$, $D_{m\text{-}s}^{\ a}$, and $D_{m\text{-}s}^{\ na}$ as a function of $D_{m\text{-}m}^{\ a}$. Solid lines are best fit of quantity on *y* axis to quantity on *x* axis. Black line is 1:1. In general, $D_{m\text{-}s}^{\ na} > D_{m\text{-}s}^{\ a} > D_{m\text{-}m}^{\ na} > D_{m\text{-}m}^{\ a}$.

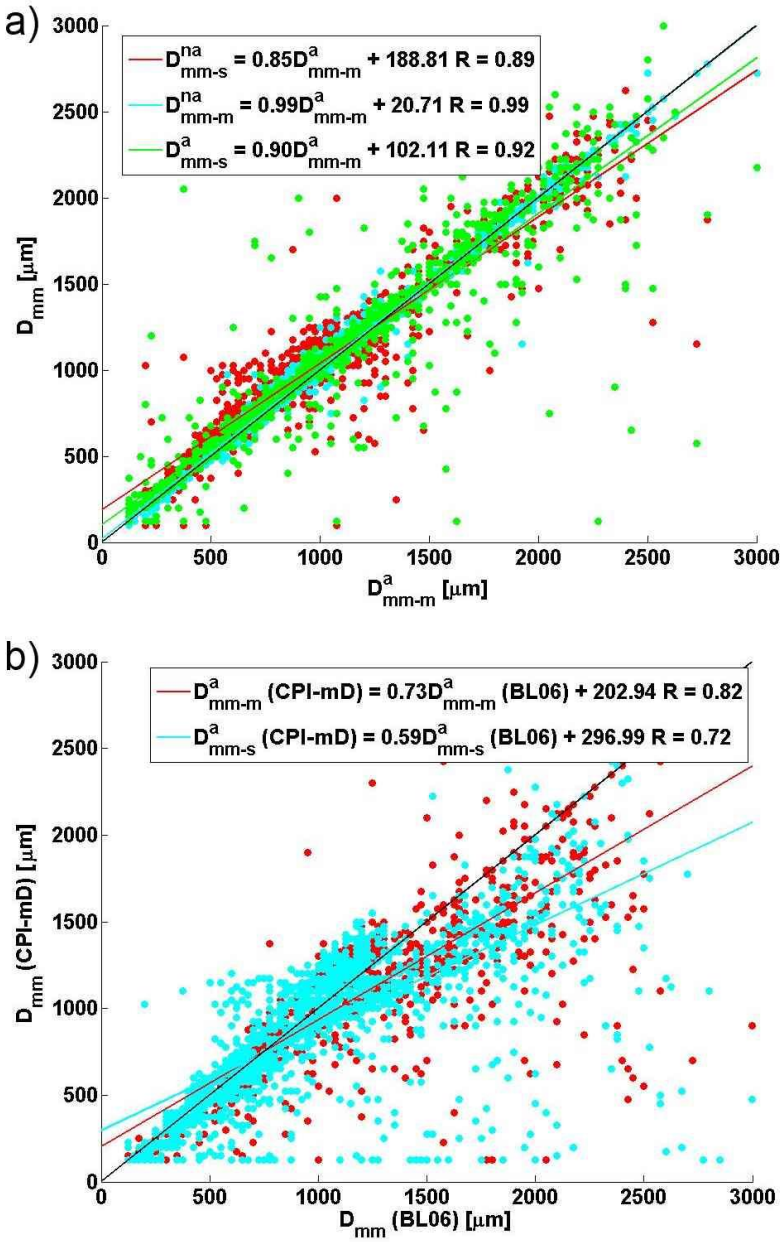


Figure 3.5. (a) As in Fig. 3.2, except for $D_{mm\text{-}s}^{\ na}$, $D_{mm\text{-}s}^{\ a}$, and $D_{mm\text{-}m}^{\ na}$ as a function of $D_{mm\text{-}m}^{\ a}$. 'BL06' method was used to calculate D_{mm}. **(b)** Scatter plot of $D_{mm\text{-}m}^{\ a}$ and $D_{mm\text{-}s}^{\ a}$ calculated using the 'CPI-mD' method as a function of $D_{mm\text{-}m}^{\ a}$ and $D_{mm\text{-}s}^{\ a}$ calculated using the 'BL06' method. Black line is 1:1. $D_{mm\text{-}s}^{\ na}$, $D_{mm\text{-}s}^{\ a}$, $D_{mm\text{-}m}^{\ na}$, and $D_{mm\text{-}m}^{\ a}$ differ by less than 5%.

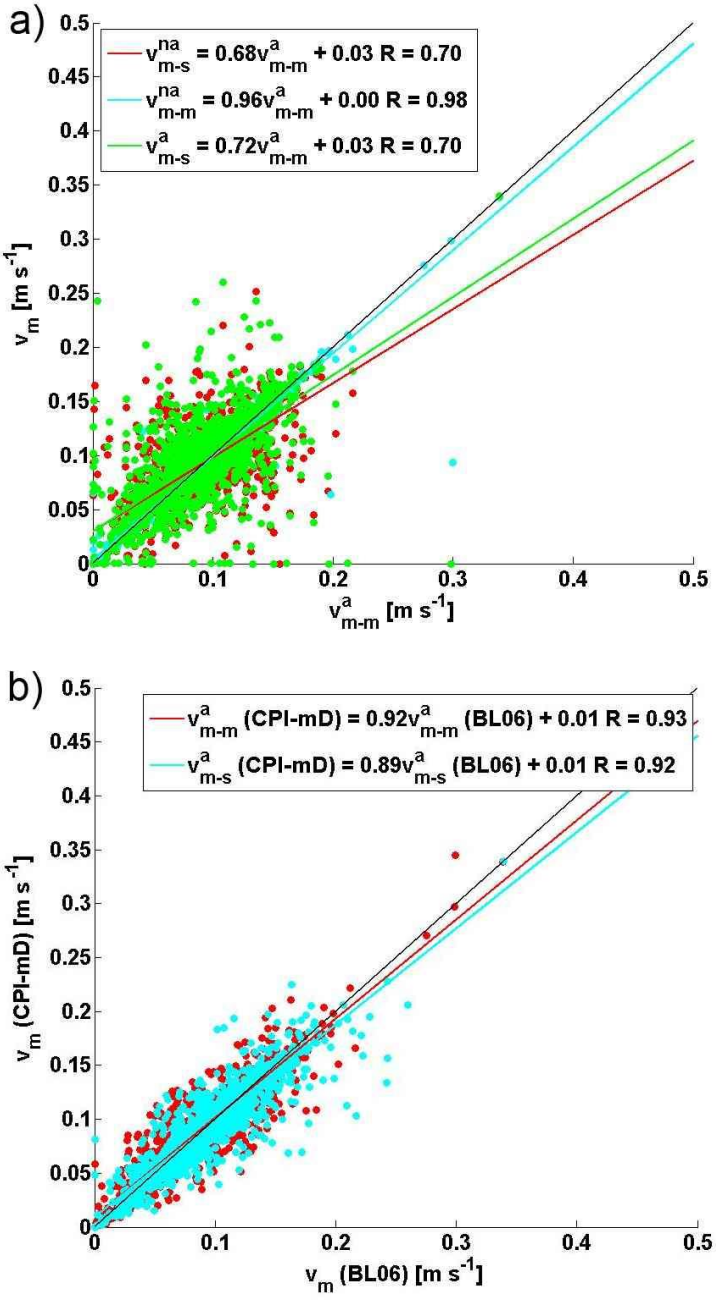


Figure 3.6. (a) As in Fig. 3.2, except for $v_{m\text{-}s}{}^{na}$, $v_{m\text{-}s}{}^{a}$, and $v_{m\text{-}m}{}^{na}$ as a function of $v_{m\text{-}m}{}^{a}$. 'BL06' method was used to calculate v_m. **(b)** Scatter plot of $v_{m\text{-}m}{}^{a}$ and $v_{m\text{-}s}{}^{a}$ calculated using the 'CPI-mD' method as a function of $v_{m\text{-}m}{}^{a}$ and $v_{m\text{-}s}{}^{ast}$ calculated using the 'BL06' method. Black line is 1:1. $V_{m-s}{}^{na}$, $v_{m-s}{}^{a}$, $v_{m-m}{}^{na}$, and $v_{m-m}{}^{a}$ differ by less than 5%.

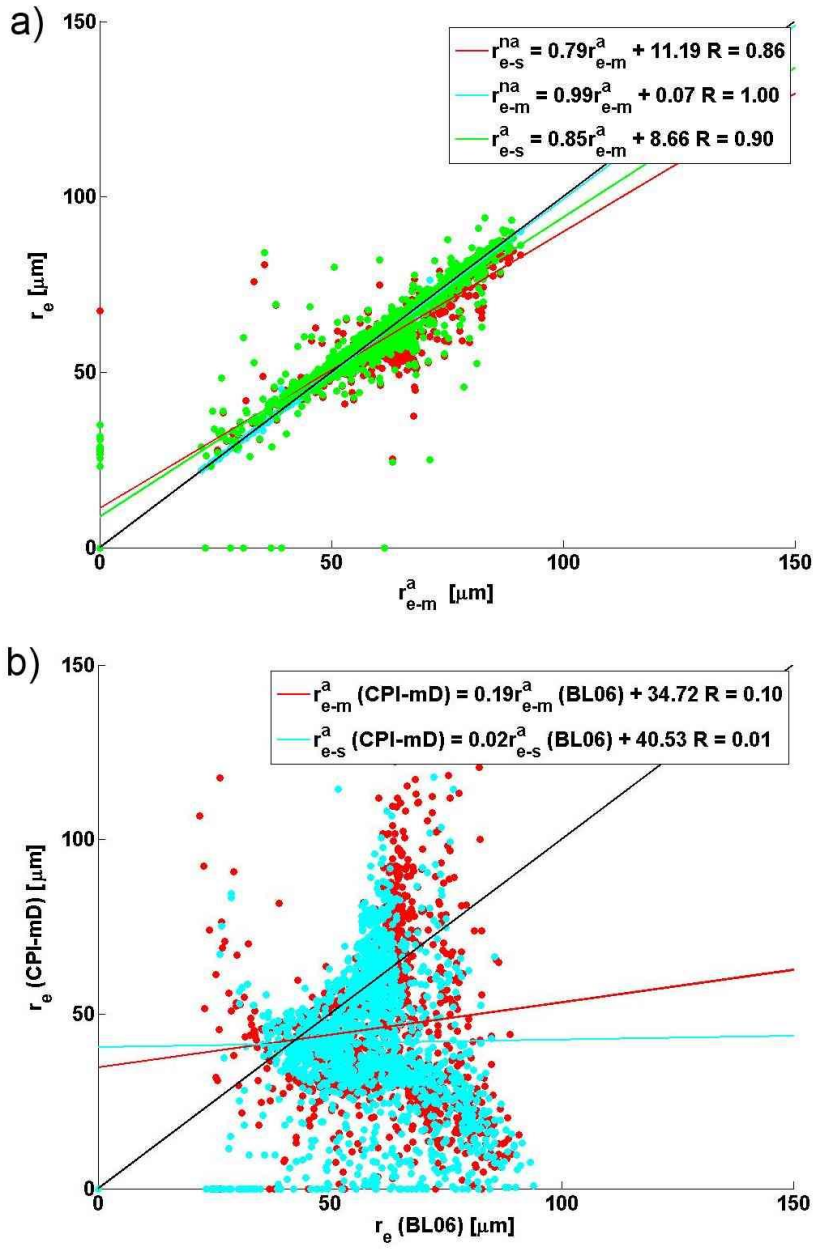


Figure 3.7. (a) As in Fig. 3.2, except for $r_{e\text{-}s}{}^{na}$, $r_{e\text{-}s}{}^{a}$, and $r_{e\text{-}m}{}^{na}$ as a function of $r_{e\text{-}m}{}^{a}$. 'BL06' method was used to calculate r_e. **(b)** Scatter plot of $r_{e\text{-}m}{}^{a}$ and $r_{e\text{-}s}{}^{a}$ calculated using the 'CPI-mD' method as a function of $r_{e\text{-}m}{}^{a}$ and $r_{e\text{-}s}{}^{a}$ calculated using the 'BL06' method. Black line is 1:1. $R_{e\text{-}s}{}^{na}$, $r_{e\text{-}s}{}^{a}$, $r_{e\text{-}m}{}^{na}$, and $r_{e\text{-}m}{}^{a}$ differ by less than 5%.

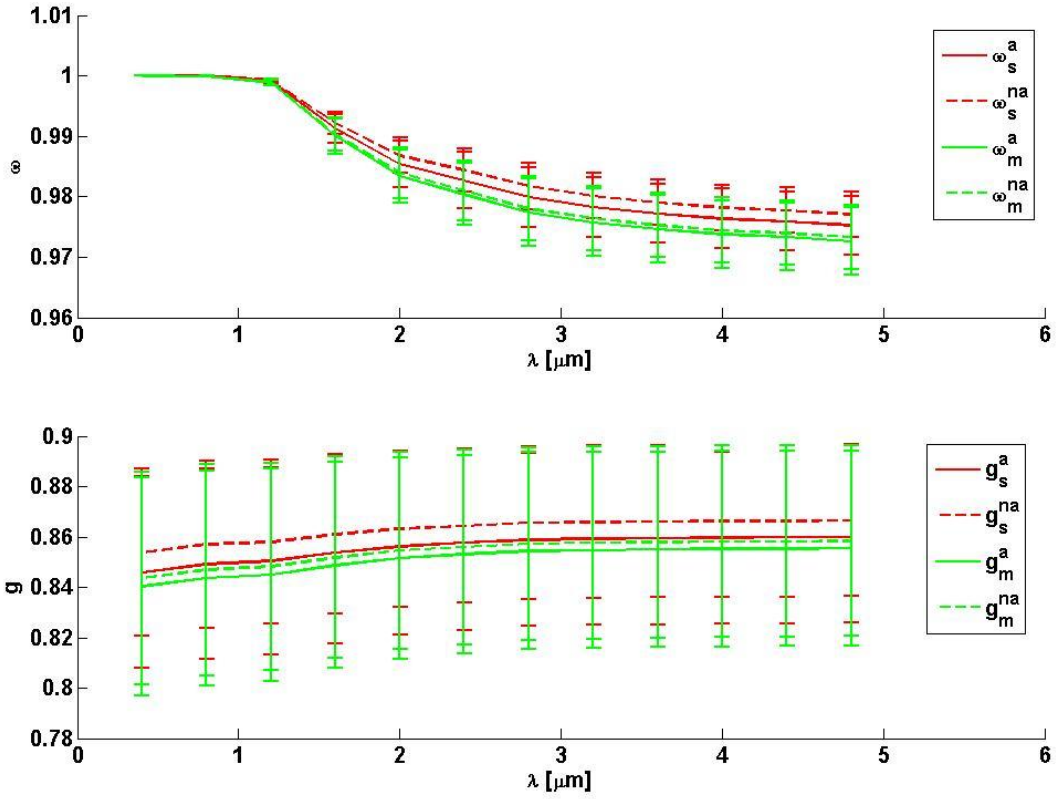


Figure 3.8. (a) Mean narrowband $\omega_{0s}{}^{a}$, $\omega_{0s}{}^{na}$, $\omega_{0m}{}^{a}$, $\omega_{0m}{}^{na}$ as a function of wavelength λ for **(b)** $g_s{}^{a}$, $g_s{}^{na}$, $g_m{}^{a}$, $g_m{}^{na}$ as a function of wavelength λ. Error bars denote one standard deviation about the mean. The difference between the four different versions of ω_0 and g is less than 0.01.

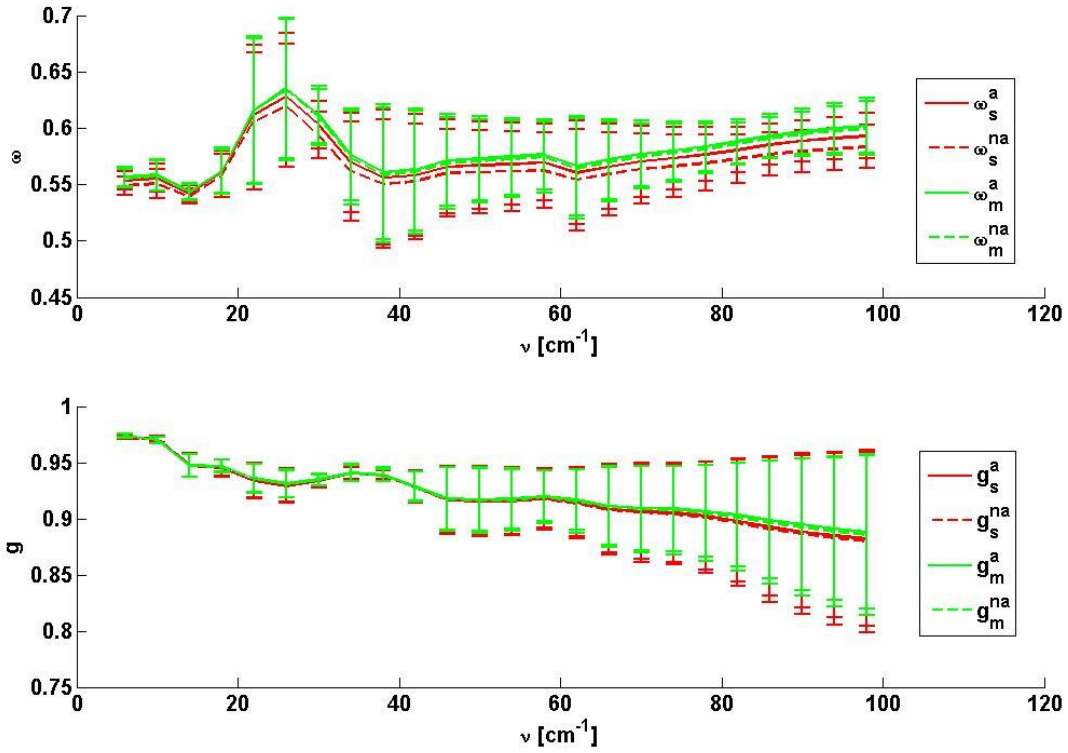


Figure 3.9. (a) Mean narrowband $\omega_{0s}{}^{a}$, $\omega_{0s}{}^{na}$, $\omega_{0m}{}^{a}$, $\omega_{0m}{}^{na}$ as a function of wavenumber υ for infrared wavelength. **(b)** $g_s{}^{a}$, $g_s{}^{na}$, $g_m{}^{a}$, $g_m{}^{na}$ as a function of wavenumber υ. Error bars denote one standard deviation about the mean. The difference between the four different versions of ω_0 and g is less than 0.01.

CHAPTER 4

THE DEPENDENCE OF ICE GAMMA SIZE DISTRIBUTIONS EXPRESSED AS VOLUMES IN N_0-λ-μ PHASE SPACE ON ENVIRONMENTAL CONDITIONS FOR CIRRUS OBSERVED DURING SMALL ICE PARTICLES IN CIRRUS (SPARTICUS).

This chapter will be submitted as an article to *Journal of Geophysical Research* by Jackson, McFarquhar, Frinlind, Atlas, and Lawson (2015). I processed and analyzed the data contained in this chapter as a part of my work, so it is included as a chapter of the dissertation.

4.1. Introduction

Cirrus clouds cover around 30% of the planet (Wylie et al. 2005). They make significant contributions to the Earth's radiation budget by cooling the earth via reflection of sunlight back to space, and by warming Earth through emission of blackbody radiation at a temperature colder than the Earth's surface, preventing heat from escaping the Earth's surface. The balance between these two competing effects is highly dependent on the microphysical properties of cirrus such as particle size and shape (Ackerman et al. 1998). Distributions of ice particle shapes and sizes also determine the ice particle total number concentration *N*, number distribution function *N(D)*, bulk extinction β, ice water content *IWC*, mass weighted terminal velocity v_m, effective radius r_e, and single scattering properties, all which must be known for developing accurate parameterizations of ice particles that are used in models with a variety of temporal and spatial scales. For example, v_m controls cloud coverage and lifetime simulated by climate models (Jakob and Klein 1999; Sanderson et al. 2008), both of which control the total radiative effect. General circulation models (GCMs) are quite sensitive to the parameterization used to represent ice cloud processes which have been noted as a weak link in GCMs (Stephens 2005). *IWC* and mass-weighted

median diameter D_m are output from remote sensing retrievals (Szrymer et al. 2012; Delanoë et al. 2005; 2014); knowledge of ice cloud properties is needed to develop and evaluate such retrievals. To improve and evaluate the representation of processes in models and retrieval schemes, knowledge of how ice properties depend on environmental conditions is required.

General circulation models (GCMs) use bulk microphysics schemes to predict one moment (e.g, *IWC*) or two moments (e.g., *N* and *IWC*) of the size distribution for each class of hydrometeors in the model (i.e. ice, snow, rain) (Gettleman et al. 2012). From *N* and *IWC* and other diagnostic relationships, the parameters of a gamma distribution used to represent the number distribution function *N(D)*, given by

$$N(D) = N_0 D^{\mu} e^{-\lambda D} \quad (1)$$

where N_0 is the intercept, μ is the dispersion, and λ the slope parameter, can be derived. Two of the parameters can be determined from *N* and *IWC* by integrating Equation (1), while the third, typically μ, is set as some prescribed value, frequently zero. In situ observations have shown that N_0, μ, and λ vary as a function of environmental conditions, depending on whether observations are obtained in low pressure systems and fronts (Sassen 1997; Spichtinger et al. 2005a), thunderstorms (Garrett et al. 2004; Jensen et al. 2009; Lawson et al. 2010), or orographic gravity waves (Spichtinger et al. 2005b), as well as depending on temperature and relative humidity (Koop et al. 2000; Krämer et al. 2009). However, sufficient data are not yet available to well characterize how these values vary with environmental conditions, and comparisons between values derived from prior data sets is complicated by use of different probes and processing methodologies on varying projects.

In situ observations are needed and have been used to determine the dependence of bulk cloud properties and the parameters of gamma distributions on environmental conditions (i.e. Dowling and Radke 1990; Ryan 1996; McFarquhar and Heymsfield 1997; Heymsfield and McFarquhar 2002; Heymsfield 2003; McFarquhar and Black 2004; Field et al. 2007; McFarquhar et al. 2007b; Heymsfield et al. 2009; 2013; Lawson et al. 2010; Muhlbauer et al. 2014; Mishra et al. 2014, etc.). McFarquhar and Heymsfield (1997), McFarquhar and Heymsfield (2002), Heymsfield et al. (2003a; 2013) and others have demonstrated that *N* and *IWC* vary as a function of temperature. But, dependence on temperature is not sufficient to explain what controls the ice properties. For example, Lawson et al. (2010) found that *IWC* in tropical anvil cirrus decreased with increasing distance from convection and with anvil age. And, Mulbauher et al. (2014) found that *IWC* was lower in ridge crest cirrus than in the other synoptic or anvil cirrus they observed during the Small Ice Particles in CirrUS (SPARTICUS) project. Further, the variability of microphysical properties with environmental conditions has not been well simulated by GCMs (Joos et al. 2008; Gettleman et al. 2010).

In addition to knowledge of mean ice properties, information about the uncertainties, variances and distributions is required to determine how uncertainties in parameterizations scale up to model output and for evaluating distributions of simulated parameters. Although some studies have examined distributions of and uncertainties in bulk cloud parameters such as *N* and *IWC*, few studies have examined how uncertainties affect the characterization of a SD as a gamma function. Almost all previous studies have assumed that a gamma fit to a SD is uniquely characterized by a single (N_0, μ, λ) triplet (i.e. Heymsfield et al. 2013; Mulbauher et al. 2014). But, McFarquhar et al. (2014) proposed that an ellipsoid of equally realizable solutions in (N_0, μ, λ) phase space should be used to characterize a SD. Within this ellipsoid, N_0, μ, and λ may vary

substantially, but are strongly correlated. To characterize cirrus size distributions, the variation of these ellipsoids in N_0- μ-λ phase space as a function of environmental conditions must be determined using a common processing framework.

The large dataset of cloud microphysical properties needed to characterize the variation of these ellipsoids in N_0- μ-λ phase space, and of N, β, and IWC as a function of environmental conditions, was obtained during The Small Ice Particles in Cirrus (SPARTICUS) project. During SPARTICUS, the Stratton Park Engineering Company (SPEC) Learjet flew 101 missions through mid-latitude ice clouds over the US Central Plains from January to June 2010 at temperatures (T) from -70℃ to 0℃. This provided an unprecedented amount of data in mid-latitude cirrus for identifying statistical relationships between mid-latitude cirrus microphysical properties and environmental conditions. In this paper, the variation of N, β, IWC, N_0, μ, and λ as a function of T, and formation mechanism is addressed. Comparisons of N, β, and IWC against values derived from previous field data (i.e. Heymsfield et al. 2003; 2013) are also made.

This study extends past studies of in situ observations of cirrus microphysical properties. First, SPARTICUS represents a comprehensive set of cirrus measurements collected under a wide range of seasonal and environmental conditions by state of the art probes with tips to mitigate shattering artifacts (Korolev et al. 2011; 2013). The data from these probes were processed using algorithms designed to remove shattering artifacts (Field et al. 2006; Lawson 2011; Jackson et al. 2014). Second, complete uncertainty and statistical analysis is used in the derivation of the microphysical parameters complimenting previous SPARTICUS analysis that did not consider such uncertainties. For example Mishra et al. (2014) found that v_m was strongly correlated with effective diameter, and Mulbauher et al. (2014) developed relationships of μ as a function of λ for subcategories of synoptic and convective cirrus during SPARTICUS. This study

extends upon those studies by characterizting how volumes of equally realizable solutions of (N_0, μ, λ) vary with environmental conditions and by using a broader classification of cirrus in order to include more of the SPARTICUS dataset. This broader classification is also consistent with classifications used in most previous studies of how cirrus varies with environmental conditions (i.e. McFarquhar and Heymsfield 2002; Heymsfield 2003; Heymsfield et al. 2013).

The rest of this chapter is organized as follows. Section 4.2 overviews the meteorological conditions sampled during SPARTICUS. Section 4.3 describes the instrumentation used and how the data were processed to determine bulk cloud parameters and gamma fits to the SDs. Section 4.4 describes how the cirrus microphysical parameters varied with environmental conditions and Section 4.5 summarizes the principal findings of this study.

4.2. Classification of meteorological conditions during SPARTICUS

In this study, data from 101 missions flown by the SPEC Learjet during SPARTICUS in mid-latitude ice clouds at -70℃ < T < 0℃ are used. Flights were flown in various regions of the US Central Plains. Twenty-seven of these flights were flown within 100 km of the Southern Great Plains (SGP) site in Lamont, OK in the vicinity of ground-based remote sensors. Since the SPEC Learjet was based in Boulder, CO for SPARTICUS, many flights sampled cirrus between Colorado and Oklahoma, in addition to dedicated flight time near the SGP site.

A qualitative analysis of the cloud macrophysical properties was performed for each mission during SPARTICUS. A qualitative analysis of visible and infrared images from the Geostationary Operational Environmental Satellite (GOES), and Weather Surveillance Radar (WSR) 88D images was used to classify each mission sampled by the SPEC Learjet. Figure 4.1 shows the radar reflectivity from the Millimeter Cloud Radar at the SGP site for all cases when

the SPEC Learjet flew within 30 km of the SGP site showing widely varying and vertically inhomogeneous structures.

Each mission was classified into one or two of three categories according to whether the sampled cirrus formed synoptically, convectively, or orographically. Figure 4.2 shows examples of GOES satellite and WSR 88D radar imagery in the vicinity of the SPEC Learjet during example missions classified as synoptic, convective, and orographic. A mission was classified as convective if radar reflectivity greater than 43 dBZ was present within 100 km of the cloud sampled at any time during its life (Rigo and Llsat 2004). For example, Figure 4.2b shows that the SPEC Learjet flew through a cloud system in eastern Colorado, circled in Figure 4.2e, on 22 April 2010, where a maximum reflectivity of ~60 dBZ was observed 90 km away and 1.5 h before the in situ sampling. In contrast, no echoes > 10 dBZ were present for the synoptic and orographic cirrus cases, indicating the lack of precipitation. A cloud was classified as orographic if its point of origin, detected by the GOES, was over a mountain. For example, In Figure 4.2c the SPEC Learjet is shown to have sampled cirrus originating over the Front Range in northern Colorado on 22 Mar 2010. All cases which did not meet the criteria for convective or orographic clouds were classified as synoptic. Table 4.1 shows the missions classified as synoptic, orographic, and convective. In total 82 missions were classified as synoptic, 16 as convectively generated cirrus, and 3 as orographic cirrus. On one flight (April 22b) both synoptic and anvil cirrus were sampled. Data from the orographic cases are not presented in this paper because there were not enough to have a statistically significant sample. Figure 4.3 shows the frequency distribution of T for the synoptically and convectively generated cirrus during SPARTICUS. 95% of the observations were sampled at $T < -20$℃, showing that warmer clouds, which should

not be considered to be cirrus, were typically only sampled upon ascent from and descent to the runway.

The classification criteria differ from those used by Muhlbauer et al. (2014), who only used 59% of the SPARTICUS data to classify cirrus into more specific categories such as ridge-crest cirrus, frontal cirrus, subtropical jetstream cirrus, and anvil cirrus. More data are included in the present study, having the advantage that more climatologically representative data are used. Furthermore, the broader classification into only convective and synoptic cirrus allows for easier comparison with past parameterizations and analyses.

4.3. Processing of SPARTICUS data

4.3.1. Instrumentation used

A variety of instruments mounted on the SPEC Inc. Learjet measured cloud microphysical properties and ambient conditions. Only probes used in the analysis presented in this paper are discussed here. The temperature T and vertical velocity w were measured by the Aircraft Integrated Metorological Measurement System (AIMMS-20). The w is measured with an accuracy of 0.75 m s^{-1} (Beswick et al. 2008) and is most useful for determining whether the Learjet was in updrafts, downdrafts, or stratiform regions. Five probes measured particle SDs. The 2D Stereo Probe (2DS), equipped with Korolev tips to reduce the impact of shattered artifacts (Lawson 2006; Korolev et al. 2011; 2013), nominally measured particles with $10 < D < 1280\ \mu m$. A second 2DS equipped with standard tips was also installed on the SPEC Learjet on 23 July 2010. A Fast Forward Scattering Spectrometer Probe (F-FSSP), equipped with a standard FSSP inlet, measured particles with $5 < D < 50\ \mu m$ and particle interarrival time Δt. A 2D Precipitation Probe (2DP), installed on the SPEC Learjet before June 4, measured particles with $200 < D < 6400\ \mu m$ (Knollenberg 1970). A High Volume Precipitation Sampler (HVPS-3)

measured particles with $150 < D < 19200\ \mu m$ for the SPARTICUS missions flown during June 2010. A Cloud Particle Imager (CPI) recorded high resolution images of particles (Lawson et al. 2001).

Section 4.3.2. Data processing

The 2DS data were processed using software developed at the University of Illinois, detailed in Jackson et al. (2014). F-FSSP data were processed using software provided by SPEC. Numerous studies have shown data collected by the F-FSSP and 2DS, even with anti-shatter tips, can be contaminated by shattering artifacts (e.g. Gardiner and Hallett 1985; Gayet et al. 1996; Korolev and Isaac 2005; Field et al. 2006; McFarquhar et al. 2007b, 2011; Jensen et al. 2009; Lawson 2011; Korolev et al. 2011, 2013; Jackson et al. 2014; Jackson and McFarquhar 2014). Algorithms that classify particles as artifacts based on particle interarrival time Δt were applied to both the 2DS and F-FSSP data (Field et al. 2006; Jackson et al. 2014; Korolev and Field 2014). The appropriate Δt threshold was found using the methodology described by Jackson et al. (2014) and Korolev and Field (2014). The algorithm was applied to individual 5000 particle segments in order to allow the Δt threshold to vary as a function of flight time; the division between the peaks for shattered and naturally occurring particles was not constant due to varying particle concentrations. The Δt threshold varied for both the FSSP and 2DS from 10^{-7} s to 10^{-3} s for the flights analyzed. The SDs from the 2DS, 2DP, and HVPS was reduced to 10 s resolution to increase the statistical significance of the number of particles detected in each bin while preserving as much of the fine scale structure as possible (c.f. McFarquhar et al. 2007c). The resolution was chosen so that the uncertainty due to sampling statistics, inversely proportional to the square root of the number of particles in a size bin i.e. $1/N_p^{1/2}$, is reasonably close to 0.1 (10%) (Hallet 2003).

The phase of the clouds was also determined. Since no Rosemount Icing Probe was installed on the aircraft, a direct measurement of icing and hence supercooled water was not available. Therefore, the presence of liquid water was determined by a combination of a manual inspection of CPI imagery for time periods with round droplets (e.g. McFarquhar et al. (2013)) as well as the shape of the FSSP SD using the methodology in McFarquhar et al. (2007a) and Jackson et al. (2012; 2014).

A composite SD derived from a combination of probes is required to calculate the microphysical parameters describing the cloud properties. In order to derive a composite *N(D)*, the *N(D)* from the different probes were compared in their overlapping size ranges. To simplify notation used in this comparison and throughout the paper, the number concentration of particles with $a < D < b$ μm from probe *xxx* is hereafter denoted $N_{xxx\ a\text{-}b}$. For example, the number concentration of particles from the FSSP with $5 < D < 45$ μm is denoted $N_{FSSP\ 5\text{-}45.}$ Figure 4.4a shows $N_{FSSP\ 5\text{-}45}/N_{2DS\ 5\text{-}45}$ after shattered artifacts have been removed as a function of median mass diameter D_{mm} derived from the 2DS. Although the average $N_{FSSP\ 5\text{-}45}/N_{2DS\ 5\text{-}45}$ increases from 0.3 to 2 as D_{mm} increases from 30 μm to 2 mm, $N_{FSSP\ 5\text{-}45}/N_{2DS\ 5\text{-}45}$ varies from 10^{-3} to 50, exhibiting a large amount of scatter for individual time periods.

The $N_{2DS\ 5\text{-}45}$ can be uncertain by a factor of at least 2 to 3 due to the poorly defined depth of field of optical array probes for particles with $D < 110$ μm and due to the large fractional uncertainty in particle size for small particles caused by the resolution of the diodes. The $N_{FSSP\ 5\text{-}45}$ can also be highly uncertain as Mie theory may not correctly size ice particles due to their nonspherical shape. Therefore, the comparison was repeated over multiple size ranges. The $N_{FSSP\ 25\text{-}45}/N_{2DS\ 25\text{-}45}$ is shown in Figure 4.4b and includes only particles for which at least three 2DS diodes were shadowed. $N_{FSSP\ 25\text{-}45}/N_{2DS\ 25\text{-}45}$ increases from 1 to 20 as D_{mm} increases. The $N_{FSSP\ 25\text{-}}$

$_{45}/N_{2DS\ 25\text{-}45}$for individual time periods varies from 10^{-2} to 100, indicating comparable scatter in Figure 4.3b compared to $N_{FSSP\ 5\text{-}45}/N_{2DS\ 5\text{-}45}$. The large scatter in $N_{FSSP\ 5\text{-}45}/N_{2DS\ 5\text{-}45}$ and $N_{FSSP\ 25\text{-}45}/N_{2DS\ 25\text{-}45}$, and the mean values of $N_{FSSP\ 25\text{-}45}/N_{2DS\ 25\text{-}45}$ indicate that the uncertainty in the number concentrations of particles $5 < D < 45$ μm and of particles $25 < D < 45$ μm due to the aforementioned factors is high. Therefore, number concentrations of particles $5 < D < 45$ μm are not reported in this study.

The $N(D)$ for $D > 45$ μm in the overlapping size ranges between the 2DS and 2DP was also compared. Following Jackson et al. (2012), N_{2DP}/N_{2DS} is shown as a function of D in Figure 4.4 at differing true airspeeds TAS, the ranges chosen to provide similar sample sizes in the different ranges. For $D < 900$ μm, images recorded by the 2DP and HVPS-3 have 4 or fewer shaded photodiodes so there is considerable uncertainty in the size and shape of such particles. Figure 4.5 shows that $0.5 < N_{2DP\ 700\text{-}900}/N_{2DS\ 700\text{-}900} < 2$ when TAS > 170 m s^{-1}, indicating that the SDs from the 2DS and 2DP agree well in this size range. It can also be seen that N_{2DP}/N_{2DS} increases with D, which may be caused by particles of D greater than the 2DS's nominal range of 1200 μm being undersampled by the 2DS due to the use of reconstruction techniques. Therefore, the 2DS is the probe used to characterize $N_{45\text{-}900}$, and the 2DP for $N_{900\text{-}6400}$. For the missions when the 2DS and HVPS-3 were both installed, neither the 2DS nor the HVPS-3 recorded particles of $D > 1$ mm, so no comparison of $N(D)$ from the 2DS and HVPS-3 was conducted. Therefore, the HVPS-3 data were not used for those missions.

4.3.3. Derivation of bulk properties

This section describes how the composite SDs derived following the methodology in Section 4.3.2 are used to calculate the bulk microphysical properties of cirrus. The bulk extinction β is given by twice the cross sectional area directly measured by the 2DS or 2DP. To

estimate *IWC*, information about the relationship between crystal mass *m* and *D*, which depends on ice crystal habit, is required. Following Jackson et al. (2014), *IWC* is derived using

$$IWC = \sum_j \sum_k f_k(D_j)\alpha_k D_j^{\beta_k} N(D_j)\Delta D_j \quad (2)$$

where $f_k(D_j)$ is the fraction of crystals in the bin centered at D_j having crystal habit k, α_k and β_k are the habit-dependent coefficients listed in Table 4.2 that define the mass of an individual crystal $m = \alpha_k D_j^{\beta k}$,and $N(D_j)$ is the number distribution function for bin j with midpoint D_j and width ΔD_j. For SPARTICUS, the 2.3 µm resolution CPI images were used to determine $f_k(D_j)$ using the habit classification scheme of Um and McFarquhar (2009). Because the CPI has a smaller sample volume than the 2DS or 2DP, the averaging period of 60 s was required to obtain a statistically significant sample was larger than the 10 s period required for the 2DS or 2DP. Hence the habit distributions were applied to each of the 2DS/2DP/HVPS size distributions occurring within the CPI averaging period.

Using the *IWC* determined from Equation (2), a threshold of 0.001 g m^3 was used to determine the presence of cloud (McFarquhar and Cober 2004), which provided 1179 min of flight data. If a threshold of *IWC* > 0.005 g m^{-3} is used to define cloud instead, 851 min of flight data exist.

4.3.4 Derivation of ellipsoids representing gamma fits

This study uses a modified version of the incomplete gamma fitting (IGF) technique (McFarquhar et al. 2014) to derive a volume of equally realizable solutions in (N_0, λ, μ) phase space to represent each SD as one or more gamma functions. Here, a fit to a gamma distribution function is defined as

$$N_{fit}(D) = N_0\left(\frac{D}{D_0}\right)^{\mu} e^{-\lambda D} \quad (3)$$

where D_0 = 1 cm, and all other parameters are as in Equation (1). The IGF fitting methodology is described by McFarquhar et al. (2014) and only briefly summarized here. In this section, the *n*th moment of the SD determined from the composite SD is denoted $M_{n,obs}$, and determined from the fit SD is denoted $M_{n,fit}$.Using $M_{0,obs}$, $M_{2,obs}$, and $M_{3,obs}$, the χ^2 difference from SD $M_{0,fit,}$ $M_{2,fit}$, and $M_{3,fit}$ is minimized using an incomplete gamma function to take into account that the measured and fit distributions only cover particles between D_{min} = 45 μm and D_{max} = 6.4 mm. All N_0, λ, and μ parameters that give χ^2 within $\Delta\chi^2$ of the minimum χ^2, χ^2_{min}, are denoted as equally realizable solutions where $\Delta\chi^2$ is determined from the statistical uncertainty in the observed SDs. The volume of equally realizable solutions can be represented by an ellipse in N_0-λ-μ phase space, where the principal axis and the lengths are determined by the eigenvectors and eigenvalues of the Hessian matrix corresponding to χ^2_{min}.

4.3.5 Technique to find modes in SDs

The IGF technique as introduced by McFarquhar et al. (2014) assumes that a SD is adequately characterized by a single gamma function, where $-1 < \mu < 5$ to make its application in models easier. However, for some SDs, multiple modes are seen as depicted in Figure 4.6a. The IGF technique was thus modified to fit separate functions to each mode and to automatically determine the boundaries between the modes. To determine whether multiple modes in a SD exist and to identify boundaries between modes, a first derivative test is applied. The first derivative of the number distribution function is calculated for each bin midpoint using a centered difference approximation where

$$\frac{d(N(D_j))}{dD} = \frac{N(D_{j+1}) - N(D_{j-1})}{(D_{j+1} + \Delta D_{j+1}/2) - (D_{j-1} - \Delta D_{j-1}/2)} \quad . \tag{4}$$

Ideally, the local minima of *N(D)* are located at the points where $\frac{d(N(D_j))}{dD} = 0$ and $\frac{d^2(N(D_j))}{dD^2} > 0$. However, local variability in the SD due to inadequate sample statistics can also produce local minima and cause false identification of modes. Therefore, a noise threshold σ for $d(N(D_j))/dD$ was calculated based on the uncertainty of $N(D_j)$ due to sampling statistics

$$\sigma = \frac{N(D_j)}{\sqrt{N_p}((D_{j+1} + \Delta D_{j+1}/2) - (D_{j-1} - \Delta D_{j-1}/2))} \quad (5)$$

The algorithm searches for all possible D_j where $d(N(D_j))/dD = \pm \sigma$., with solutions henceforth denoted $D_{c,i} = \{D_{c,1}, D_{c,2}, \ldots\}$. Thereafter, for all $D_{c,i}$ where $\frac{d(N(D_{c,i}))}{dD} < 0$ and $\frac{d(N(D_{c,i+1}))}{dD} > 0$, so that $\frac{d^2(N(D_j))}{dD^2} > 0$, $(D_{c,i} + D_{c,i+1})/2$ is identified as a boundary between modes. The number of modes is determined by the number of $D_{c,i}$ that meet this criterion.

In order to derive the $M_{n,fit}$ for a multimodal SD, contributions from the different modes are added as given by

$$M_{n,fit} = \sum_i \int_{D_{min,i}}^{D_{max,i}} N_{0_i} (\frac{D}{D_0})^{n+\mu_i} e^{-\lambda_i D} dD \quad (6)$$

For each mode *i*, an ellipsoid of equally realizable (N_{0i}, μ_i, λ_i) is generated. N_{01}, μ_1, and λ_1 are then randomly chosen from the ellipsoid of equally realizable (N_{01}, μ_1, λ_1). To ensure continuity in *N(D)*, a constraint such that

$$N_{0_i} (\frac{D_{max_i}}{D_0})^{n+\mu_i} e^{-\lambda_i D_{max_i}} = N_{0_{i+1}} (\frac{D_{max_i}}{D_0})^{n+\mu_{i+1}} e^{-\lambda_{i+1} D_{max_i}} \quad (7)$$

needs to be applied. In order to do this, a factor *f* to force continuity is calculated using

$$f = N_{0_i}\left(\frac{D_{max_i}}{D_0}\right)^{n+\mu_i} e^{-\lambda_i D_{max_i}} / N_{0_{i+1}}\left(\frac{D_{max_i}}{D_0}\right)^{n+\mu_{i+1}} e^{-\lambda_{i+1} D_{max_i}} \quad (8)$$

Since finding the (N_{0i}, μ_i, λ_i) for each subsequent mode that enforces this constraint requires much computation time, an optimization is performed where a point is randomly chosen from the ellipse of equally realizable triplets (N_{0i+1}, μ_{i+1}, λ_{i+1}) are chosen until a point is chosen such that $.6 < f < 1.67$ or 40 points have been chosen. If 40 points have been chosen and $0.6 < f < 1.67$ is not satisfied, then the data point is excluded from the analysis as the multimodal fit is discontinuous. 6.0% of the multimodal SDs are discontinuous under this criteria. N_{0i} is then multiplied by f before the integration in Equation (6) is performed.

Figure 4.6a shows an example of a bimodal *SD* from 14 Mar 2010 during SPARTICUS. The associated calculations of σ and $d(N(D_j))/dD$ are shown in Figure 4.6b. Fifty randomly chosen equally realizable unimodal IGF fits and 50 randomly chosen equally realizable multimodal IGF fits are also shown in Figure 5a. The black circles in Figure 4.6b show the $D_{c,i}$ identified for this SD. The $\frac{d(N(D_{c,1}))}{dD} < 0$ and $\frac{d(N(D_{c,2}))}{dD} > 0$ in Figure 4.6b, so that the boundary between modes is placed at $(D_{c,1} + D_{c,2})/2$. The boundaries between $D_{c,2}$ and $D_{c,3}$ do not correspond to modes because $\frac{d(N(D_{c,2}))}{dD} > 0$ and $\frac{d(N(D_{c,3}))}{dD} > 0$, so that there is no local minima between $D_{c,2}$ and $D_{c,3}$. The $\frac{d(N(D_{c,3}))}{dD} > 0$ and $\frac{d(N(D_{c,4}))}{dD} < 0$, indicating a local maxima between $D_{c,3}$ and $D_{c,4}$. The multiplying factor f varies from 0.78 to 1.32 with a median of 1.01 in Figure 4b. The mean χ^2 in Figure 4.6a is 0.051 for the unimodal fits and 0.078 for the multimodal fits. The algorithm detected that 83.3% of the 7,074 SDs had one mode, 16.7% of the SDs had 2 modes, and no SDs had more than 2 modes during SPARTICUS. Discontinuities in the composite SD at 900 μm caused by the use of the 2DP instead of the 2DS for particles $D > 900$

μm can also produce a false mode in the SDs. Since $N_{>900}$ is nonzero for 11% of all multimodal SDs, discontinuities in the composite SDs are not likely the cause of bimodality.

Figure 4.7 shows the $M_{n,fit}$ derived by choosing the (N_0, μ, λ) that minimizes χ^2 as a function of $M_{n,SD}$ for the 0th, 2nd, and 3rd moments that were used in the fitting procedure for the time periods when multiple modes were identified. The $M_{n,fit}$ was calculated twice: by first fitting one mode to all SDs regardless of the number of modes identified in the procedure, and second, by fitting multiple modes to the SD. The goodness of fit (R^2) values in Figure 4.6a indicate that there is equal correlation (R^2 = 0.99) in $M_{0,fit}$ to $M_{0,SD}$ from the multimodal SDs compared to the unimodal SDs (R^2 = 0.98) when choosing the (N_0, μ, λ) that minimizes χ^2. However, the slope of the best fit of $M_{0,fit}$ to $M_{0,SD}$ from both the unimodal and bimodal fits are equal to 1. However, there is comparable scatter (R^2 = 0.94) in $M_{0,fit}$ to $M_{0,SD}$ from the multimodal SDs compared to the unimodal SDs when randomly choosing the (N_0, μ, λ) from the volume of equally realizable solutions. This indicates variability induced in $M_{0,fit}$ by randomly choosing the (N_0, μ, λ) from the volume of equally realizable solutions is greater than the change in $M_{0,fit}$ by fitting multiple modes to the SD rather than one. Furthermore, the R^2 values in Figures 4.6 b,c indicate that there is comparable scatter (R^2 = 0.98) in $M_{2,fit}$ to $M_{2,SD}$ and $M_{3,fit}$ to $M_{3,SD}$ compared to the fit of $M_{0,fit}$ to $M_{0,SD}$ for the multimodal SDs and compared to $M_{2,fit}$ to $M_{2,SD}$ and $M_{3,fit}$ to $M_{3,SD}$ from the unimodal SDs (R^2 = 0.99). The slopes of 1.06 and 1.03 of the fits $M_{2,fit}$ to $M_{2,SD}$ and $M_{3,fit}$ to $M_{3,SD}$ from the multimodal SDs are closer to 1 than the slopes of 1.14 and 1.27 of the fits $M_{2,fit}$ to $M_{2,SD}$ and $M_{3,fit}$ to $M_{3,SD}$ from the unimodal SDs. When (N_0, μ, λ) is randomly selected, then the slopes of the aforementioned fits range from 0.8 to

1.2. Therefore, random selection of (N_0, μ, λ) has a greater impact on the calculated moments than does using multiple modes of the SDs to represent the fit.

4.4. SPARTICUS microphysical properties and IGF fits

4.4.1. N(D) as a function of environmental conditions

In this section, the dependence of observed cirrus *N(D)* on *T*, and cirrus formation mechanism is described. Normalized frequency distributions of *N(D)* in the synoptically and convectively generated cirrus were generated as a function of *T* in order to determine how *N(D)* varies as a function of environmental conditions. Figures 4.8a-h show the normalized frequency distribution of *N(D)* for four different temperature ranges chosen such that approximately equal numbers of SDs are included in each range. The color shading represents the normalized frequency of occurrence of *N(D)*, the solid line represents the median *N(D)*, and the dashed lines represent the 10^{th} and 90^{th} percentiles.

Figure 4.8 shows that the mean boundary between the first and second mode was located at 115 μm, similar to prior studies that have shown the boundary between modes to be from 100 to 125 μm (Mitchell et al. 1996; McFarquhar and Heymsfield 1997; Ivanova et al. 2001; Jensen et al. 2009). This finding with the SPARTICUS data is important because many of the previous studies did not use probes with anti-shatter tips or were not processed with the same techniques to eliminate artifacts as were the SPARTICUS data. Mitchell et al. (1996) hypothesized that the smaller particle mode corresponds to particles that have grown primarily by vapor deposition, while the large particle mode, having a lower slope and hence a wider range of particle sizes, corresponds to particles that have grown by both vapor deposition and aggregation. Lawson et al. (2010) hypothesized that the mode of smaller particles in their observed bimodal distributions in

updrafts of tropical anvil cirrus corresponded to particles that formed via homogeneous nucleation. For the SPARTICUS data, 16.8% of the SDs in synoptic cirrus were bimodal, while 11.8% of the SDs in convectively generated cirrus were bimodal. Furthermore, there is increased bimodality of the SDs as *T* increases, as 5.1% of the SDs are bimodal for T < -45°C, while 37% of them are bimodal for T > -45°C, consistent with Zhao et al. (2010) who hypothesized that the increasing bimodality with *T* was caused by heterogeneous nucleation in the presence of larger particles sedimenting from above.

For both synoptic and convective cirrus, the maximum *D* increases with *T*, from 2 mm to 4 mm for synoptic cirrus in Figures 4.8a-d, and from 0.5 mm to 4 mm for convectively generated cirrus in Figures 4.8e-g. The mean number concentration of particles with $D < 115$ μm $N_{< 115}$ decreases from 23.1 L^{-1} at $T < -54$°C to 19.5 L^{-1} at T > -40°C for synoptic cirrus and from to 33.6 L^{-1} to 18.6 L^{-1} indicating higher amounts of small ice crystals at colder temperatures. The differences are greater than statistical uncertainty of 4.3% for the mean $N_{< 115}$. This indicates that particles are growing to larger sizes as *T* increases. In addition, there is a greater number concentration of smaller ice crystals at $T < -54$°C in convective cirrus than in synoptic cirrus, which could be caused by enhanced homogeneous nucleation in convective updrafts. However, the mean *w* in convectively generated cirrus was 0.02 m s^{-1} at $T < -54$°C, so it is clear that the majority of the sampling was in the convectively generated anvils and not the convection itself.

There are some differences between *N(D)* from SPARTICUS and past observations of *N(D)* in other cirrus depicted in Figure 4.8. In Figures 4.8a-d, the median *N(D)* from Heymsfield et al. (2013) made in arctic and midlatitude cirrus is equal to *N(D)* for $D < 500$ μm for -54°C $< T < -40$°C in synoptic cirrus according to a Student's t test at a 95% significance level, but less for $D > 500$ μm. This could be due to the fact that the Heymsfield et al. (2013) curve includes arctic

cirrus which had SDs with higher slopes for the same *T* compared to the midlatitude cirrus observed by Heymsfield et al. (2013). In Figures 4.8f-g, the *N(D)* from Heymsfield et al. (2013) is less than the *N(D)* for all *D* observed in convective cirrus according to a Student's t test at a 95% significance level. Since Heymsfield et al. (2013) sampled tropical cirrus that typically generated by weaker updrafts than midlatitude cirrus, (Heymsfield and McFarquhar 2002), this could explain the difference. In Figures 4.8c,g, the Heymsfield et al. (2003a) *N(D)* is different in a statistically significant sense than the median *N(D)* from SPARTICUS, but this difference is lower than in Figures 4.8d,h. The observations of Heymsfield et al. (2003a) in Figures 4.8c,g are taken at cloud top, which may have lower *N(D)* compared to data taken at varying altitudes relative to cloud top.

4.4.2 IGF fits as a function of environmental conditions – unimodal SDs

In this section, the dependence of gamma fit parameters on temperature and cirrus formation mechanism is discussed. For families of SDs obtained under similar environmental conditions, volumes of equally realizable solutions in (N_0, μ, λ) phase space are generated as based on the fraction of ellipses characterizing single SDs that encompass a point in the (N_0, μ, λ) phase space. The variation of the volumes for either single modes or two modes of the SD with *T* and formation mechanism, is then examined in order to hypothesize what microphysical processes are dominant in the cirrus observed during SPARTICUS.

In order to construct the volumes of equally realizable solutions for a family of SDs, McFarquhar et al. (2014) fit a single ellipse around all (N_0, μ, λ) that are contained within at least 1% of the volumes in the family of SDs. However, the superposition of many ellipses is not usually a true ellipse. Therefore in order to account for the possibly nonelliptical shape of the

volume, a modification of their technique to construct the volume representing the family of SDs is presented here. A separate ellipse in (N_0, μ) phase space of equally realizable (N_0, μ) for 50 specific, equally spaced values of λ between 1 and 5000 cm^{-1} is constructed. For each λ, an ellipse is constructed to contain all of the (N_0, μ) that are contained with at least 1% of the volumes, using the same frequency threshold as McFarquhar et al. (2014). Although the surface of equally realizable (N_0, μ) for a given λ is not a true ellipse either, a more complex shape would be impractical to implement in a model due to the computational time required to represent the shape. These surfaces can then be used in models to describe a SD. Thus, if a model prognoses one moment of a SD, the λ can be determined. With this value of λ, a point is then randomly chosen from the ellipse of equally realizable (N_0, μ) for that given λ and then a Monte Carlo version of any parameterization can be applied.

Since it is difficult to visualize the surface of equally realizable (N_0, μ) for all given values of λ, two dimensional projections of this surface along with normalized frequency distributions are presented in relevant cross-sections to simplify the visualization and interpretation of these volumes. Figure 4.9 shows 2-d cross sections of the normalized frequency of points in the volumes of equally realizable solutions of SDs with one mode in (N_0, λ), (N_0, μ), and (μ, λ) phase spaces for synoptic cirrus for the given range of *T*. The centers of the ellipses in (N_0, λ) and (μ, λ) phase space for each λ are shown by the solid dotted line, and the 2d projection of the surface of equally realizable (N_0, μ) for all given values of λ shown by a solid black curve. The N_0 coordinate of the center of the ellipse is correlated with λ with $R > 0.92$. The range of realizable λ changes from 1 to 3000 cm^{-1} in Figure 4.9a to 1 to 1000 cm^{-1} in Figure 4.8d as *T* increases, while the range of realizable N_0 is 10^2 to 10^{12} L^{-1} μm^{-1} in Figures 4.8e-h and does not seem to depend on temperature. Therefore, λ extends to larger values for a given N_0 as *T*

decreases, consistent with an increase in the number of large particles as T increases. Figure 4.9e-h shows that N_0 is also correlated with μ for synoptic cirrus. The range of realizable μ with T changes with T in Figure 4.9e over -1 to 3 in Figure 4.8e to over -1 to 4 in Figure 4.9h, indicating larger values of μ for equivalent N_0 as T increases, consistent with a decrease in the number of small particles with increasing T. The decrease in the number of small particles and increase in the number of large particles with T suggests that particle growth by vapor deposition and aggregation is more efficient in the clouds sampled at higher T.

In Figures 4.9i-l, the μ coordinate of the center of the ellipse is inversely correlated with λ (R = -0.61) for T < -54℃ but is directly correlated with λ (R > 0.85) for higher T, with μ increasing from 0 to 2 in Figure 4.9j-l as λ increases. This indicates that the SD is narrower when λ is lower for T < -54℃, but broader when λ is lower for T > -54℃. The narrowing of the SD with decreasing λ at T < -54℃ is consistent with growth primarily by vapor deposition. The broadening of the SD with decreasing λ at T > -54℃ is consistent with particle growth by both vapor deposition and aggregation. The direct correlations between μ and λ, and between N_0 and μ have been observed in past studies of cirrus (c.f. McFarquhar et al. 2007c; Heymsfield et al. 2013; McFarquhar et al. 2014), so this study extends their conclusions to data processed using a consistent processing methodology that uses a nonelliptical shape to represent a volume of equally realizable solutions.

Figure 4.10 shows 2-d cross sections of the normalized frequency of points in the volumes of equally realizable solutions of SDs with one mode in (N_0, λ), (N_0, μ), and (μ, λ) phase spaces for convective cirrus at the given ranges of T. Similar to Fig. 4.10, Fig. 4.11 shows that N_0 is correlated with μ and λ for convectively generated cirrus. The μ is inversely correlated with λ for T < -54℃ but is directly correlated with λ for higher T in Fig. 4.10, similar to Fig. 4.8.

The λ and μ extend to larger values for a given N_0 with T in Fig. 4.10 similar to trends seen in Fig. 4.9. However, for T < -54°C, the range of realizable λ extends from 0 to 4000 cm^{-1}, larger than the range in Fig. 4.9. This is consistent with fewer large ice crystals at T < -54°C for convectively generated cirrus compared to synoptically generated cirrus as seen in Section 4.1.

The ellipses in (N_0, μ) phase space shown in Figures 4.9 and 4.10 for a given λ are then used to construct three dimensional volumes. The three dimensional volumes are created by generating the smallest possible surface that encompasses all points that are contained within each ellipse in (N_0, μ) phase space for all values of λ. Figure 4.11 shows these resulting volumes representing the family of SDs for the given ranges of T and during time periods when one mode was identified in the SD. The matrices and centers describing each ellipse are attached as a supplemental MATLAB .mat file. In Fig. 4.11, N_0 varies from 10^2 to 10^{12} L^{-1} μm^{-1}, μ from -1 to 4, and λ from 1 to 4000 cm^{-3}, indicating that the gamma fit parameters can undertake a wide range of values for a given family of SDs. However, it is important that the significant correlations between parameters be taken into account when giving volumes of equally realizable solutions describing the fit parameters.

4.4.3 IGF fits as a function of environmental conditions – bimodal SDs

The previous analysis determined how the volumes of equally realizable (N_0, μ, λ) of the unimodal SDs varied as a function of environmental conditions and T. In this section, a description of how these volumes for the bimodal SDs observed during SPARTICUS, generated using the techniques developed in Section 4.4.2, vary as a function of environmental conditions. The volumes of equally realizable (N_0, μ, λ) between the two modes are also compared to hypothesize the microphysical processes producing the observed results.

Figure 4.12 shows 2-d cross sections of the normalized frequency of points in the volumes of equally realizable solutions of the mode with lower *D* of SDs with two modes in (N_0, λ), (N_0, μ), and (μ, λ) phase spaces for synoptic cirrus at the given ranges of *T*. Similar to Figures 4.9 and 4.10, N_0 and μ are directly correlated with λ, and N_0 and μ are also correlated. However, the ranges of realizable (N_0, μ, λ) are different than in Figures 4.9 and 4.10. The range of realizable λ is 0 to 600 cm^{-3} in Figures 4.11a-d, less than the 0 to 3000 cm^{-3} in Figure 4.8a. This, and the correlation N_0 and μ with λ of indicate that, λ is lower for the same N_0 and μ in the mode with lower *D* than for the SDs with one mode, indicating that bimodal SDs more likely have a greater number of large particles than unimodal SDs, suggesting a particle size threshold for bimodality to occur.

The realizable range of λ of the mode with smaller *D* does not change with *T*. This indicates that the particles in the mode with smaller *D* have not grown more efficiently in the clouds sampled at higher *T*. Since, particles in the mode with smaller *D* may be produced by nucleation in the presence of sedimenting aggregates as suggested by the increasing bimodality, this may indicate that the particles in the mode with smaller *D* have nucleated more recently since they have not grown to larger sizes. The range of realizable μ -0.75 to 4.5 in Fig. 4.12e-h, which is higher than -1 to 4 for the unimodal SDs, indicates that the mode with smaller *D* is narrower than the unimodal SDs. This suggests that growth by aggregation is not as likely for the particles in the mode with smaller *D*.

Figure 4.13 shows 2-d cross sections of the normalized frequency of points in the volumes of equally realizable solutions of the mode with higher *D* of SDs with two modes in (N_0, λ), (N_0, μ), and (μ, λ) phase spaces for synoptic cirrus respectively at the given ranges of *T*. Similar to all the previous analysis, N_0 and μ are still directly correlated with λ, as are N_0 and μ.

The range of realizable λ widens from 100 to 300 cm^{-1} in Fig. 4.13a to 20 to 300 cm^{-1} in Figure 4.13d, indicating lower λ for the same N_0 as T increases, indicating the increased presence of larger particles with increasing T, unlike as was seen for the analysis of the mode with smaller D. Therefore, particles in mode with larger D have grown more efficiently in the clouds sampled at higher T. The range of realizable μ is -1 to 5 in Figure 4.13e-h, wider than the range of μ -0.75 to 4.5 in Figure 4.12. Therefore, there is higher μ for the same λ in the mode of the SD with higher D. This indicates that the mode with higher D is broader than the mode with lower D. This is consistent with Mitchell et al. (1996) who attributed this to particles in the mode with higher D growing by both vapor deposition and aggregation, and particles in the mode with lower D primarily by vapor deposition.

Figure 4.14 shows 2-d cross sections of the normalized frequency of points in the volumes of equally realizable solutions of both modes of SDs with two modes in (N_0, λ), (N_0, μ), and (μ, λ) phase spaces for convectively generated cirrus respectively at the given ranges of T. Similar to what was seen in Figures 4.9 to 4.13, N_0 and μ are directly correlated with λ, and N_0 and μ are correlated. The range of realizable λ is 0 to 500 cm^{-3} in Figure 4.14a,b smaller to what was seen in Figure 4.12, and the range of realizable μ is -1 to 4.5, indicating lower λ for the same μ. This shows that the mode with smaller D is broader in convective cirrus than in synoptic cirrus. In Figures 4.14g-h, the range of realizable λ is 0 to 200 cm^{-3}, and the range of realizable μ is -1 to 4.5. So, there is also lower λ for the same μ in the mode with larger D in convective cirrus than in synoptic cirrus. If, say, convective updrafts enhance the supersaturation, then particles would grow more efficiently by vapor deposition, which could contribute to the broader SDs, and hence larger particle sizes for the convective cases. Similar to

Figures 4.12 and 4.13, there is higher μ for the same λ, which again shows that the mode with larger *D* is broader than the mode with smaller *D*.

The ellipses in (N_0, μ) phase space shown in Figures 4.12, 4.13 and 4.14 for a given λ are then used to construct three dimensional volumes. Figure 4.15 shows these resulting volumes representing the family of SDs for the given ranges of *T* and during time periods when two modes were identified in the SD. In Figure 4.15, N_0 varies from 10^2 to 10^{12} L^{-1} μm^{-1}, μ from -1 to 5, and λ from 1 to 700 cm^{-3}, again indicating a wide range of equally realizable solutions for a given family of SDs. In Figure 4.15, the surfaces representing the mode with smaller *D* have μ from -1 to 3 and -1 to 5 for the mode with larger *D*. This again shows that mode with smaller *D* is narrower than the mode with larger *D*, again suggesting that particles in the mode with smaller *D* grew primarily by vapor deposition, and in the mode with larger *D* by both vapor deposition and aggregation.

4.4.4 Bulk moments as a function of environmental conditions

It is also important to know how bulk parameters vary with environmental conditions to assess what physical processes occur in cirrus, and to evaluate results of numerical models that predict how such cloud parameters vary. In this section, the dependence of *N, β, IWC,* median diameter D_m, and D_{mm} on *T* and cirrus formation mechanism is described. Normalized frequency distributions of *N, β, IWC,* D_m, and D_{mm} for synoptically and convectively generated cirrus are displayed as a function of *T*. In the following figures, the color scale represents the fractional contribution that the parameters within the given bin on the *x*-axis make at the given *T*. The solid black line represents the median of the distribution, and the dashed lines are the 10^{th} and 90^{th} percentiles. A correlation is deemed significant if the probability that a random sample has *R* is

equal to the value stated is less than 5%. The parameterization of N as a function of T from Heymsfield et al. (2013) is shown in red. Field et al. (2007) also developed a parameterization to estimate M_n from M_2 and T considering particles with $D > 100$ μm from which $N = M_0$ and $D_m = M_1/M_0$ can be derived. These parameters are shown in green in the relevant figures.

Figure 4.16a and 4.16b shows the normalized frequency distribution of N for particles with $D > 25$ μm as a function of T for synoptically and convectively generated cirrus respectively. The median of N has a correlation coefficient with T of -0.47 and -0.48 in Fig. 4.16a and Fig. 4.16b, respectively, showing that the median of N has a weak negative correlation with T during SPARTICUS. This reduction in N with T is consistent with increased aggregation efficiency at higher T. The negative correlation of N with T was also shown by Heymsfield et al. (2013), but Kraemer et al. (2009) showed a positive correlation of N with T in their study.

The medians of β and IWC have correlation coefficients of > 0.9 with T for the synoptic cirrus in Fig. 4.16c and Fig. 4.16e, and of > 0.5 with T for the convective cirrus in Fig. 4.16d and Fig. 4.16f. The increases of β and IWC with T are consistent with the observations of Heymsfield et al. (2013) and Lubeke et al. (2013). Because the median N was inversely correlated with T for the SPARTICUS data, the increases of β and IWC with T are consistent with the presence of larger particles in the warmer clouds and with the analysis in Section 4.4.2 that showed that λ decreased with T. The increase in β and IWC with T could also be due to the increased water vapor at higher T, if, for example, the environment was water saturated, which would contribute to more efficient depositional growth. However, without reliable measurements of relative humidity, definitive statements about the rates of depositional growth cannot be made.

Figure 4.17 shows the normalized frequency of median diameter D_m and D_{mm} as a function of T. In Figures 4.17a,b, both the medians of D_m and D_{mm} have correlation coefficients greater than 0.6 with T. The lack of bimodality and observations of D_m < 110 μm for T < -45℃ suggests that aggregation is not a significant microphysical process at these T. Thus, it is more likely that increased depositional growth is contributing to the increase of D_m with T for the cirrus sampled at T < -45℃. However, the observations of D_m > 110 μm and increased bimodality suggests that aggregation becomes a more important factor for the cirrus sampled at T < -45℃ during SPARTICUS.

There are some possible explanations for the differences between the SPARTICUS microphysical properties and prior observations. Kraemer et al. (2009) observed N of 500-10,000 L^{-1} for T > -60℃, at least 20 times higher than the median and extrema of N 10-100 L^{-1} in Figures 4.16a,b. It is worth noting that Kraemer et al. (2009) used a standard FSSP to derive N. Since shattering is hypothesized to greatly contribute to N measured by FSSPs in ice clouds (Korolev et al. 2011), it is possible that the higher N in Kraemer et al. (2009) are induced by shattered particles on the FSSP inlets. The shattering of large particles on the FSSP inlet would also cause N to increase with T since the effective radius of ice crystals increases with T in Kraemer et al. (2009). In Figures 4.16a,b, N from Heymsfield et al. (2013) is up to a factor of 2 less than the median of N from SPARTICUS for T < -30℃. Since the relationship of N with T from Heymsfield et al. (2013) represents a combination of synoptic and convective cirrus, this could contribute to the differences shown. The N from Field et al. (2007) is less than N from Heymsfield et al. (2013), with the median N differing by a factor of 3 for T > -40℃. Given that Field et al. (2007) only considered particles of D > 125 μm in their parameterization, the

differences between SPARTICUS and their data are caused by the omission of small crystals since the median D_m < 125 μm for T < -35℃ in Figures 4.16a,b.

The difference in the range of *IWC* observed by Lubeke et al. (2013) and during SPARTICUS could be due to the fact that Lubeke et al. (2013) observed cirrus in different regions and environmental conditions, especially at T < -55℃. In particular, only tropical and arctic cirrus were sampled by Lubeke et al. (2013) at T < -55℃. However, the observations from Lubeke et al. (2013) at T > -55℃ include midlatitude, tropical, and arctic cirrus. The difference between the median IWC from SPARTICUS and that from Lubeke et al. (2013) is less at T > -55℃. The convection observed by Heymsfield et al. (2013) was tropical rather than extratropical as observed during SPARTICUS, which may contribute to the differences in *N* and *IWC* shown for the convective cirrus. Furthermore, the anvil cirrus sampled were not inside any intense convection, but in stratiform regions of convective systems, which can lead to lower *IWC* than from datasets that include measurements closer to convective cores such as Heymsfield et al. (2013). Nevertheless, the direction of the trends in *N*, D_{mm} and *IWC* with *T* during SPARTICUS are consistent with the direction of the trends observed by Heymsfield et al. (2013) and Lubeke et al. (2013).

4.5. Conclusions

This paper examined how gamma fit parameters and bulk parameters, such as number concentration *N*, bulk extinction β, and ice water content *IWC*, median diameter D_m, and median mass diameter D_{mm} varied as a function of temperature and formation mechanism for midlatitude cirrus during the Small Ice Particles in CirrUS (SPARTICUS) experiment. During SPARTICUS, the SPEC Learjet conducted 101 sorties in midlatitude ice clouds with temperatures *T* from -70℃

< T < 0℃ using state of the art probes with shatter mitigating tips and processed using techniques that remove shattered artifacts. This study is unique in that the most comprehensive data set of midlatitude cirrus properties ever collected in a field project using such shattered artifact removal techniques was analyzed. In addition, this study used a novel technique to represent SDs as gamma functions by representing the fit parameters as a volume of equally realizable solutions in (N_0, μ, λ) phase space. These volumes and bulk parameters were examined as a function of formation mechanism and temperature T and compared to recent observations of ice clouds. This helped identify the processes that contribute to the formation and maintenance of midlatitude ice clouds. The principal advances and findings of this study are as follows:

1. An automated technique was developed to detect the presence of multiple modes in a SD. A multi-mode fit provided better agreement with the 0^{th}, 2^{nd}, and 3^{rd} moments of the SD than the single-moment fits did.
2. When two modes were identified, the boundary between them was on average of D = 115±359 μm, similar to previous observations of bimodal SDs from Mitchell et al. (1996), McFarquhar and Heymsfield (1997), Jensen et al. (2009), Zhao et al. (2010), and Lawson et al. (2010).
3. No more than 2 modes were detected in the SDs during SPARTICUS for any time period. 5.1% of the SDs are bimodal for T < -45℃, while 37% are bimodal for T > -45℃. This increase in bimodality with T was consistent with the finding of Zhao et al. (2010) who hypothesized that the increased bimodality with higher T may be due to heterogeneous nucleation in the presence of particles sedimenting from above.

4. Similar to McFarquhar et al. (2014) and other previous studies, a strong correlation between the gamma fit parameters was observed in both the unimodal and bimodal SDs.
5. For the same $N_{0,}$ values of λ and μ in the volumes of equally realizable solutions increased with T for the SDs with one mode. This indicated that particles were growing to larger sizes by vapor deposition and aggregation as T increased.
6. The μ and N_0 in the volumes of equally realizable solutions were larger for the same λ in the mode with larger D compared to the mode with smaller D. This was consistent with past studies that attributed particles in the first mode primarily to growth by vapor deposition and particles in the second mode to growth by both aggregation and vapor deposition.
7. N was correlated with T while bulk extinction β, and ice water content IWC, median diameter D_m, and median mass diameter D_{mm} were all strongly correlated with T during SPARTICUS for both synoptic and convectively generated cirrus. This was consistent with more efficient particle growth by vapor deposition and aggregation at the clouds sampled at higher temperatures.

The results in this paper apply to the midlatitude synoptic and convective ice clouds sampled during SPARTICUS, and do not necessarily extend to ice clouds in other regions or generated under differing meteorological conditions. These results also extend previously observed correlations between gamma fit parameters to a framework that characterizes each mode of a single SD with an ellipse of equally realizable solutions instead of a single (N_0, μ, λ) triplet. The results in this paper provide knowledge on the microphysical processes that occur in midlatitude ice clouds. Future studies should extend this analysis to ice clouds sampled with

probes that use anti-shatter tips and shattered artifact removal algorithms in a broader range of environmental conditions including those sampled in the tropics and the arctic.

4.6 Acknowledgements

This work was supported by the Office of Biological and Environmental Research (BER) of the U.S. Department of Energy (DE-SC0001279 and DE-SC0008500), and by Stratton Park Engineering Company. Data were obtained from the ARM program archive, sponsored by the U.S. DOE, Office of Science, BER, Environmental Sciences Division. Data were downloaded from the ARM archive, and the raw 2DS data were provided by SPEC. In order to visualize the flight tracks overlaid on GOES visible and infrared satellite images for each flight, the website, given by the link (http://www-pm.larc.nasa.gov/cgi-bin/site/showdoc?docid=4&cmd=field-experiment-homepage&exp=ARM-SPARTICUS-2009) created by Patrick Minnis was used to generate Figure 4.2. In order to visualize historical WSR 88D data, the tool provided by the Iowa State University of Science and Technology, available at (https://mesonet.agron.iastate.edu/GIS/apps/rview/warnings.phtml) was used to generate Figure 4.2.

CHAPTER 4 FIGURES AND TABLES

Table 4.1. Classification of SPARTICUS missions as a function of formation mechanism. All dates are in 2010.

Formation Mechanism	**Mission**
Synoptic	Jan 12ab, 14ab, 15, 19ab, 20ab, 21, 23, 26, 27, 31 Feb 1, 3ab, 4, 10abc, 11ab, 17ab, 19ab, 22 Mar 10abc, 14ab, 15ab, 17ab, 19, 23abc, 26, 27abc, 30abc Apr 1ab, 2, 5, 6abc, 11ab, 12, 14a, 16ab, 17abc, 19ab, 28ab, 29 Jun 7, 12b, 17, 18ab, 24bc July 23
Convective	Jan 21 Apr 14b, 22ab, 23, 24ab Jun 4ab,11ab, 12a, 14, 15ab, 24a
Orographic	Mar 22, Apr 11b, Jun 2

Table 4.2. Table of *m-D* relations used to calculate *IWC* using Equation (3).

Habit	α [g cm$^{-\beta}$]	β	Reference
1. Sphere	0.91* π /6	3	
2. Column	D < 30 μm 0.91*π/6 30 < D < 300 μm 0.001666 D ≥ 300 μm 0.000907	D < 30 μm 0.91*π/6 30 < D < 300 μm 1.91 D ≥ 300 μm 1.74	*Mitchell* [1996] – "Hexagonal Columns"
3. Plate	0.00739	2.45	*Mitchell* [1996] – "Hexagonal Plates"
4. Stellar	D < 90 μm 0.00583 D ≥ 90 μm 0.000270	D < 90 μm 2.42 D ≥ 100 μm 1.67	*Mitchell* [1996] – "Stellar crystal with broad arms"
5. Dendrite	D < 100 μm 0.00583 D ≥ 300 μm 0.000012	D < 100 μm 2.42 D ≥ 300 μm 1.80	*Mitchell* [1996] – "Broad branched crystal"
6. Rosette and 7. Budding Rosette	D < 90 μm 0.000012 D ≥ 90 μm 0.00308	D < 90 μm 1.52 D ≥ 100 μm 2.26	*Mitchell* [1996] – "Bullet rosettes, 5 bullets"
8. Small irregular and 9. Big irregular	D < 100 μm look-up table based on Gaussian random sphere D ≥ 100 μm 0.00294	D < 100 μm look-up table based on Gaussian random sphere D ≥ 100 μm 1.9	*Nousainen and McFarquhar* [2004] and *Brown and Francis* [1996]

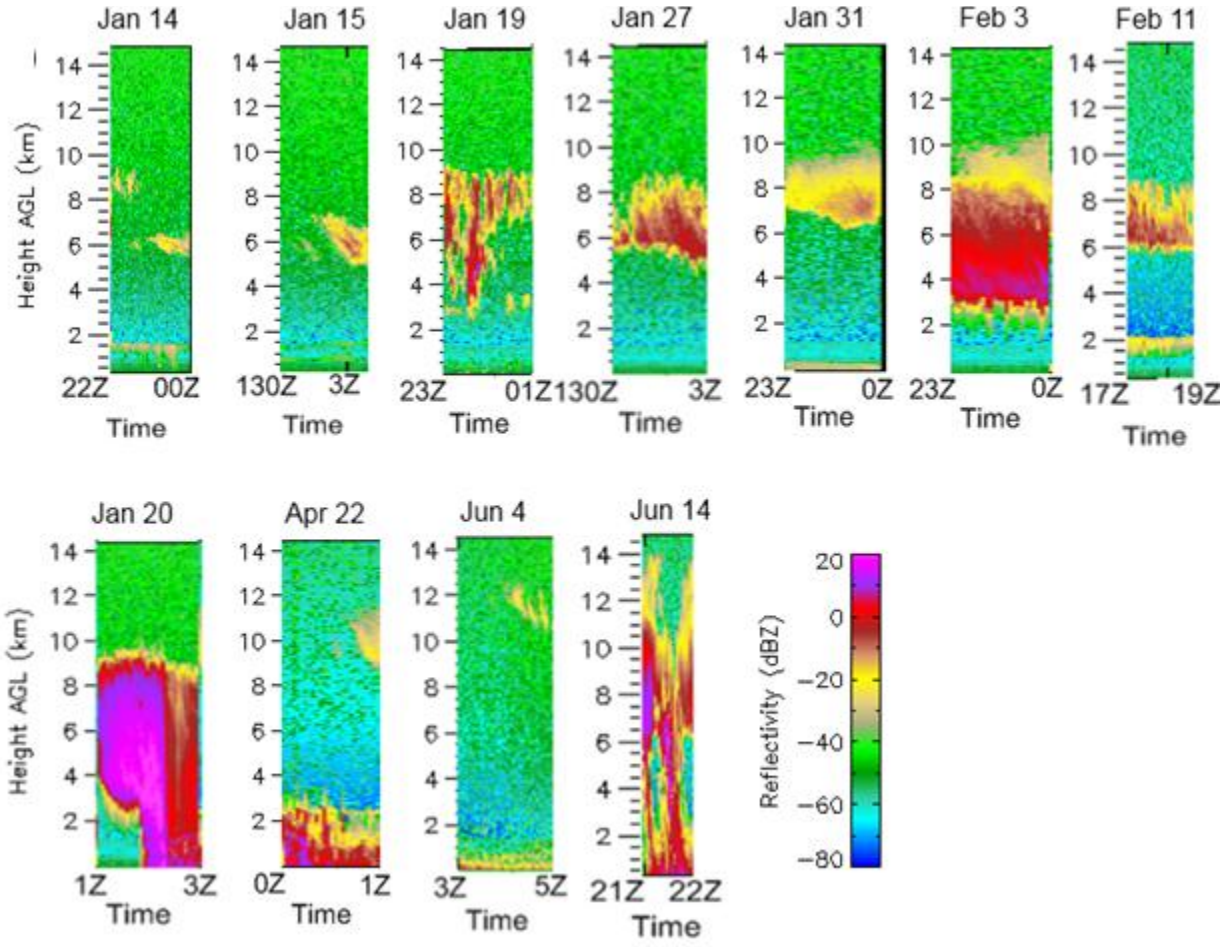


Figure 4.1. Example images of radar reflectivity as a function of UTC time from the Millimeter Cloud Radar at the SGP site. Radar images taken from the SPARTICUS wiki from the Aerial Airborne Facility archive.

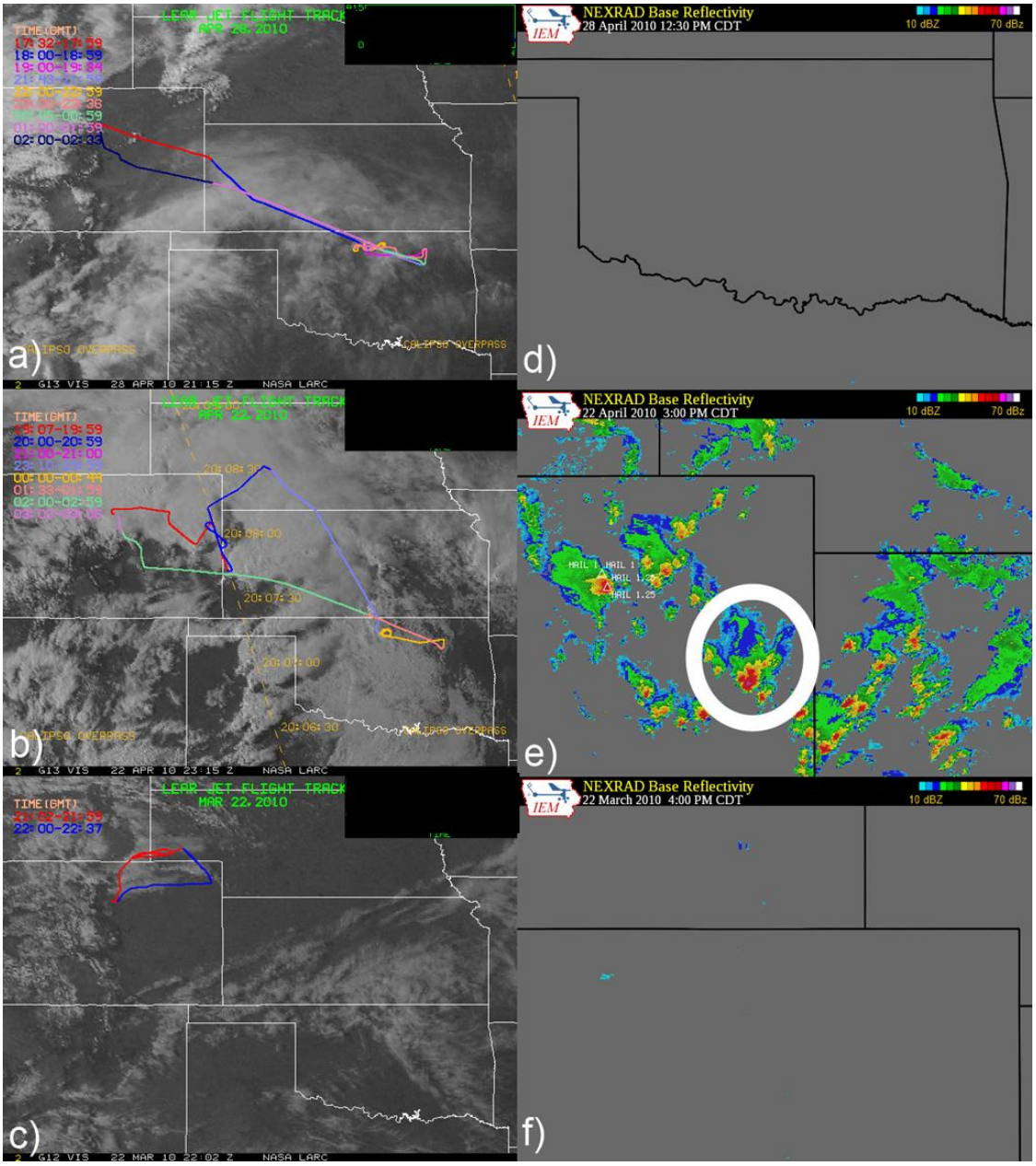


Figure 4.2. Flight tracks overlaid over GOES visible satellite imagery for the Apr 28 2010 mission sampling synoptic cirrus **(a)**, Apr 22 2010 convective mission **(b)** and Mar 22 2010 orographic mission **(c).** Images obtained from Patrick Minnis' and Iowa State University of Science and Technology software mentioned in the acknowledgments section.

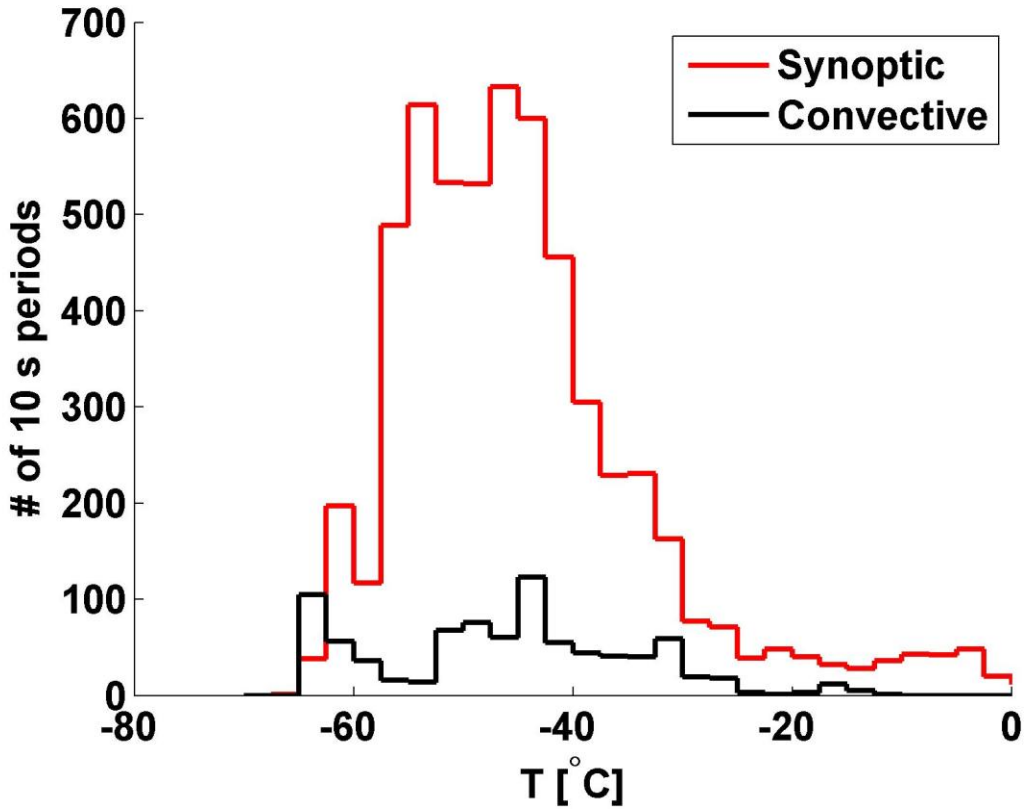


Figure 4.3. Frequency distribution of *T* in 10 s time periods for the synoptically and convectively generated cirrus during SPARTICUS.

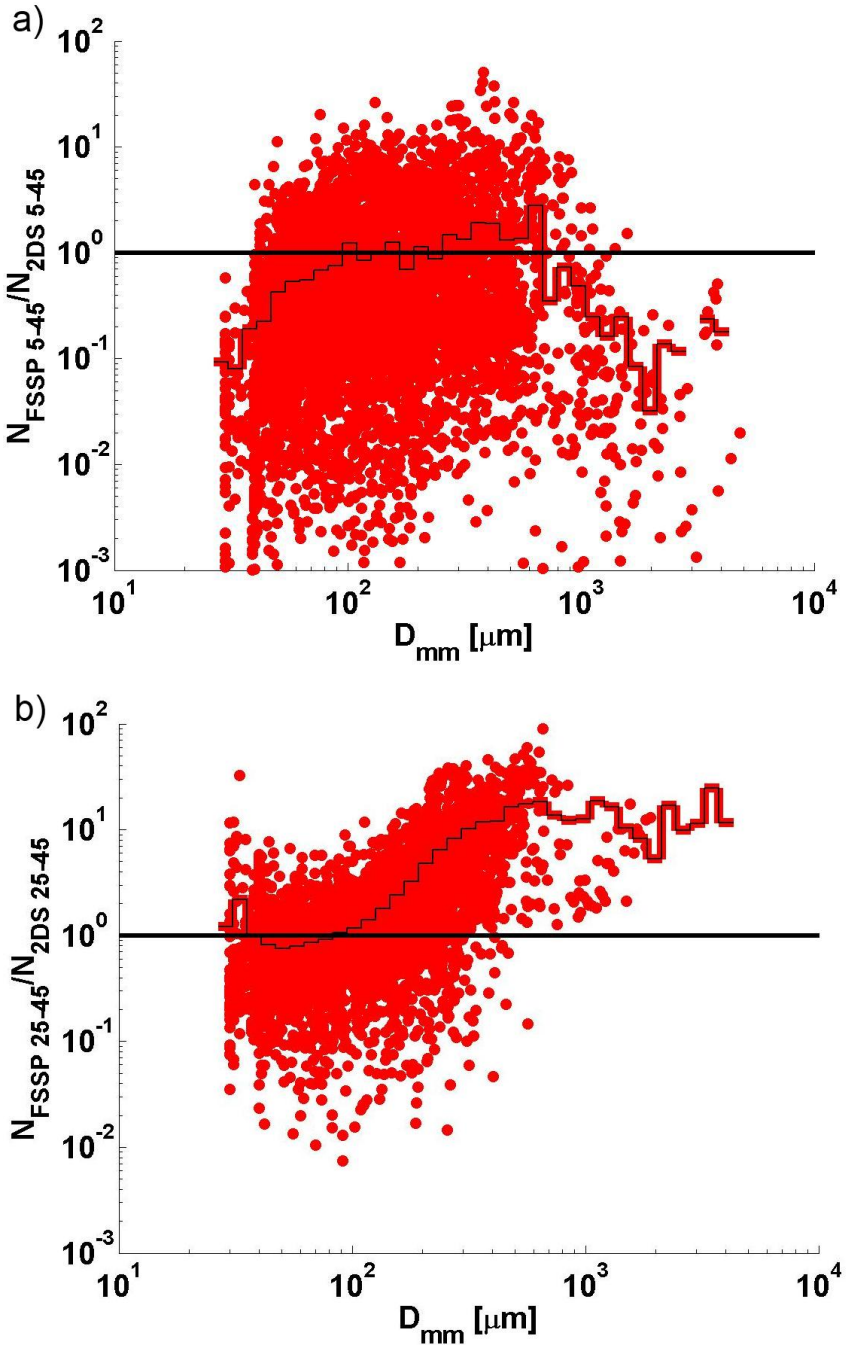


Figure 4.4. 10 s averages of **(a)** $N_{FSSP\ 5\text{-}45}/N_{2DS\ 5\text{-}45}$ and **(b)** $N_{FSSP\ 25\text{-}45}/N_{2DS\ 25\text{-}45}$ as a function of D_{mm}. Shattered artifact removal algorithms are applied to both 2DS and Fast FSSP from all flight days.

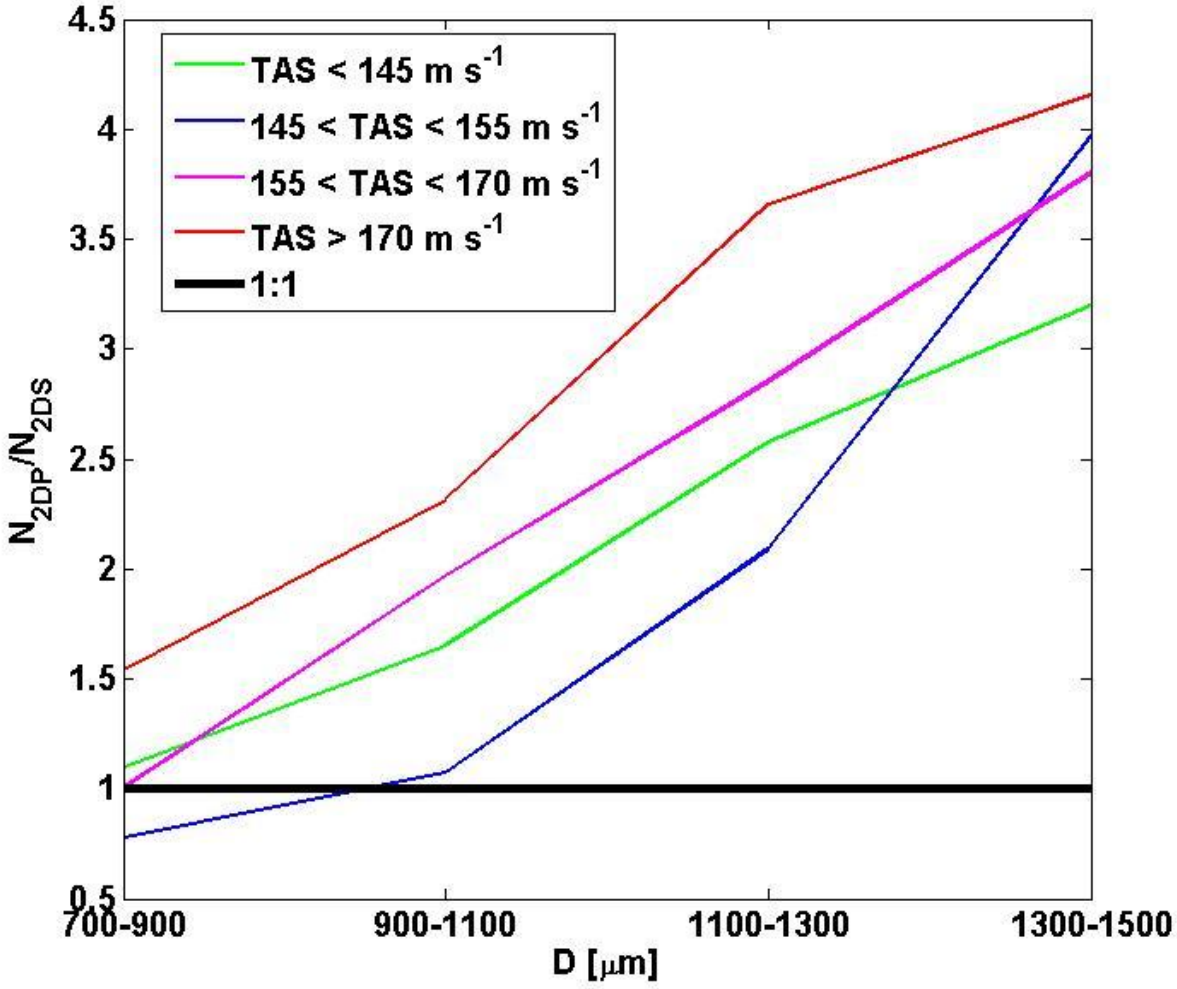


Figure 4.5. Mean N_{2DP}/N_{2DS} as a function of D for the differing true airspeeds TAS.

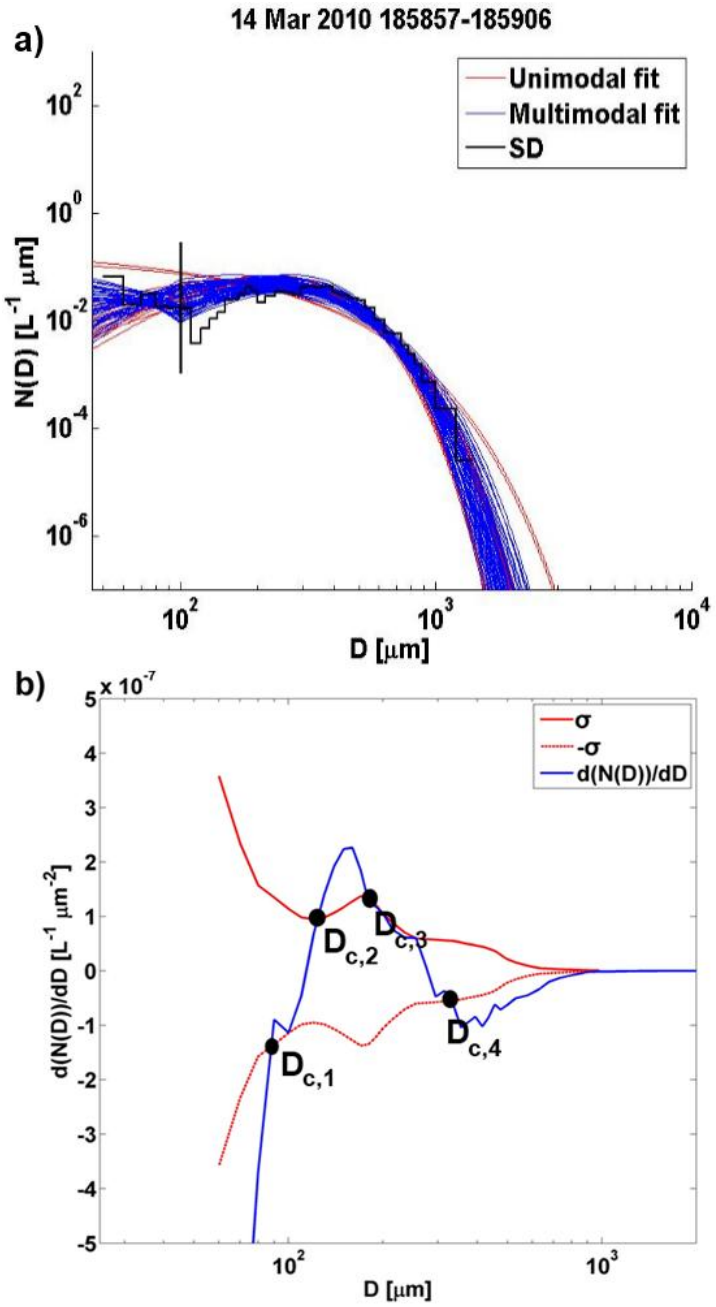


Figure 4.6. Conceptual figure of automated technique to find modes applied to the mean *N(D)* from 14 Mar 2010 185857-185906 UTC. **(a)** Mean *N(D)*. Black line = boundary between modes. Red curves represent equally realizable unimodal gamma fits to *N(D)* while blue curve represents multimodal gamma fits. **(b)** d(N(D)/dD and σ as a function of D for the *N(D)* shown in **(a)**.

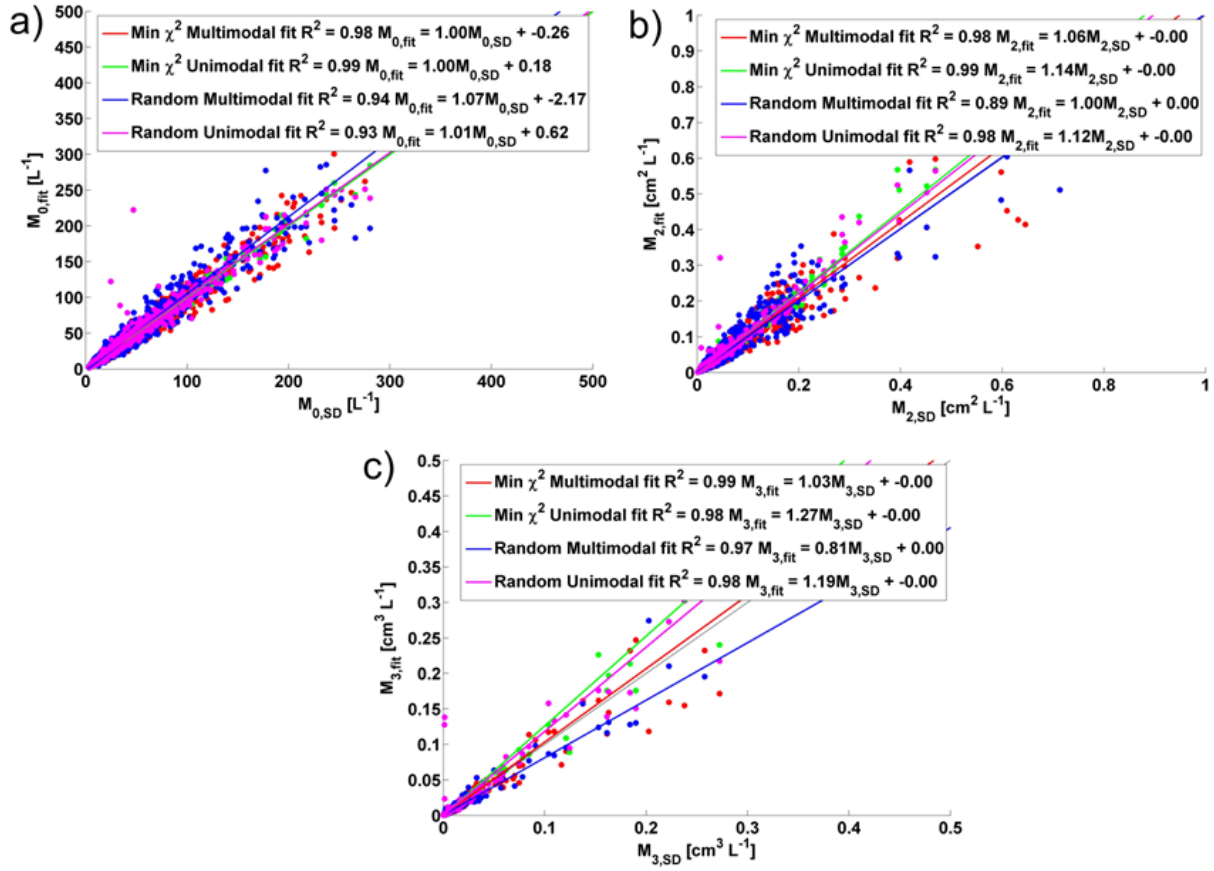


Figure 4.7. (a) $M_{0,fit}$ as a function of $M_{0,obs}$. **(b)** $M_{2,fit}$ as a function of $M_{2,obs}$. **(c)** $M_{3,fit}$ as a function of $M_{3,obs}$. $M_{x,fit}$ derived using the (N_0, μ, λ) triplet that minimizes χ^2 in red and green, and by randomly choosing a (N_0, μ, λ) triplet from the volume of equally realizable solutions in red and blue, Lines represent the linear best fit of $M_{x,fit}$ as a function of $M_{x,obs}$. Only data from time periods where 2 or more modes were identified in the composite SD are included in Figure 6.

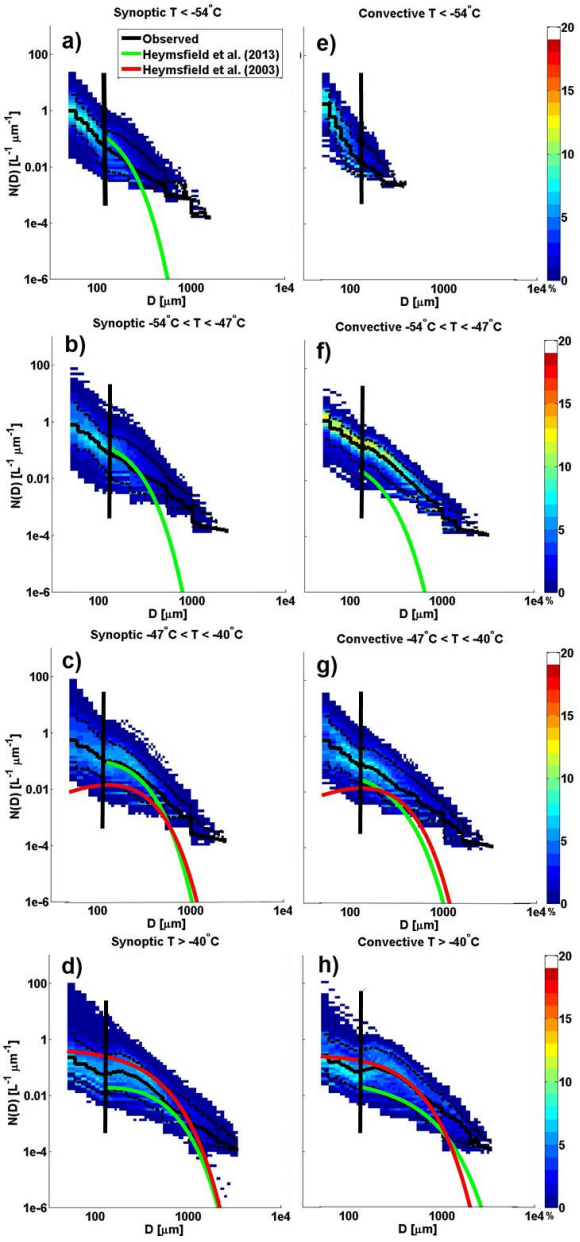


Figure 4.8. Normalized distribution of *N(D)* (per cent) for all ice clouds during SPARTICUS when **(a)** T < -54°C, **(b)** -54°C < T < -47°C, **(c)** -47°C < T < -40°C, **(d)** T > -40°C. Solid black horizontal line denotes median, dashed lines denote 10th and 90th percentiles of *N(D)*. Solid colored lines show the Heymsfield et al. (2003a; 2013) curves. Solid vertical black line shows mean location of boundary between first and second mode. Colors represent normalized frequency.

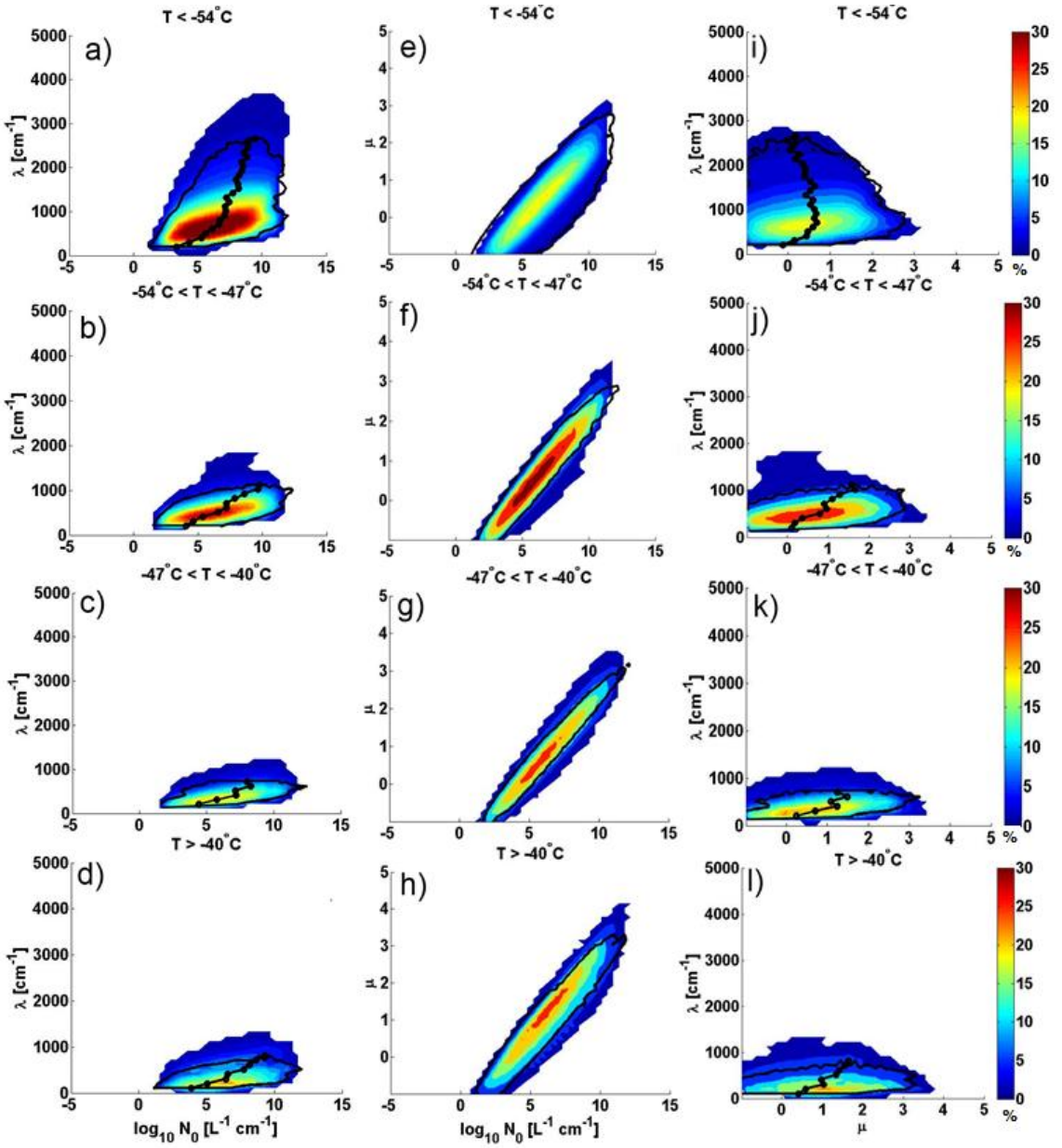


Figure 4.9. Normalized occurrence of given point in (N_0, λ) phase space (in percent) inside ellipsoids of equally realizable fits for SDs with one mode when **(a)** T < -54°C, **(b)** -54°C < T < -47°C, **(c)** -47°C < T < -40°C, **(d)** T > -40°C. **(e,f,g,h)** as **(a,b,c,d)** but in (N_0, μ) phase space. **(i,j,k,l)** as in **(a,b,c,d)**, but in (λ, μ) phase space. The black curve represents the 2-d projection of the surfaces in the given phase space. The dashed dotted line represents the center of the ellipse of equally realizable (N_0, μ) for a given λ.

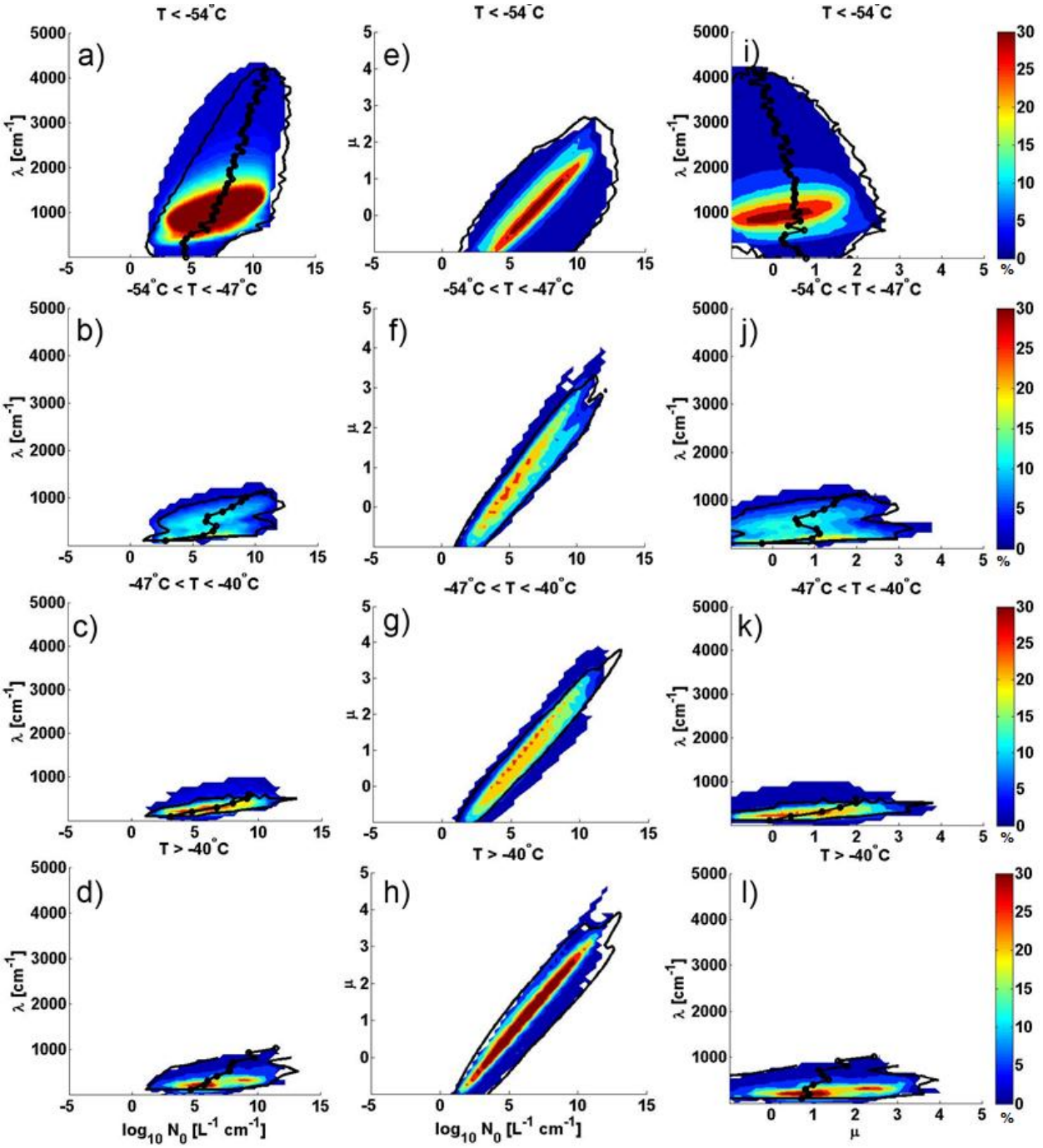


Figure 4.10. As Fig. 4.8, but for convectively generated cirrus.

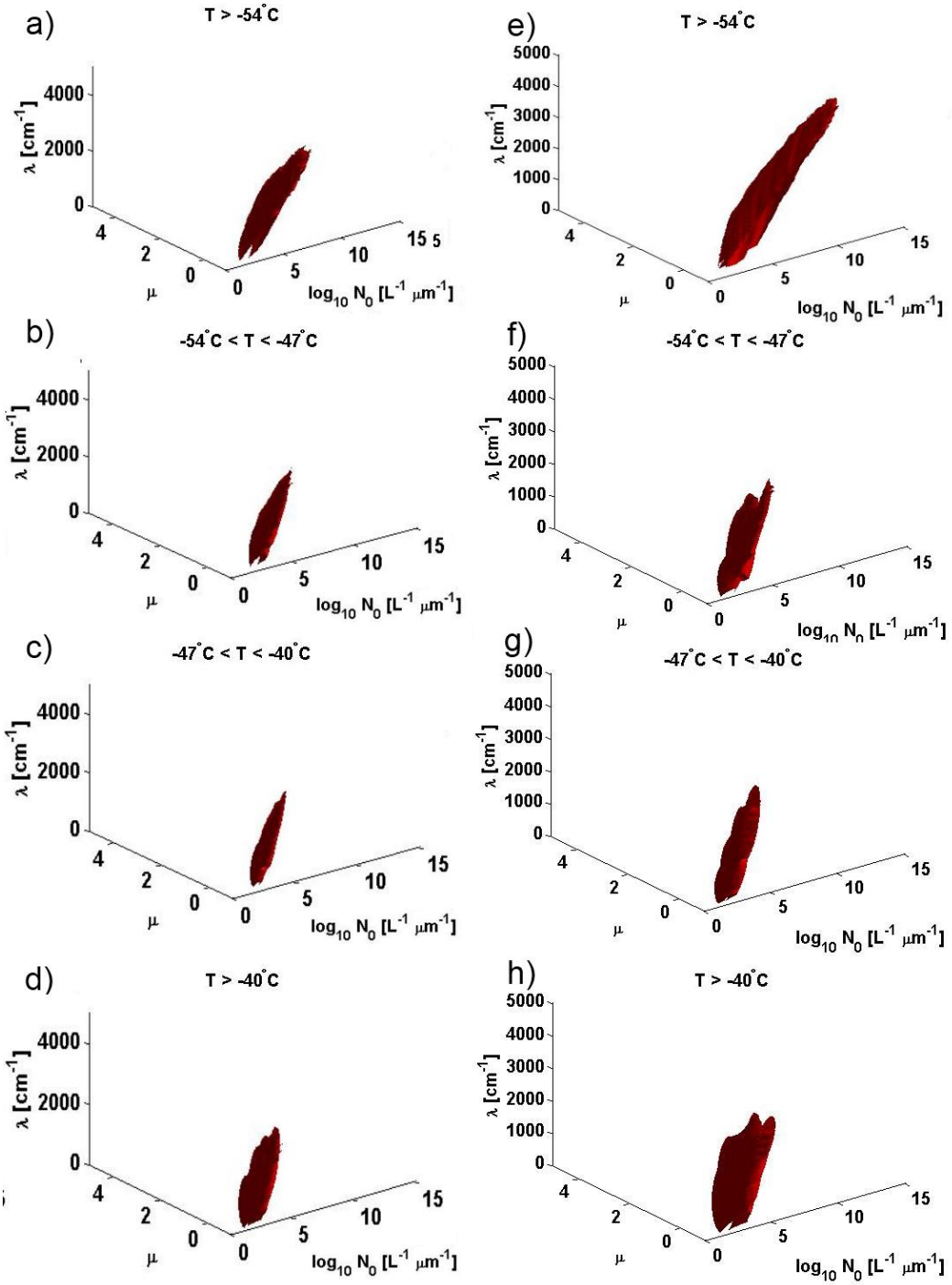


Figure 4.11. Volumes in (N_0, λ μ) phase space representing the family of SDs in for **(a)** T < -54°C, **(b)** -54°C < T < -47°C, **(c)** -47°C < T < -40°C, **(d)** T > -40°C for synoptically generated cirrus. (**e,f,g,h)** as in **(a,b,c,d)**, but for convectively generated cirrus.

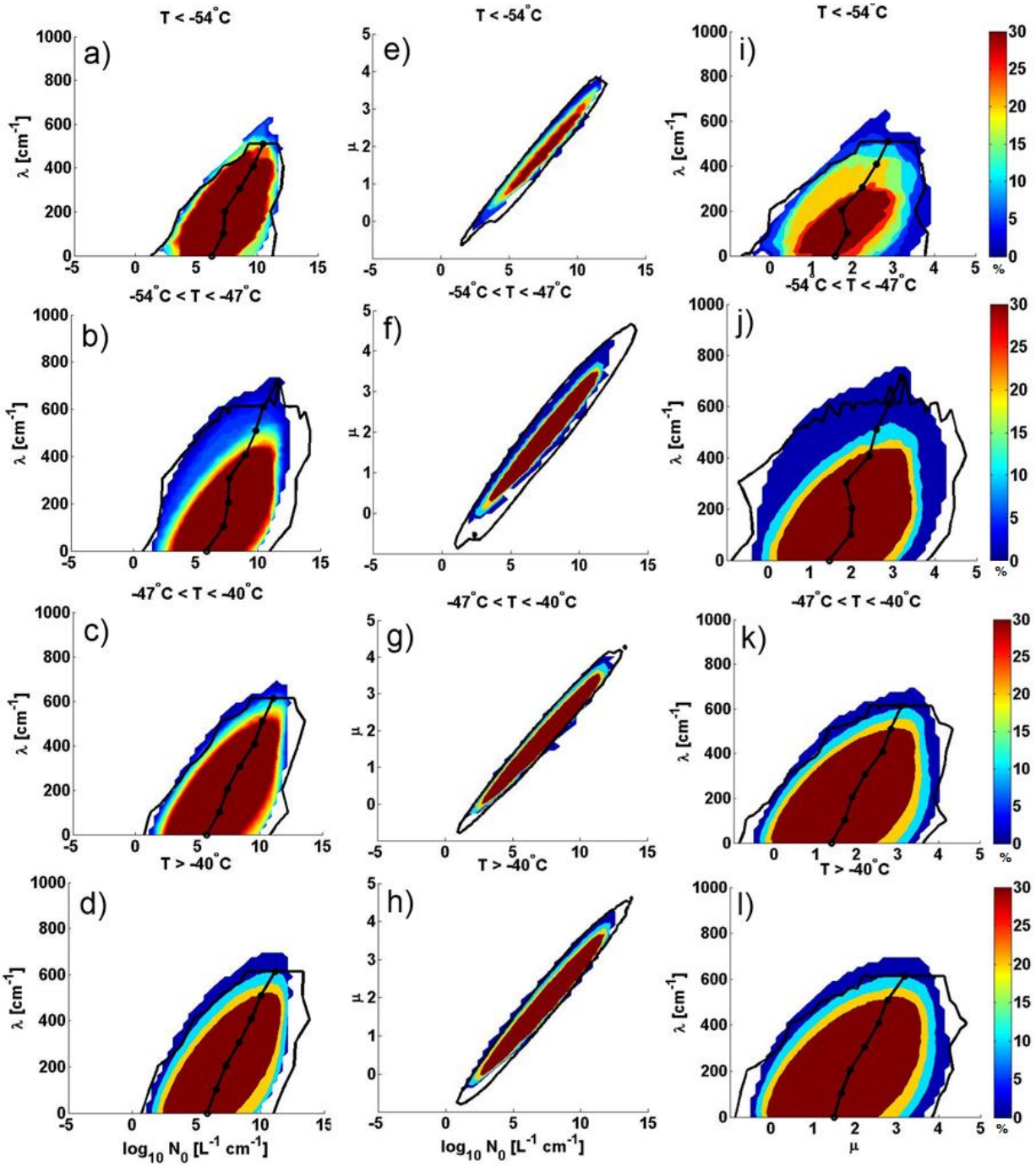


Figure 4.12. As in Figure 4.8, but for the mode with smaller *D* of the bimodal SDs in synoptic cirrus.

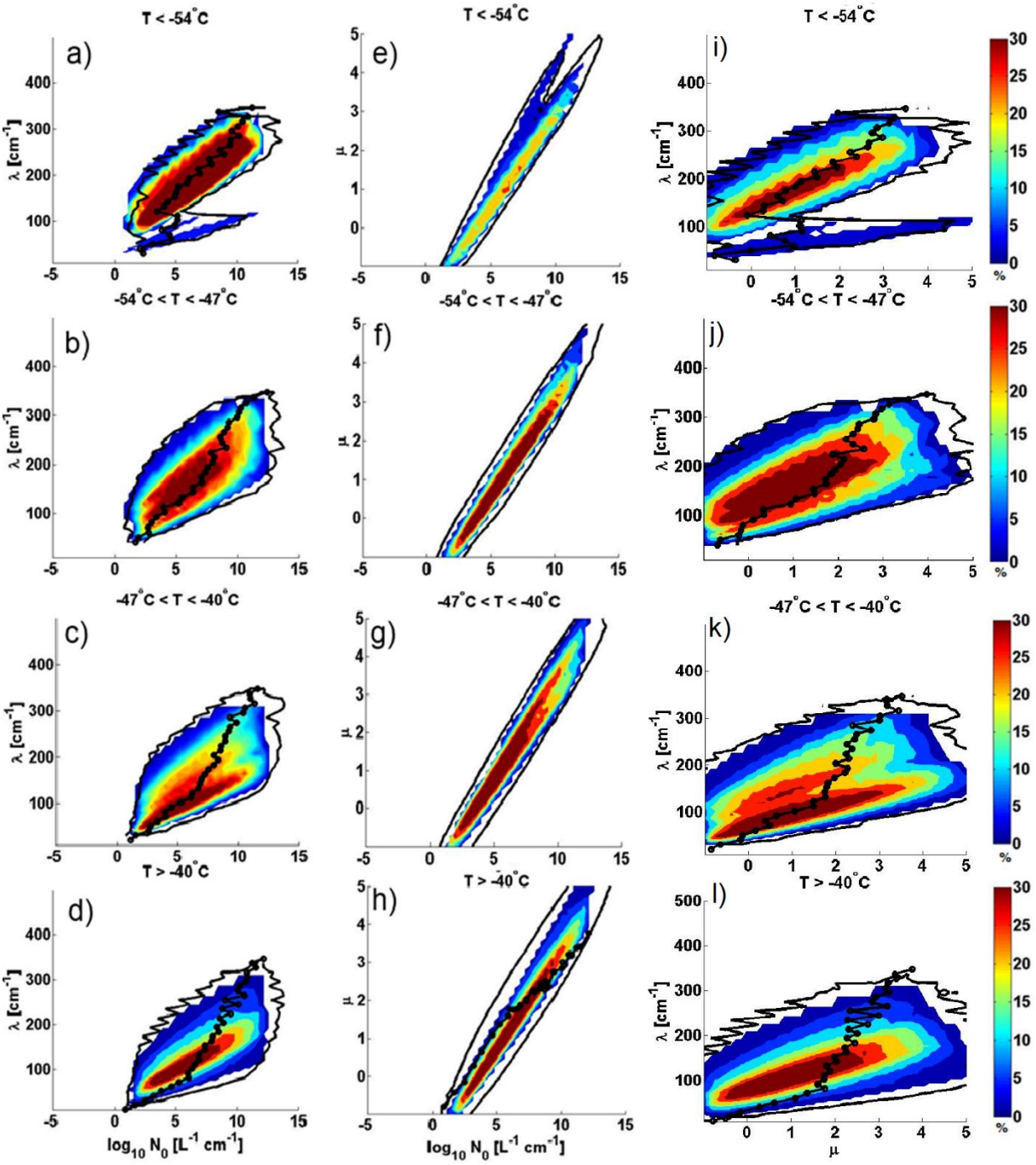


Figure 4.13. As in Figure 4.8, but for the mode with larger *D* of the bimodal SDs in synoptic cirrus.

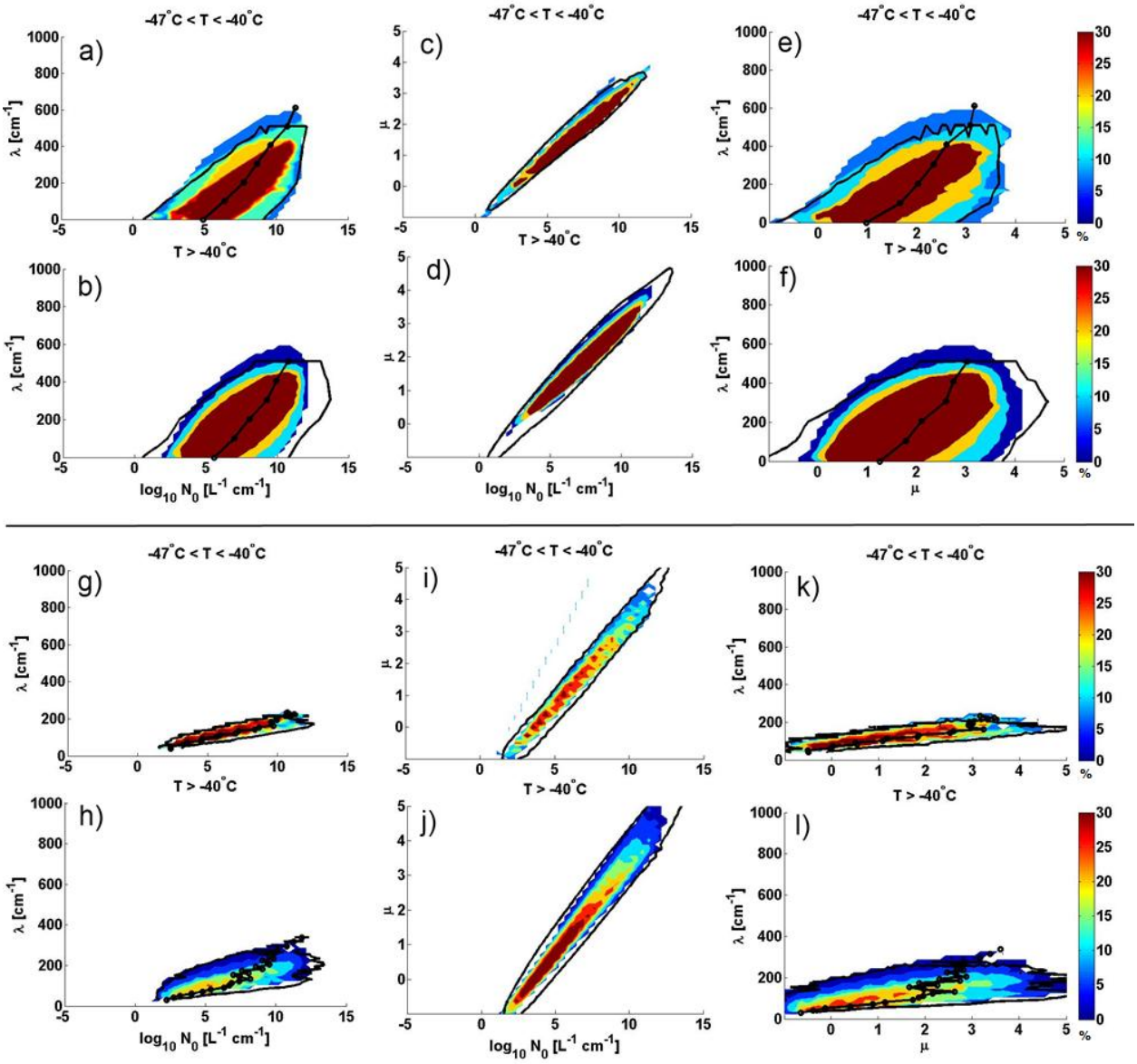


Figure 4.14. (a-f) As in Figure 4.8, but for the mode with smaller *D* of the bimodal SDs in convectively generated cirrus. **(g-l)** As in Figure 8, but for the mode with larger *D* of the bimodal SDs in convectively cirrus.

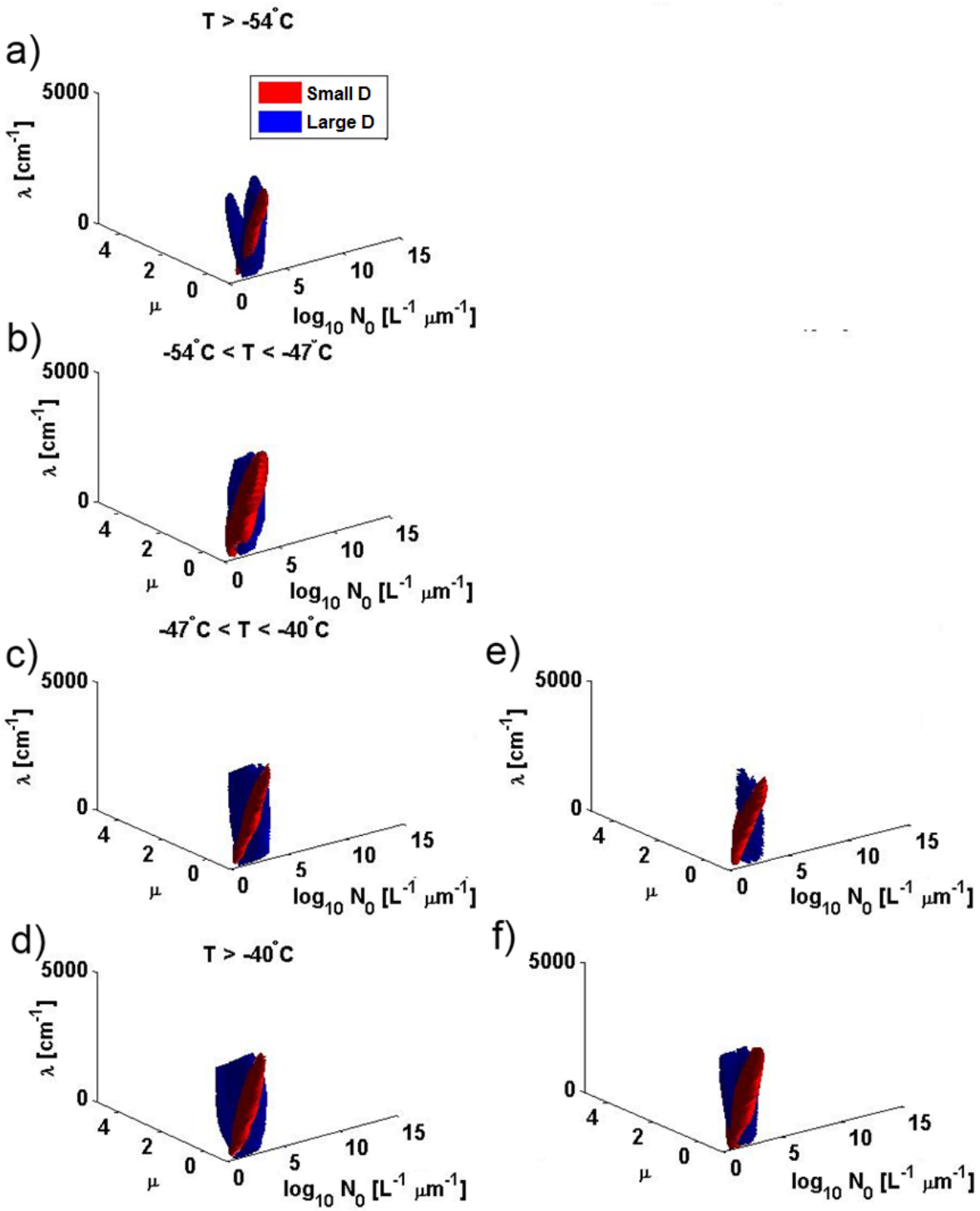


Figure 4.15. As Fig. 4.10, but for time periods when two modes were identified in the SD. Panels **(a-d)** represent synoptic cirrus, and **(e-f)** represent convectively generated cirrus.

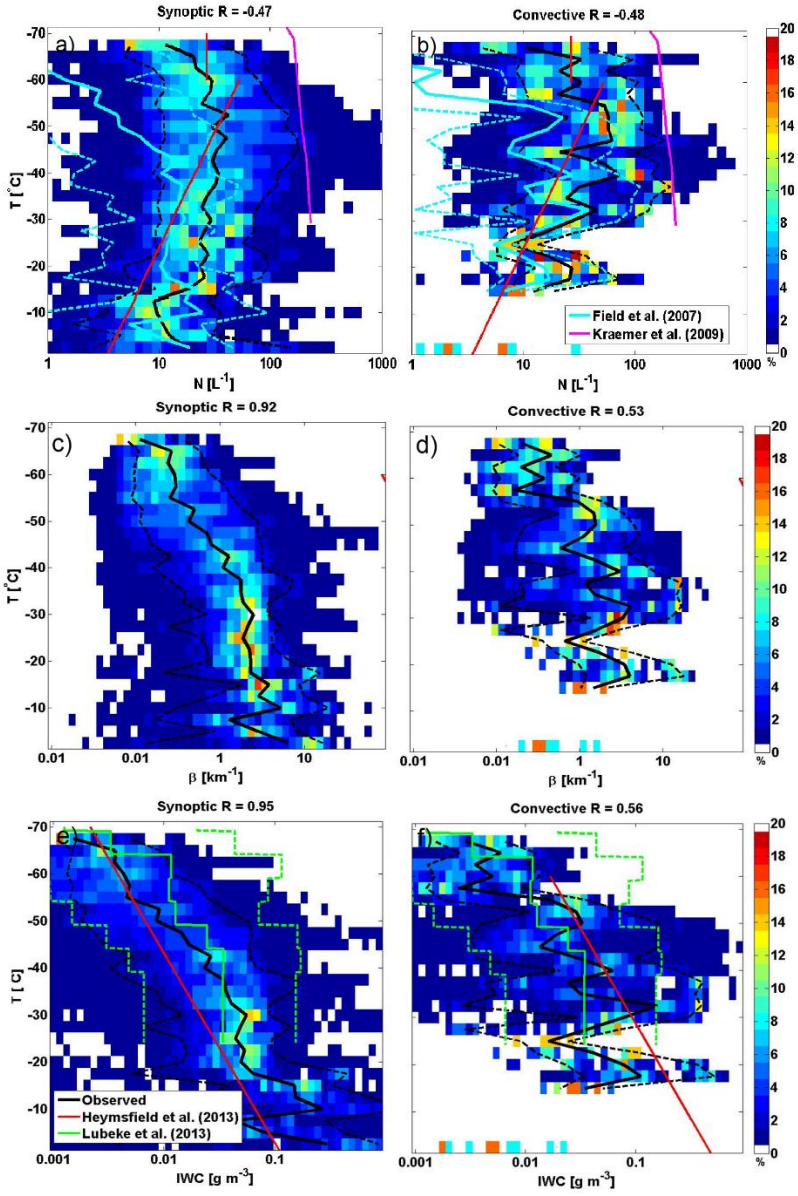


Figure 4.16. Frequency of *N* normalized by *T* for **(a)** synoptically generated cirrus and **(b)** convectively generated cirrus. **(c,d)** as in **(a,b)** except for β instead of *N*. **(e,f)** as in **(a,b)** except for *IWC* instead of *N*. Solid line represents the median for a given *T* range, and the dashed lines represent the 10th and 90th percentiles. Red line shows parameterization from Heymsfield et al. (2013). Green lines represent the 10th, 50th and 90th percentiles from Lubeke et al. (2013).

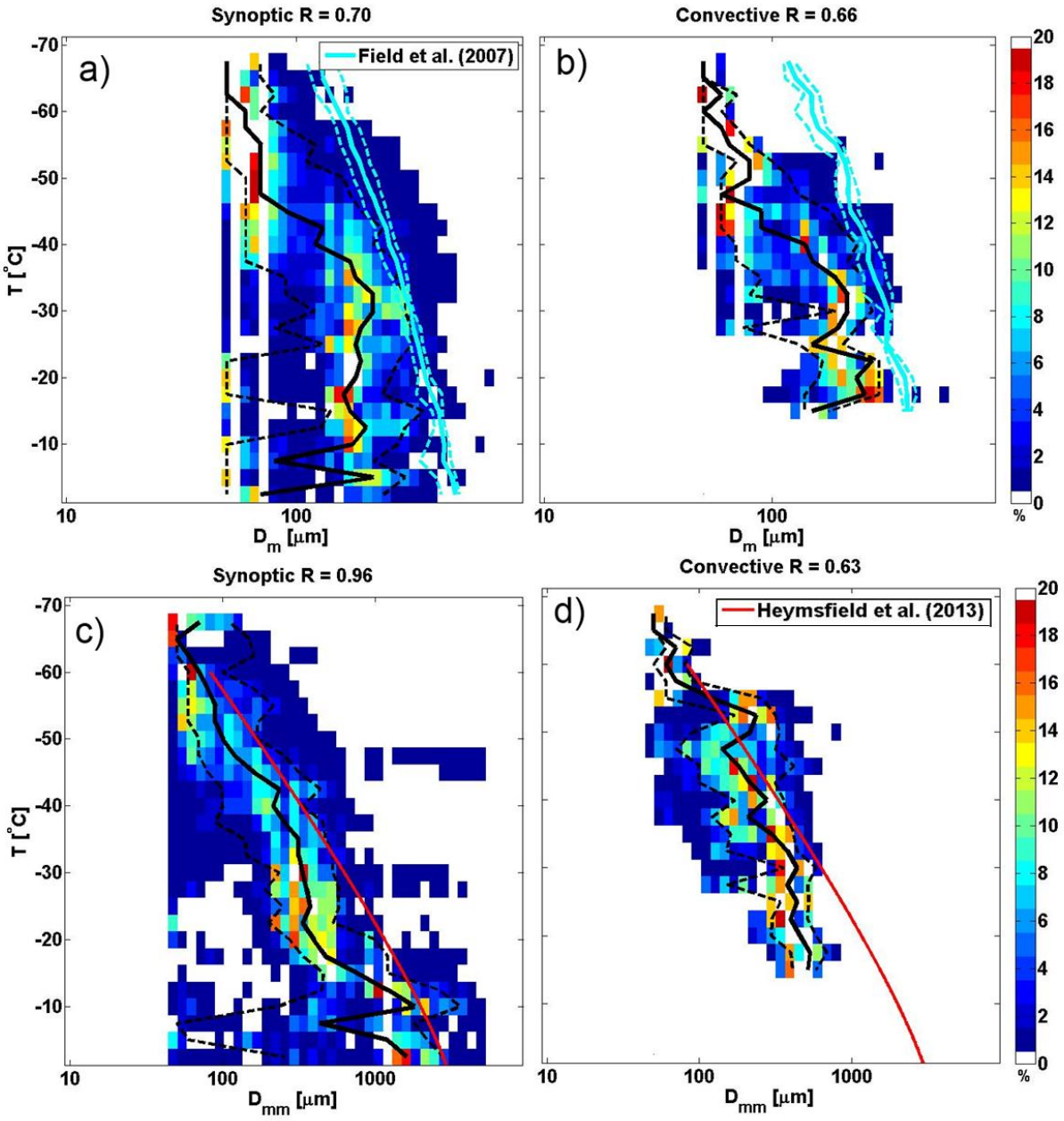


Figure 4.17. Frequency of D_m normalized by T for **(a)** synoptically generated cirrus and **(b)** convectively generated cirrus. **(c,d)** as in **(a,b)** except for D_{mm} instead of D_m. Solid line represents the median for a given T range, and the dashed lines represent the 10th and 90th percentiles. Red line shows parameterization from Heymsfield et al. (2013).

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1. Summary

Ice clouds have significant impacts on the Earth's radiative budget. Their radiative impact highly depends on ice cloud microphysical properties. Climate and weather prediction models have to make certain assumptions about how the various processes are represented. Observations of how cloud properties vary with environmental conditions can help evaluate some parameterizations used in models. However, sufficient data are not available to characterize how ice crystal properties vary as a function of environmental conditions. Furthermore, many of these assumptions are derived from historical datasets collected by in situ probes, namely optical array probes that can be contaminated by shattered artifacts generated by large particles shattering on the probe tips and inlets.

There were thus two main objectives of this dissertation. The first was to quantify the uncertainty in 2DC ice measurements due to shattering and to determine the optimum way to reduce the bias due to shattering in microphysical properties derived from 2DCs. This was done by analyzing data from two different field experiments that sampled ice clouds: the Indirect and Semi-Direct Aerosol Campaign (ISDAC) conducted in 2008, and the Instrumentation Development and Education in Airborne Science project (IDEAS-2011) conducted in 2011. One flight during ISDAC and two flights during IDEAS-2011 with modified 2DCs with tips designed to deflect remnants of shattered large crystals away from the sample volume flown immediately adjacent to conventional 2DCs with standard hemispheric tips were analyzed. For these flights, the impact of the redesigned tips and processing algorithms designed to remove such shattered particles (Korolev and Isaac 2005; Field et al 2006) on the measured number distribution

function *N(D),* and bulk cloud properties such as the bulk extinction β, ice water content *IWC*, asymmetry parameter g, single scatter albedo ω_0, median diameter D_m, median mass diameter D_{mm}, effective radius r_e, and mass weighted fall speed v_m was determined.

The second main objective was to examine how mid-latitude cirrus microphysical properties varied as a function of temperature and formation mechanism using data collected during the Small Ice Particles in Cirrus (SPARTICUS) project. During SPARTICUS, the Stratton Park Engineering Company (SPEC) Learjet, using probes with anti-shattering tips, collected 190 h of data, thus representing an unprecedented amount of data collected using the newly designed probe tips. A more complete uncertainty analysis of the representation of functional forms of *N(D)* than performed in past studies was conducted. In particular, a novel representation of the observed *N(D)* as a gamma distribution $N(D) = N_0 D^{\mu} e^{-\lambda D}$ was derived from the SPARTICUS data, whereby a volume of equally realizable solutions in (N_0, μ, λ) phase space was created using the incomplete gamma fitting technique of McFarquhar et al. (2014). It was subsequently determined how these volumes varied as a function of environmental parameters.

5.2. Principal conclusions

There were several principal conclusions of this study which are briefly summarized here. In Chapter 2, Monte Carlo simulations of the shattering of large particles on probe tips were conducted in an attempt to explain the differences between the observed arrival times between particles in the 2DC sample volumes recorded during ISDAC and IDEAS-2011. These simulations showed that the differences in the interarrival times in the mode corresponding to natural particles were likely due to the different number concentrations of natural particles observed, differences in the sample volumes of the 2DCs used, and different airspeeds during

ISDAC and IDEAS-2011. The differences between the interarrival times in the mode corresponding to shattered particles were likely due the differences in how the particles shattered on the 2DC probe tips, due to either variations in habits or the geometries of the probes used. Differences in the number of artifacts observed during IDEAS-2011 and ISDAC were associated with the observed differences in natural number concentration, 2DC sample volumes, and airspeed, as well as the details of the shattering process.

The analysis in Chapter 2 also showed that particles with *D* as large as 500 μm could be produced by the shattering of large crystals on probe tips and the ratios of shattered particles to naturally occurring particles were correlated with median mass diameter as well as the presence of graupel. For IDEAS-2011, the modified tips removed as many shattered artifacts as the use of the anti-shattering algorithm, while for ISDAC the algorithms removed fewer artifacts than the tips. In general, both techniques are required to best mitigate shattered artifacts.

Since it is not known whether the modified tips and processing algorithms combined remove all shattered artifacts, the *N(D)* from the two 2DC probes used during IDEAS-2011 were compared against the *N(D)* measured by the Holographic Detector for Clouds (HOLODEC) which derives *N(D)* from reconstructed holograms (Fugal et al. 2003). The comparison showed that *N(D)* from the HOLOLDEC and from the 2DC with modified tips and processed to remove shattered artifacts were comparable for the limited time periods examined. Because the two probes use very different mechanisms to derive *N(D)* and because the HOLODEC permits easy identification of shattered artifacts from identification of particles' positions in a three dimensional volume, the agreement suggests that the modified tips and processing algorithms remove the majority artifacts from the 2DC data.

The analysis in Chapter 3 focused on the impact of anti-shatter tips and shattered artifact removal algorithms on bulk properties derived from 2DC ice crystal size distributions. The bulk extinction β, ice water content *IWC*, effective radius r_e, median mass diameter D_{mm}, mass weighted terminal velocity v_m, asymmetry parameter g and single scatter albedo ω_0 changed by less than 25% when either shattered artifact removal algorithms or modified tips were used compared to when neither were used. However, there was a 60% change in *IWC*, and D_{mm} when the assumed relationship between mass and *D* was changed, showing shattering is not the largest uncertainty in derivation of these quantities. Meanwhile, D_m increased by an average of 43% when modified tips were used in place of standard tips. Therefore, the analysis in Chapters 2 and 3 indicates that quantities derived from lower order moments of the 2DC size distribution (i.e. *N*, *N(D)*, D_m) are contaminated by shattered artifacts much more than quantities derived from higher order moments of the 2DC size distribution (i.e. β, *IWC*, r_e, D_{mm}, v_m, g). This implies that historical parameterizations using β, *IWC*, r_e, D_{mm}, v_m, g derived from data collected by 2DCs with standard tips may still be acceptable to use provided that the accuracy needed in the application of these parameterizations are within the differences induced by shattering listed in Chapters 2 and 3. However, quantities based on lower order moments (i.e. *N*, *N(D)*, D_m) derived from data collected by 2DCs with standard tips are likely problematic due to contamination from shattered artifacts.

The analysis presented in Chapter 4 gave representations of *N(D)* observed during the Small Particles In Cirrus (SPARTICUS) experiment as representations of one or two as gamma distributions given by $N(D) = N_0 D^{\mu} e^{-\lambda D}$, with an automated technique determining the division between the modes. The fit parameters were given by volumes of equally realizable solutions in (N_0, μ, λ) phase space. Using data collected in SPARTICUS in a wide variety of

meteorological conditions, the dependence of the bulk parameters N, β, IWC, D_m, and D_{mm} on environmental conditions was determined and compared against data collected in cirrus during recent campaigns. Similar to the findings of McFarquhar et al. (2014) and others, there was a strong codependence between gamma fit parameters. The codependence changed with temperature T in a manner consistent with the presence of larger particles at higher T, and the increase in β, IWC, D_m, and D_{mm} with T. This was consistent with more efficient particle growth at higher T.

There were some observed differences between the two modes of the SDs. The boundary between the two modes was, on average, at 115 μm, similar to previous observations of bimodal SDs. Furthermore, no more than 2 modes were identified in the SDs observed during SPARTICUS, and the frequency of the bimodality increased with increasing T, consistent with hypothesis of Zhao et al. (2010) that heterogeneous nucleation may occur in the presence of falling particles. The differences between the second and first mode of the SDs were consistent with the findings of Mitchell et al. (1996) who hypothesized that particles in the small particle mode grow by vapor deposition only, and particles in the large particle mode by both vapor deposition and aggregation.

5.3. Future studies

For any future studies examining how shattered artifacts affect the derivation of $N(D)$ and bulk cloud parameters, the investigations should be conducted with a variety of probes with different response times and configurations and data should be obtained in a wider variety of cloud conditions because inevitably a limited set of conditions were sampled during ISDAC and IDEAS-2011. The results of this paper cannot necessarily be generalized to other optical array probes, probe installations, aircraft, or sampled environmental conditions. Future studies should

concentrate on acquiring data in ice clouds at colder temperatures (e.g. < -40°C), in a wider range of aircraft operating parameters (e.g. true air speeds above 150 m s^{-1}), in a wider range of crystal habits (e.g. more rimed particles and graupel), using different probe installations (e.g. on opposite wings of the aircraft)), using different aircraft (i.e. Wyoming King Air, NCAR G-V), with a wider variety of probes (e.g. two-dimensional stereo probe, cloud imaging probe), and using different designs of the probe tips for deflecting shattered particles from the probe sample volume (e.g. Korolev et al. 2013a).

Finally, even though the SPARTICUS analysis presented in Chapter 4 represents the largest data set of mid-latitude cirrus sampled by probes with shatter mitigating tips in the largest variety of environmental conditions across seasons, there is still a much wider variety of conditions that must be sampled to determine what controls measured ice crystal properties. In particular, data covering tropical and arctic cirrus are especially lacking.

References

Ackerman, T.P., K-N. FP Liou, J. Valero, and L. Pfister 1988: Heating rates in tropical anvils, *J. Atmos. Sci.* **45**, 1606-1628, doi: 10.1175/1520-0469(1988)045<1606:HRITA>2.0.CO;2

Atlas, D., S.Y. Matrosov, A.J. Heymsfield, M. Chou, and D.B. Wolff, 1995: Radar and radiation properties of ice clouds, *J. Appl. Metorol.*, **34**, 2329-2345.

Austin, R. T., A. J. Heymsfield, and G. L. Stephens, 2009: Retrieval of ice cloud microphysical parameters using the CloudSat millimeter-wave radar and temperature, *J. Geophys. Res.*, **114**, D00A23, doi: 10.1029/2008JD010049

Avramov, A., and J. Y. Harrington, 2010: Influence of parameterized ice habit on simulated mixed-phase Arctic clouds. *J. Geophys. Res.*, **115**, D03205, doi: 10.1029/2009JD012108

Baker, B.A., and R.P. Lawson 2006: Improvement in determination of ice water content from two-dimensional particle imagery: Part I: Image to mass relationships, *J. Appl. Meteorol.*, *45*(9), 1282-1290

Baum, B., D. Kratz, P. Yang, S. Ou, Y. Hu, P. Soulen, and S.-C. Tsay, 2000: Remote sensing of cloud properties using MODIS airborne simulator imagery during SUCCESS: I. Data and models, *J. Geophys. Res.*, **105**, 11,767–11,780. doi: 10.1029/1999JD901089

Baum, B. A., A.J. Heymsfield, P. Yang, and S.T. Bedka, 2005a: Bulk scattering models for the remote sensing of ice clouds. Part 1: Microphysical data and models. *J. Appl. Meteor.*, **44**, 1885-1895.

Baum, B. A., P. Yang, A. J. Heymsfield, S. Platnick, M. D. King, Y.X. Hu, and S. T. Bedka, 2005b: Bulk scattering properties for the remote sensing of ice clouds. Part II: narrowband models. *J. Appl. Meteor.*, **44**, 1896–1911.

Baum, B. A., P. Yang, S. L. Nasiri, A. K. Heidinger, A. J. Heymsfield, and J. Li, 2007: Bulk scattering properties for the remote sensing of ice clouds. Part 3: high resolution spectral models from 100 to 3250 cm^{-1}. *J. Appl. Meteor. Clim.*, **46**, 423-434.

Baum, B.A., and 7 co-authors, 2011: Improvements in shortwave bulk scattering and absorption models for the remote sensing of ice clouds, *J. Appl. Meteorol.*, **50**, 1037-1056.

Baumgardner, D., and A. Korolev, 1997: Airspeed Corrections for Optical Array Probe Sample Volumes. *J. Atmos. Oceanic Technol.*, **14**, 1224–1229

Beswick, K.M., M. W. Gallagher, A.R. Webb, E.G. Norton, and F. Perry, 2008: Application of the Aventech AIMMS20AQ airborne probe for turbulence measurements during the Convective Storm Initiation Project, *Atmos. Chem. Phys.*, **8**, 5449-5463, doi: 10.5194/acp-8-5449-2008

Boudala, F.S., G.A. Isaac, Q. Fu, and S.G. Cober, 2002: Parameterization of effective ice particle sizes for high-latitude clouds, *Int. J. Climatol*., **22**, 1267-1284.

Boudala, F. S. and Isaac, G. A., 2006: Bulk microphysics parameterization of ice fraction for application in climate models, *Q. J. R. Meteorol. Soc.*, **132**, 2377–2392 .

Boudala, F.S., Issac, G.A., McFarlane, N.A., and Li, J., 2007: The sensitivity of the radiation budget in a climate simulation to neglecting the effect of small ice particles, *J. Climate*, **20**, 3527-3541.

Brown, P. R. A., and P. N. Francis, 1995: Improved measurements of the ice water content in cirrus using a total-water probe, *J. Atmos Oceanic Technol.*, **12**, 410–414.

Chen, T., Rossow, W.B. and Y. Zhang, 2000: Radiative effects of cloud-type variations, *J. Clim.*, **13**, 264-286.

Cober, S.G., G. A. Isaac, A.V. Korolev, and J.W. Strapp, 2001: Assessing cloud-phase conditions, *J. Appl. Meteorol.*, **40**, 1967-1983.

Cooper, W. A., 1978: Cloud physics investigation by the University of Wyoming in HIPLEX 1977, *Bureau of Reclamation Rep.*, AS, **119**, 321 pp.

Curry, J.A. 1995: Interactions among aerosols, clouds, and climate of the Arctic Ocean, *Sci. of Tot. Environ.*, **160-161**, 777-791.

Curry, J.A., E.E. Ebert, and J.L. Schramm, 1993: Impact of clouds on the surface radiation balance of the Arctic Ocean, *Meteorol. Atmos. Phys.*, **51**, 197-217.

Delanoë, J., A. Protat, J. Testud, D. Bouniol, A.J. Heymsfield, A. Bansemer, P.R.A. Brown, and R.M Forbes, 2005: Statistical properties of the normalized ice particle size distribution, *J. Geophys. Res.*, **110**, D10201.

Delanoë, J., A. Protat, D. Bouniol, A. Heymsfield, A. Bansemer, and P. Brown, 2007: The characterization of ice cloud properties from Doppler radar measurements. *J. Appl. Meteor. Climatol.*, **46**, 1682–1698. doi: 10.1029/2004JD005405

Delanoë, J., and R. J. Hogan, 2008: A variational scheme for retrieving ice cloud properties from combined radar, lidar, and infrared radiometer, *J. Geophys. Res.*, 113, D07204, doi: 10.1029/2007JD009000

Dong, M. and G.G. Mace, 2006: Cirrus microphysical properties and air motion statistics using cloud radar Doppler moments. Part I: algorithm description, *J. Appl. Metorol. Clim.*, **45**, 1690-1709.

Donovan, D. P. and A. C. A. P. van Lammeren, 2001: Cloud effective particle size and water content profile retrievals using combined lidar and radar observations: 1. Theory and examples, *J. Geophys. Res.*, **106**, 27,425–27,448, doi: 10.1029/2001JD900243

Dowling, D. R., and L. F. Radke, 1990: A summary of the physical properties of cirrus clouds, *J. Appl. Meteorol*., **29**, 970–978, doi: 10.1175/1520-0450(1990)029<0970:ASOTPP>2.0.CO;2

Ebert, E. E. and J. A. Curry, 1992: A parameterization of ice cloud optical properties for climate models, *J. Geophys. Res.*, **97**, 3831–3836, doi: 10.1029/91JD02472

Efron, B., and R.J. Tibshirani, 1993: An Introduction to the Bootstrap, 436 pp., Chapman and Hall, Boca Raton, Florida, USA

Febvre, G., J.F. Gayet, V. Scherbakov, C. Gourbeyre, and O. Jourdan, 2012: Some effects of ice crystals on the FSSP measurements in mixed-phase clouds, *Atmos. Chem. Phys.*, **12**, 8963-8977.

Field, P. R., and A. J. Heymsfield, 2003: Aggregation and scaling of ice crystal size distributions. *J. Atmos. Sci.*, **60**, 544–560.

Field, P.R., A.J. Heymsfield, and A. Bansemer, 2006: Shattering and particle interarrival times measured by optical array probes in ice clouds, *J. Atmos. Oceanic. Technol.*, **23**, 1357-1370

.

Field, P.R., A. J. Heymsfield, and A. Bansemer, 2007: Snow size distribution parameterization for midlatitude and tropical ice clouds. *J. Atmos. Sci.*, **64**, 4346–4365.

Field, P.R., and 8 co-authors 2003: Ice particle interarrival times measured with a Fast FSSP, *J. Atmos. Oceanic Technol.*, **20**, 249–261.

Fontaine, E. and 8 co-authors, 2014: Constraining mass-diameter relations from hydrometeor images and cloud radar reflectivities in tropical continental and oceanic convective anvils, *Atmos. Chem. Phys.*, **14**, 11367-11391, doi: 10.5194/acp-14-11367-2014

Francis, P. N., Hignett, P. and Macke, A. 1998: The retrieval of cirrus cloud properties from aircraft multi-spectral reflectance measurements during EUCREX'93. *Q.J.R. Meteorol. Soc.*, **124**, 1273–1291.

Fridlind, A.M., A.S. Ackerman, G. McFarquhar, G. Zhang, M.R. Poellot, P.J. DeMott, A.J. Prenni, and A.J. Heymsfield, 2007: Ice properties of single-layer stratocumulus during the Mixed-Phase Arctic Cloud Experiment: 2. Model results, *J. Geophys. Res.*, 112, D24202, doi:10.1029/2007JD008646.

Fu, Q., 1996: An accurate parameterization of the solar radiative properties of cirrus clouds for climate models, *J. Climate*, **9**, 2058–2082.

Fu, Q, and K. N. Liou, 1993: Parameterization of the radiative properties of cirrus clouds. *J. Atmos. Sci.*, **50**, 2008–2025.

Fugal, J. P., R. A. Shaw, E. W. Saw, and A. V. Sergeyev, 2004: Airborne digital holographic system for cloud particle measurements. *Applied Optics,* **43,** 5987-5995.

Fugal, J.P, and, R.A. Shaw, 2009: Cloud particle size distributions measured with an airborne digital in-line holographic instrument, *Atmos. Meas. Tech.* **2**, 259-271.

Fugal, J.P.,T. J. Schulz, and R. A. Shaw, 2009: Practical methods for automated reconstruction and characterization of particles in digital in-line holograms. *Meas. Sci. and Tech.*, **20**.

Gardiner, B.A., and J. Hallett, 1985: Degradation of in-cloud forward scattering spectrometer probe measurements in the presence of ice particles, *J. Atmos. Oceanic Technol*., **2**, 171-180.

Garrett, T.J., A. J. Heymsfield, M. J. McGill, B. A. Ridley, D. G. Baumgardner, T. P. Bui, and C. R. Webster 2004: Convective generation of cirrus near the tropopause, *J. Geophys. Res.*, **109**, D21203, doi: 10.1029/2004JD004952

Gayet, J., G. Febvre, and H. Larsen, 1996: The reliability of the PMS FSSP in the presence of small ice crystals, *J. Atmos. Oceanic Technol.*, **13**, 1300–1310.

Gettelman, A., and 9 co-authors 2010: Global simulations of ice nucleation and ice supersaturation with an improved cloud scheme in the Community Atmosphere Model, *J. Geophys. Res.*, **115**, D18216, doi:10.1029/2009JD013797

Hallett, J. 2003: Measurement in the atmosphere, in *Handbook of Weather, Climate and Water: Dynamics, Climate, Physical Meteorology, Weather Systems and Measurements*, edited by T. D. Potter, and B. R. Colman, pp. 711–720, John Wiley Hoboken, N. J.

Harrington, J.Y., and P. Q. Olsson, 2001: On the potential influence of ice nuclei on surface-forced marine stratocumulus cloud dynamics, *J. Geophys. Res.* **106**, 27,473-27,484, doi: 10.1029/2000JD000236

Harrington, J.Y., T. Reisin, W.R. Cotton, and S.M. Kreidenweis, 1999: Cloud resolving simulations of Arctic stratus: Part II: Transition-season clouds, *Atmos. Res.*, **51**, 45-75.

Hartmann, D.L., M.E. Ockert-Bell, and M.L. Michelsen, 1992: The effect of cloud type on Earth's energy balance: global analysis, *J. Climate*, **5**, 1281-1304.

Heymsfield, A.J., 2003a: Properties of tropical and midlatitude ice cloud particle ensembles. Part I: median mass diameters and terminal velocities, *J. Atmos. Sci.*, **60**, 2573-2591.

Heymsfield, A.J., 2003b: Properties of tropical and midlatitude ice cloud particle ensembles. Part II: Applications for mesoscale and climate models, *J. Atmos. Sci*., **60**, 2592-2611.

Heymsfield, A.J., 2007: On measurements of small ice particles in clouds. *Geophys. Res. Lett.*, **34,** L23812

Heymsfield, A. J., and J. L. Parrish, 1978: A Computational Technique for Increasing the Effective Sampling Volume of the PMS Two-Dimensional Particle Size Spectrometer. *J. Appl. Meteor*., **17**, 1566–1572

Heymsfield, A.J. and C.M.R. Platt, 1984: A parameterization of the particle size spectrum of ice clouds in terms of ambient temperature and ice water content, *J. Atmos. Sci.*, **41**, 846-855

Heymsfield, A.J., and L.M. Miloshevich, 1991: Limit to greenhouse warming?, *Nature*, **351**, 14-15.

Heymsfield, A. J., A. Bansemer, P.R. Field, S. L. Durden, J. L. Stith, J.E. Dye, W. Hall, and C. A. Grainger, 2002a: Observations and parameterizations of particle size distributions in deep tropical cirrus and stratiform precipitating clouds: Results from in situ observations in TRMM field campaigns, *J. Atmos. Sci.*, **59**, 3457–3491.

Heymsfield, A. J., S. Lewis, A. Bansemer, J. Iaquinta, L. M. Miloshevich, M. Kajikawa, C. Twohy, and M. R. Poellot 2002b: A general approach for deriving the properties of cirrus and stratiform ice cloud particles, *J. Atmos. Sci.*, **59**, 3–29. Heymsfield, A. J., A. Bansemer, C. G. Schmitt, C. Twohy, and M. R. Poellot, 2004a: Effective ice particle densities derived from aircraft data, *J. Atmos. Sci.*, **61**, 982–1003.

Heymsfield, A.J, and G.M. McFarquhar, 2002: Mid-latitude and tropical cirrus microphysical properties. Cirrus, *Optical Society of America*, ed. D. Lynch, 78-101

Heymsfield, A.J., and L.M. Miloshevich, 2003: Parameterizations for the cross-sectional area and extinction of cirrus and stratiform ice cloud particles, *J. Atmos. Sci.*, **60**, 936-956.

Heymsfield, A. J., and Coauthors, 2008: Testing IWC retrieval methods using Radar and ancillary measurements with in situ data. *J. Appl. Meteor. Climatol.*, **47**, 135–163.

Heymsfield, A.J., A. Bansemer, G. Heymsfield, and A.O. Fierro 2009, Microphysics of maritime tropical convective updrafts at temperatures from -20˚ to -60˚C. *J. Atmos. Sci.*, **66**, 3530-3562, doi: 10.1175/2009JAS3107.1

Heymsfield, A. J., C. Schmitt, A. Bansemer, and C. H. Twohy, 2010: Improved representation of ice particle masses based on observations in natural clouds. *J. Atmos. Sci.*, **67**, 3303–3318

Heymsfield, A.J., C. Schmitt, and A. Bansemer 2013: Ice Cloud Particle Size Distributions and Pressure-Dependent Terminal Velocities from In Situ Observations at Temperatures from 0° to −86°C. *J. Atmos. Sci.*, **70**, 4123–4154, doi: 10.1175/JAS-D-12-0124.1

Hogan, R. J., M. P. Mittermaier, and A. J. Illingworth, 2006: The retrieval of ice water content from radar reflectivity factor and temperature and its use in evaluating a mesoscale model. *J. Appl. Meteor. Climatol.*, **45**, 301–317.

Holroyd, E. W., 1987: Some techniques and uses of 2D-C habit classification for snow particles. *J. Atmos. Oceanic Technol.*, **4**, 498–511.

Ivanova, D., D. L. Mitchell, W.P. Arnott, and M. Poellot, 2001: A GCM parameterization for bimodal size spectra and ice mass removal rates in mid-latitude cirrus clouds, *Atmos. Res.*, **59-60**, 89–113.

IPCC, 2014: Climate Change 2014: Mitigation of Climate Change. *Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate*

Change [Edenhofer, O., R. Pichs-Madruga, Y. Sokona, E. Farahani, S. Kadner, K. Seyboth, A. Adler, I. Baum, S. Brunner, P. Eickemeier, B. Kriemann, J. Savolainen, S. Schlömer, C. von Stechow, T. Zwickel and J.C. Minx (eds.)]. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA

Jackson, R.C., and 9 co-authors, 2012: The dependence of ice microphysics on aerosol concentration in arctic mixed-phase stratus clouds during ISDAC and M-PACE, *J. Geophys. Res.*, **117**, D15207, doi: 10.1029/2012JD017668

Jackson, R.C. and G. M. McFarquhar, 2014: An Assessment of the Impact of Antishattering Tips and Artifact Removal Techniques on Bulk Cloud Ice Microphysical and Optical Properties Measured by the 2D Cloud Probe. *J. Atmos. Oceanic Technol.*, **31**, 2131–2144. doi: http://dx.doi.org/10.1175/JTECH-D-14-00018.1

Jackson, R.C., G.M. McFarquhar, J. Stith, M. Beals, R. Shaw, J. Jensen, J. Fugal, and A. Korolev, 2014: An assessment of the impact of anti-shattering tips and artifact removal techniques on cloud ice size distributions measured by the 2D Cloud Probe, *J. Atmos. Oceanic Technol*, **31**, 2567–2590, doi: 10.1175/JTECH-D-13-00239.1

Jakob, C. and S.A. Klein, 1999: The role of vertically varying cloud fraction in the parametrization of microphysical processes in the ECMWF model. *Quart. J. Roy. Meteor. Soc.*, **125,** 941-965.

Jensen, E.J., and 13 co-authors, 2009: On the importance of small ice crystals in tropical anvil cirrus, *Atmos. Chem. Phys.*, **9**, 5519-5537.

Joos, H., P. Spichtinger, U. Lohmann, J. F. Gayet, and A. Minikin, 2008: Orographic cirrus in the global climate model echam5, *J. Geophy. Res.*, **123,** D18205.

King, M. D., S. Platnick, P. Yang, G. T. Arnold, M. A. Gray, J. C. Riedi, S. A. Ackerman, and K. Liou, 2004: Remote sensing of liquid water and ice cloud optical thickness and effective radius in the Arctic: Application of airborne multispectral MAS data, *J. Atmos. Oceanic Technol.*, **21**, 857–875.

Knollenberg, R.G., 1970: The optical array: An alternative to scattering or extinction for airborne particle size determination. *J. Appl. Meteor.* **9,** 86-103.

Koop, T., B. Luo, A. Tsias, and T. Peter ,2000: Water activity as the determinant for homogeneous nucleation in aqueous solutions, *Nature*, **406**, 611-614, doi:10.1038/35020537

Korolev, A.V., 2007: Reconstruction of the sizes of spherical particles from their shadow images. Part I: theoretical considerations, *J. Atmos. Oceanic Technol.*, **24**, 376-389.

Korolev, A.V., J.W. Strapp, and G.A. Isaac, 1998: Evaluation of the accuracy of PMS optical array probes. *J. Atmos. Oceanic. Technol.*, **15**, 708-720

Korolev, A., and Isaac, G.A. 2005: Shattering during sampling by OAPs and HVPS. Part I: Snow particles, *J. Atmos. Tech.*, **22**, 528-542.

Korolev, A.V., and E. F. Emery, J. W. Strapp, S. G. Cober, G. A. Isaac, M. Wasey, D. Marcotte, 2011: Small ice particles in tropospheric clouds: fact or artifact?, *Bull. Amer. Meteorol. Soc.*, **92**, 967–973.

Korolev, A.V., and 6 co-authors 2011: Small ice particles in tropospheric clouds: fact or artifact?, *Bull. Amer. Meteorol. Soc.*, **92**, 967–973.

Korolev, A., E. Emery, and K. Creelman, 2013a: Modification and Tests of Particle Probe Tips to Mitigate Effects of Ice Shattering, *J. Atmos. Oceanic Technol.*, **30**, 690–708.

Korolev, A., E. Emery, J.W. Strapp, S.G. Cover, and G.A. Isaac, 2013b: Quantification of the effects of shattering on airborne ice particle measurements, *J. Atmos. Oceanic Technol.*, **30**, 2527–2553

Korolev, A. and P. Field, 2014: Assessment of performance of the inter-arrival time algorithm to identify ice shattering artifacts in cloud particle probes measurements, *Atmos. Meas. Tech. Dicuss.*, **7**, 10249-10292, doi:10.5194/amtd-7-10249-2014

Kr*ä*mer, M., and 11 co-authors, 2009: Ice supersaturations and cirrus cloud crystal numbers, *Atmos. Chem. Phys*., **9**, 3505-3522, doi:10.5194/acp-9-3505-2009

Kristjánsson, J. E., J. M. Edwards, and D. L. Mitchell, 2000: Impact of a new scheme for optical properties of ice crystals on climates of two GCMs, *J. Geophys. Res.*, **105**, 10,063–10,079, doi: 10.1029/2000JD900015

Kulie, M. S., and R. Bennartz, 2009: Utilizing spaceborne Radars to retrieve dry snowfall. *J. Appl. Meteor. Climatol.*, **48**, 2564–2580.

Lawson, R.P., 2011: Effects of ice particles shattering on the 2D-S probe, *Atmos. Meas. Tech.*, **4**, 1361-1381.

Lawson, R.P., B.A. Baker, C.G. Schmitt, and T.L. Jensen ,2001: An overview of microphysical properties of Arctic clouds observed in May and July 1998 during FIRE-ACE, *J. Geophys. Res.*, **106**, 14989-15104, doi: 10.1029/2000JD900789

Lawson, R.P., D. O'Connor, P. Zmarzly, K. Weaver, B.A. Baker, Q. Mo, and H. Jonsson, 2006: The 2D-S (Stereo) probe: design and preliminary tests of a new airborne, high speed, high-resolution particle imaging probe, *J. Atmos. Oceanic Technol.* **23**, 1462-1477, doi: 10.1175/JTECH1927.1

Lawson, R. P., E. Jensen, D. L. Mitchell, B. Baker, Q. Mo, and B. Pilson, 2010: Microphysical and radiative properties of tropical clouds investigated in TC4 and NAMMA, *J. Geophys. Res.*, **115**, D00J08, doi: 10.1029/2009JD013017

Lindzen, R. , M.D. Chou, and A.Y. Hou, 2001: Does the Earth have an adaptaive infrared iris? *Bull. Amer. Met. Soc.* **82**, 417–432. doi:10.1175/1520-0477(2001)082<0417:DTEHAA>2.3.CO;2

Liou, K. N., Y. Gu, Q. Yue, and G. McFarquhar, 2008: On the correlation between ice water content and ice crystal size and its application to radiative transfer and general circulation models, *Geophys. Res. Lett.*, **35**, L13805.

Locatelli, J.D. and P.V. Hobbs, 1974: Fall speeds and masses of solid precipitation particles, *J. Geophys. Res.*, **79**(15), 2185-2197.

Lubeke, A.E., L.M. Avallone, C. Schiller, J. Meyer, C. Rolf, and M. Kramer 2013: Ice water content of Arctic, midlatitude, and tropical cirrus – Part 2: Extension of the database and new statistical analysis, *Atmos. Chem. Phys.*, **13**, 6447-6459, doi:10.5194/acp-13-6447-2013, doi:10.5194/acp-13-6447-2013

Mace, G. G., A. J. Heymsfield, and M. R. Poellot, 2002: On retrieving the microphysical properties of cirrus clouds using the moments of the millimeter-wavelength Doppler spectrum, *J. Geophys. Res.*, **107**, 4815, doi: 10.1029/2001JD001308

Macke, A., P. N. Francis, G. M. McFarquar, and S. Kinne, 1998: The role of ice particle shapes and size distributions in the single scattering properties of cirrus clouds. *J. Atmos. Sci.*, **55**, 2874–2883.

Matrosov, S. Y., A. V. Korolev, and A, J. Heymsfield, 2002: Profiling cloud ice mass and particle characteristic size from doppler radar measurements, *J. Atmos. Oceanic Technol.*, **19**, 1003–1018

May, P.T., J.H. Mather, G. Vaughn, K.N. Bower, C. Jakob, G.M. McFarquhar, and G.G. Mace, 2008: The Tropical Warm Pool International Cloud Experiment, *Bull. Amer. Meteor. Soc.*, **89**, 629-645

McFarquhar, G. M., 2001: Comments on ‘‘Parameterization of effective sizes of cirrus-cloud particles and its verification against observation’’ by Zhian Sun and Lawrie Rikus. (October B, 1999, 125, 3037–3055), *Quart. J. Roy. Meteor. Soc.*, **127**, 261–265.

McFarquhar, G.M., and A.J. Heymsfield, 1996: Microphysical characteristics of three anvils sampled during the Central Equatorial Pacific Experiment, *J. Atmos. Sci.*, **53**(17), 2401-2423, doi: 10.1175/1520-0469(1996)053<2401:MCOTAS>2.0.CO;2

McFarquhar, G.M., and A.J. Heymsfield, 1997: Parameterization of tropical cirrus ice crystal size distributions and implications for radiative transfer: results from CEPEX, *J. Atmos. Sci.*, **54**, 2187-2200.

McFarquhar, G.M., and A.J. Heymsfield, 1998: The definition and significant of an effective radius for ice clouds, *J. Atmos. Sci.*, **55**, 2039-2052

McFarquhar, G.M., A.J. Heymsfield, A. Macke, J. Iaquinta, and S.M. Aulenbach, 1999: Use of observed ice crystal sizes and shapes to calculate mean scattering properties and multi-

spectral radiances: CEPEX 4 April 1993 case study. *J. Geophys. Res.*, **104**, 31763-31779, doi: 10.1029/1999JD900802

McFarquhar, G.M., P. Yang, A. Macke, and A.J. Baran, 2002: A new parameterization of single scattering solar radiative properties for tropical anvils using observed ice crystal size and shape distributions, *J. Atmos Sci*., **59**, 2458-2478.

McFarquhar, G.M., S. Iacobellis, and R.C.J. Somerville, 2003: SCM Simulations of tropical ice clouds using observationally based parameterizations of microphysics, *J. Climate*., **16**, 1643-1664.

McFarquhar, G.M., and R.A. Black, 2004: Hurricane observations of particle size and phase and their implications for mesoscale modeling of microphysical processes. *J. Atmos Sci*., **61**, 422-439

McFarquhar, G.M., and S.G. Cober, 2004: Single-scattering properties of mixed-phase Arctic clouds at solar wavelengths: Impacts on radiative transfer. *J. Climate*, **17**, 3799-3813

McFarquhar, G.M., and 7 co-authors, 2007a: Ice properties of single-layer stratocumulus during the Mixed-Phase Arctic Cloud Experiment: 1. Observations, *J. Geophys. Res.*, **112**, D24201.

McFarquhar, G. M., J. Um, M. Freer, D. Baumgardner, G. L. Kok, and G. G. Mace, 2007b: Importance of small ice crystals to cirrus properties: Observations from the Tropical Warm Pool International Cloud Experiment (TWP-ICE), *Geophys. Res. Lett.*, **34**, L13803, doi: 10.1029/2007GL029865

McFarquhar, G. M., M. S. Timlin, R. M. Rauber, B. F. Jewett, J. A. Grim, and D. P. Jorgensen, 2007c: Vertical Variability of Cloud Hydrometeors in the Stratiform Region of Mesoscale Convective Systems and Bow Echoes. *Mon. Wea. Rev.*, **135**, 3405–3428, doi: 10.1175/MWR3444.1

McFarquhar, G.M., and G. Zhang, M.R. Poellot, G.L. Kok, R. McCoy, T. Tooman, and A.J. Heymsfield, 2007a: Ice properties of single-layer stratocumulus during the Mixed-Phase Arctic Cloud Experiment: 1. Observations, *J. Geophys. Res.*, **112**, D24201.

McFarquhar, G.M., B. Schmid, A. Korolev, J.A. Ogren, P.B. Russell, J. Tomlinson, D.D. Turner, and W. Wiscombe, 2011: Airborne instrumentation needs for climate and atmospheric research. *Bull. Amer. Meteor. Soc.*, **92**, 1193-1196.

McFarquhar, G.M., T. Hsieh, M. Freer, J. Mascio, and B.F. Jewett 2014: The characterization of ice hydrometeor gamma size distributions as volume in N_0-μ-λ phase space: implications for microphysical process modeling, *J. Atmos. Sci.*, in press

Milbrandt, J.A. and M.K. Yau, 2005: A multimoment bulk microphysics parameterization, Part I: Analysis of the role of the spectral shape parameter, *J. Atmos. Sci.*, 62, 3051-3064, doi: 10.1175/JAS3534.1

Mitchell, D. L. 1996: Use of mass- and area-dimensional power laws for determining precipitation particle terminal velocities, *J. Atmos. Sci*., **53**, 1710-1723.

Mitchell, D.L, S.K. Chai, Y. Liu, A.J. Heymsfield, and Y. Dong 1996, Modeling cirrus clouds, Part I: treatment of bimodal size spectra and case study analysis, *J. Atmos. Sci.*, *53*, 2952-2966, doi: 10.1175/1520-0469(1996)053<2952:MCCPIT>2.0.CO;2

Mitchell, D. L., P. Rasch, D. Ivanova, G. McFarquhar, and T. Nousiainen, 2008: Impact of small ice crystal assumptions on ice sedimentation rates in cirrus clouds and GCM simulations, *Geophys. Res. Lett.*, **35**, L09806, doi: 10.1029/2008GL033552

Mitchell, D.L., R.P. D'Entremont, and R.P. Lawson, 2010: Inferring cirrus size distributions through satellite remote sensing and microphysical databases, *J. Atmos. Sci.*, **67**, 1106-1125.

Mitchell, D. L., S. Mishra, and R. P. Lawson, 2011: Representing the ice fall speed in climate models: Results from Tropical Composition, Cloud and Climate Coupling (TC4) and the Indirect and Semi-Direct Aerosol Campaign (ISDAC), *J. Geophys. Res.*, **116,** D00T03, doi: 10.1029/2010JD015433

Mishra, S. D.L. Mitchell, D.D. Turner, and R. P. Lawson, 2014: Parameterization of ice fall speeds in midlatitude cirrus: Results from SPartICus, *J. Geophys. Res.*, **119**, 3857-3876, doi:10.1002/2013JD020602

Morrison, H., G. Thompson, and V. Tatarskii, 2009: Impact of cloud microphysics on the development of trailing stratiform precipitation in a simulated squall line: Comparison of one and two-moment schemes, *Mon. Wea. Rev.*, **138**, 4254-4267, doi: 10.1175/2008MWR2556.1

Morrison, H., and 11 co-authors, 2011: Intercomparison of cloud model simulations of Arctic mixed-phase boundary layer clouds observed during SHEBA/FIRE-ACE. *J. Adv. Model. Earth Syst.*, **3**, M06003.

Morrison, H., G. de Boer, G. Feingold, J. Harrington, M.D. Shupe, and K. Sulia, 2012: Resilience of persistent Arctic mixed-phase clouds, *Nature Geoscience*, **5,** 11-17.

Moshtagh, N., 2006: Minimum volume enclosing ellipse. Technical report available from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.116.7691&rep=rep1&type=pdf

Muhlbauer, A., T. P. Ackerman, J. M. Comstock, G. S. Diskin, S. M. Evans, R. P. Lawson, and R. T. Marchand, 2014: Impact of large-scale dynamics on the microphysical properties of midlatitude cirrus, *J. Geophys. Res. Atmos.*, **119**, 3976–3996, doi:10.1002/2013JD020035.

Nasiri, S. L., B. A. Baum, A. J. Heymsfield, P. Yang, M. R. Poellot, D. P. Kratz, and Y. Hu, 2002: The Development of midlatitude cirrus models for MODIS Using FIRE-I, FIRE-II, and ARM in situ data. *J. Appl. Meteor.*, **41**, 197–217.

Nousainen, T., and G.M. McFarquhar, 2004: Light scattering by quasi-spherical ice crystals, *J. Atmos. Sci.*, **61**, 2229-2248, doi: 10.1175/1520-0469(2004)061<2229:LSBQIC>2.0.CO;2

Pokharel, B., and G. Vali, 2011: Evaluation of collocated measurements of radar reflectivity and particle sizes in ice clouds, *J. Appl. Meteorol. Clim.*, **50**, 2104-2119.

Protat, A., J. Delanoë, D. Bouniol, A. J. Heymsfield, A. Bansemer, and P. Brown, 2007: Evaluation of ice water content retrievals from cloud radar reflectivity and temperature using a large airborne in situ microphysical database. *J. Appl. Meteor. Climatol.*, **46**, 557–572.

Pruppacher, H.R. and Klett, J.D., 1997: Microphysics of clouds and precipitation, Atmospheric and Oceanographic Sciences Library, Kluwer Academic Publishers, Dordrecht, The Netherlands

Ramanathan, V., and W. Collins, 1991: Thermodynamic regulation of ocean warming by cirrus clouds deduced from observations of the 1987 El-Nino. *Nature*, **351**, 27-32.

Ramaswamy, V. and V. Ramanathan, 1987: Solar absorption by cirrus clouds and the maintenance of the tropical upper troposphere thermal structure, *J. Atmos. Sci.*, **46**, 2293-2310.

Rigo, T. and M.C. Llasat, 2004: A methodology for the classification of convective structures using meteorological radar: Application to heavy rainfall events on the Mediterranean coast of the Iberian Peninsula, *Nat. Hazard. Earth. Sys.*, **4**, 59-68, doi:10.5194/nhess-4-59-2004

Ryan, B.F., 1996: On the global variation of precipitating layer clouds, *Bull. Amer. Meteor. Soc.*, **77**, 53-70, doi:10.1175/1520-0477(1996)077<0053:OTGVOP>2.0.CO;2

Rolland, P., K. N. Liou, M. D. King, S. C. Tsay, and G. M. McFarquhar, 2000: Remote sensing of optical and microphysical properties of cirrus clouds using Moderate-Resolution Imaging Spectroradiometer channels: Methodology and sensitivity to physical assumptions, *J. Geophys. Res.*, **105**, 11,721–11,738, doi: 10.1029/2000JD900028

Sanderson, B.M., C. Piani, W.J. Ingram, D.A. Stone, and M.R. Allen, 2008: Towards constraining climate sensitivity by linear analysis of feedback patterns in thousands of perturbed-physics GCM simulations, *Clim. Dynamics*, **30**, 175-190

Sassen, K., 1997: Cirrus clouds: A modern perspective, in *Cirrus*, edited by D. Lynch, pp. 11–40, Oxford University Press.

Saw, E., R. A. Shaw, J. P. L. C. Salazar, and L. R. Collins, 2012: Spatial clustering of polydisperse inertial particles in turbulence: II. Comparing simulation with experiment. *New Journal of Physics*, **14**, 105031.

Schmitt, C. G. and A. J. Heymsfield, 2009: The size distribution and mass-weighted terminal velocity of low-latitude tropopause cirrus crystal populations. *J. Atmos. Sci.*, **66**, 2013–2028.

Schumann, U., B. Mayer, K. Gierens, S. Unterstrasser, P. Jessberger, A. Petzold, C. Voigt, and J-F. Gayet, 2011: Effective radius of ice particles in cirrus and contrails. *J. Atmos. Sci.*, **68**, 300–321.

Siebert, S., S. Gerashchenko, A. Gylfason, K. Lehmann, L.R. Collins, R.A. Shaw, and Z. Warhaft 2010: Towards understanding the role of turbulence on droplets in clouds: In situ and laboratory measurements, *Atmos. Res.*, **97**, 426-437

Solomon, A., Shupe, M. D., Persson, P. O. G., and Morrison, H. 2011: Moisture and dynamical interactions maintaining decoupled Arctic mixed-phase stratocumulus in the presence of a humidity inversion, *Atmos. Chem. Phys. Discuss.*, **11**, 13469-13524.

Spichtinger, P., K. Gierens, and H. Wernli, 2005a: A case study on the formation and evolution of ice supersaturation in the vicinity of a warm conveyor belt's outflow region, *Atmos. Chem. Phys.*, **5**, 973–987, doi:10.5194/acp-5-973-2005

Spichtinger, P., K. Gierens, and A. Dornbrack, 2005b: Formation of ice supersaturation by mesoscale gravity waves, *Atmospheric Chem. Phys.*, **5**, 1243–1255, SRef-ID: 1680-7324/acp/2005-5-1243

Spuler, S. and J. Fugal 2011: Design of a digital, in-line, holographic imaging system for airborne measurement of clouds. *App. Optics*, **50**, 1405-1412.

Stackhouse, P.W.J. and Stephens, G.L., 1991: A theoretical and observational study of the radiative properties of cirrus: Results from FIRE 1986, *J. Atmos. Sci*, **48**, 2044-2059.

Stephens, G.L., 2005: Cloud feedbacks in the climate system, a critical review, *J. Climate*, **18**, 237-273.

Stephens, G.L., S. Tsay, P.W. Stackhouse Jr., and P.J. Flatau, 1990: The relevance of the microphysical and radiative properties of cirrus clouds to climate and climate feedback, *J. Atmos. Sci.*, **47**, 1742-1753.

Szrymer, W., A. Tatarevic, and P. Kollias, 2012: Ice clouds microphysical retrieval using 94-GHz Doppler radar observations: Basic relations within the retrieval framework, **117,** *D14203*, doi:10.1029/2011DJ106675

Takano, Y., and K. N. Liou, 1989: Solar radiation transfer in cirrus clouds. Part I: Single-scattering and optical properties of hexagonal ice crystals, *J. Atmos. Sci.*, **46**, 3–19.

Ulbrich, C.W., 1983: Natural variations in the analytical form of the raindrop size distribution, *J. Appl. Meteorol. Clim.*

Um, J. and G.M. McFarquhar, 2007: Single-Scattering Properties of Aggregates of Bullet Rosettes in Cirrus, *J. Appl. Meteor. Climatol.*, **46**, 757–775

Um, J., and G.M. McFarquhar, 2009: Single scattering properties of aggregates of plates, *Quart. J. Roy. Meteol. Soc.*, **135**, 291-304, doi:10.1002/qj.378

Um, J., and G.M. McFarquhar, 2011: Dependence of the single-scattering properties of small ice crystals on idealized shape models, *Atmos. Chem. Phys.*, **11**, 3159-3171, doi:20.5194/acp-11-3159-20111

van Zadelhoff, G.-J., A. J. Heymsfield, D. P. Donovan, and M. J. McGill, 2007: Evaluating lidar-radar microphysics retrieval using in situ measurements, *J. Geophys. Res.*, **112,** doi: 10.1029/2006JD007202

Vidaurre, G., and J. Hallett, 2009: Particle impact and breakup in aircraft measurement, *J. Atmos. Oceanic Technol.*, **26**, 972-983.

Vogelmann, A. M., and T. P. Ackerman, 1995: Relating Cirrus Cloud Properties to Observed Fluxes: A Critical Assessment. *J. Atmos. Sci.*, **52**, 4285–4301.

Wang, Z., and K. Sassen, 2002a: Cirrus cloud microphysical property retrieval using Lidar and Radar measurements. Part I: algorithm description and comparison with In Situ Data. *J. Appl. Meteor.*, **41**, 218–229.

Warren, S. G., Hahn, C. J., London, J., Chervin, R. M., and Jenne, R. L. 1986: Global distribution of total cloud cover and cloud type amounts over land, NCAR Tech. Note 273+STR, NCAR Boulder, CO, 29 pp. + 200 maps\ Wylie. D, D.L. Jackson. W.P. Menzel, and J.J. Bates, 2005: Trends in global cloud cover in two decades of HIRS Observations, *J. Clim.*, **18**, 3021-3031.

Wylie. D, D.L. Jackson. W.P. Menzel, and J.J. Bates, 2005: Trends in global cloud cover in two decades of HIRS Observations, *J. Clim.*, **18**, 3021-3031.

Yang, F., M. Ovchinnikov, and R. A. Shaw, 2013a: Minimalist model of ice microphysics in mixed-phase stratiform clouds, *Geophys. Res. Lett.*, **40**, 3756–3760, doi:10.1002/grl.50700

Yang, P., B.A. Baum, A.J Heymsfield, Y.X. Hu, S. Tsay, and S. Ackerman, 2003: Single scattering properties of droxtals, *J. Quant. Spectroscopy and Radiative Transfer*, **79-80**, 1159-1169.

Yang, P., H. Wei, H.L. Huang, B.A. Baum, Y.X. Hu, G.W. Kattawar, M.I. Mishchenko, and Q. Fu, 2005: Scattering and absorption property database for nonspherical ice particles in the near- through far-infrared spectral region. *Appl. Opt.*, **44**, 5512-5523.

Yang, P., L. Bi, B. A. Baum, K.N. Liou, G. W. Kattawar, M. I. Mishchenko, B. Cole, 2013b: Spectrally Consistent Scattering, Absorption, and Polarization Properties of Atmospheric Ice Crystals at Wavelengths from 0.2 to 100 μm. *J. Atmos. Sci.*, **70**, 330–347.

Yoshida, R., H. Okamoto, Y. Hagihara, and H. Ishimoto, 2010: Global analysis of cloud phse and ice crystal orientation from Cloud-Aerosol Lidar and Infrared Pathfinder Satellite from Cloud-Aerosol Lidar and Infrared Pathfinder Satellite Observation (CALIPSO) data using attenuated backscattering and depolarization ratio. *J. Geophys. Res.*, **115**, D00H32, doi:10.1029/2009JD012334

Zender, C.S. and J.T. Kiehl, 1994: Radiative sensitivities of tropical anvils to small ice crystals, *J. Geophys. Res.*, **99**, 25,869-25,880, doi: 10.1029/94JD02090

Zhang, Y., A. Macke, and F. Albers, 1999: Effect of crystal size spectrum and crystal shape on stratiform cirrus radiative forcing, *Atmos. Research*, **52**, 59-75.

Zhao, Y., G. G. Mace, and J. M. Comstock, 2010: The occurrence of particle size distribution bimodality in midlatitude cirrus as inferred from ground-based remote sensing data. J. *Atmos. Sci.*, **68**, 1162–1177.

APPENDIX A

CODES FOR ANALYSIS OF PROBE DATA

A.1 Introduction

This appendix describes the codes that were used in this dissertation to derive the particle size distributions from the optical array probe data. Section 1 provides a brief introduction to the University of Illinois Optical Array Probe Processing Software (IOPPS). Section 2 outlines the basic methodology by which the probe data are processed, whereas Section 3 describes the implementation of the methodology in the processing code. IOPPS is run within the MATrix LABoratory (MATLAB) software, is consists of the files listed in Table A.1.

Table A.1. List of MATLAB codes

File Name	Description
calculate_reject_unified.m	**Calculates reason particles are rejected (or contains separate flag if not rejected)**
calc_sa_randombins.m	**Calculates probe sample area of particle as a function of D_{max}**
calc_vt.m	**Calculates the particle fallspeed of a single particle.**
CGAL_RecSize.mexa64	**Calculates minimum rectangle around particle using CGAL library (https://www.cgal.org/)**
CGAL_EllipseSize.mexa64	**Calculates minimum ellipse around particle using CGAL library**
dropsize.m	**Generates D_{max} for particle using different definitions**
dropsize_new.m	**Generates D_{max} according to longest chord through particle**
Holroyd.m	**Runs automated habit identification scheme of Holroyd (1987) on particle**
lwc_calc.m	**Calculates liquid water content from a size distribution**
Imageview.m	**Views image file**
imgProc_dm.m	**Generates particle-by-particle parameters from an image netCDF file**
ImgView.m	**Code for the ImgView program**
ImgView.fig	**ImgView program**
read_binary_2ds.m	**Converts a raw .2DS file to netCDF format**
read_binary_hvps.m	**Converts a raw .HVPS file to netCDF format**

Table A.1. (cont.)

read_binary_image_plows.m	**Converts a raw NCAR 2DC file to netCDF format**
read_binary_image_ec.m	**Converts an Environment Canada 2DC/2DP file to netCDF format**
read_binary_image.m	**Convert CIP binary image to netCDF format**
single_area.m	**Calculates the area of a paricle using a-D relationship from Mitchell (1996), used in calculation of particle fallspeed**
sizeDist.m	**Generate size distributions.**

There are three steps that are followed to generate size distributions from raw 2D data file using IOPPS. The first step processes the raw 2D data file. Section A.3.2 describes the processing of an OAP raw data file using this code. The second step involves generating shape parameters for each particle, such as the particle maximum dimension and cross sectional area that are used to determine the size and shape of a particle as well as to determine if a particle is an artifact rather than a real particle. Section A.2.1 describes the particle shape parameters and particle acceptance criteria in detail, while Section A.3.3 and A.3.4 describe how to generate these particle shape parameters from raw 2D data. The third and final step involves using the particle shape parameters to determine the size distributions. Section A.3.5 describes how to derive the particle size distributions from the particle shape parameters. Thereafter, Section A.4 describes how to view the size distributions and Section A.5 describes how to view the particle images.

A.2. Processing methods

This section describes the methodology used to determine the particle size distributions from the optical array probe images. The raw data needs to be converted to netCDF format in the first step. In the second step, numerous parameters are generated for each particle to describe

their shapes. These include parameters such as the projected area A_p the length of the particle in the direction perpendicular to the photodiode array *L*, and the length of the particle in the direction parallel to the diode array *W*. In addition to these parameters, several different versions of D_{max} are generated by the software, which include *image_diam_circle_fit*, *image_diam_horiz_chord_corr*, *image_diam_following_bamex_code*, and *image_diam_vert_chord* and the parameters related to the minimum rectangle and ellipse surrounding the particle in Table A.2. As mentioned in Chapter 2, some of these parameters can be used to eliminate shattered artifacts.

Table A.2. List of parameters generated for each particle.

netCDF Variable Name	**Description**
'area_hole_ratio'	A_p /area of hole inside particle
'bin_stats'	Number of times photodiode x is shadowed for the file
'Date'	Day of month
'edge_at_max_hole'	# of photodiodes between edges of the particle at the slice that contains the largest gap inside the particle.
'holroyd_habit'	Habit code according to Holroyd (1987) algorithm: 'M' = not calculated, zero image 'C' = not calculated, center is out 't' = tiny 'o' = oriented 'l' = linear 'a' = aggregate 'g' = graupel 's' = sphere 'h' = hexagonal 'i' = irregular 'd' = dendrite

Table A.2. (cont.)

'image_area' (A_p)	# of shaded photodiodes[1]
'image_auto_reject'	ASCII reject code generated by calculate_reject_unified.m: '0' = Accepted 'a' = Aspect ratio > 6 't' = Aspect ratio > 5 + image touching edge 'p' = % of photodiodes shadowed in rectangle encompassing particle < 25% 'h' = Hollow particle 's' = Split image 'z' = Zero area image 'f' = Zero area image
'image_axis_ratio'	Max(L, W)/min(L, W)
'image_center_in'	1 = image's center inside photodiode array, 0 = image's center outside photodiode array
'Image_diam_AreaR'	Area equivalent diameter in mm
'image_diam_circle_fit'	D_{max} calculated from Heymsfield and Parish (1978) in mm
'image_diam_following_bamex_code'	D_{max} from maximum length chord through particle in mm
'image_diam_horiz_chord_corr'	As above, but Korolev (2007) correction applied for hollow spherical particles before calculating D_{max} in mm
'image_diam_horiz_chord'	D_{max} from # of slices + 1 in mm
'image_diam_minR'	D_{max} of smallest circle enclosing particle in mm
'image_diam_vert_chord'	D_{max} from maximum length in photodiode direction in mm
'Image_EllipseAngle'	Angle of ellipse with respect to time direction I in radians
'Image_EllipseL'	Length of smallest ellipse enclosing image in mm
'Image_EllipseW'	Width of smallest ellipse enclosing image in mm
'image_hollow'	1 = image is hollow, 0 = not hollow
'image_length' (L)	Length of particle in time direction in # of photodiodes.

[1] The area of the particle is then determined by multiplying the number of shaded photodiodes by the length of the photodiode in the direction perpendicular to the array by the length of the photodiode parallel to the photodiode array.

Table A.2. (cont.)

'image_longest_y'	Longest chord through particle in time direction
'image_max_bottom_edge_touching'	Maximum # of times the bottom photodiode is shadowed in succession
'image_max_top_edge_touching'	Maximum # of times the top photodiode is shadowed in succession
'Image_perimeter'	Perimeter of image in mm
'Image_RectangleAngle'	Angle of rectangle with respect to time direction in radians
'Image_RectangleL'	Length of smallest rectangle enclosing image in mm
'Image_RectangleW'	Width of smallest rectangle enclosing image in mm
'image_touching_edge'	1 = Image touching edge 0 = Image entirely in array
'image_width' (*W*)	Length of particle in photodiode direction in # of shaded photodiodes[1]
'inter_arrival'	Interarrival time in seconds
'max_hole_diameter'	Diameter of largest hole inside particle in # of photodiodes[1]
'msec'	Milliseconds portion of timestamp for record
'parent_rec_num'	The index of the record in which the particle resides
'part_z'	Depth of particle in object plane calculated via lookup table
'particle_microsec'	Microseconds portion of particle time
'particle_millisec'	Milliseconds portion of particle time
'particle_num'	Particle # (starting with 1 for 1st particle and so on)
'particle_time'	Particle UTC time [HHMMSS]
'percent_shadow_area'	$A/(LW)$
'position'	Start slice # of particle in frame (time direction)
'size_factor'	Divide maximum dimension by this quantity to enable Korolev (2007) correction
'Time'	Record UTC time [HHMMSS]

Since there are other artifacts present in optical array probe data in addition to shattered artifacts including streakers and stuck bits, these artifacts are removed in this second step.

Particles are rejected if (i) $L/W > 5$ (or 6 if particle touches edge of photodiode array) to remove streakers and stuck bits with large aspect ratios. If a particle is a spherical hollow particle, it is assumed to be an out of focus particle and a correction factor is used to adjust the particle size according to Korolev (2007).

After these image parameters are generated, size distributions from the particle

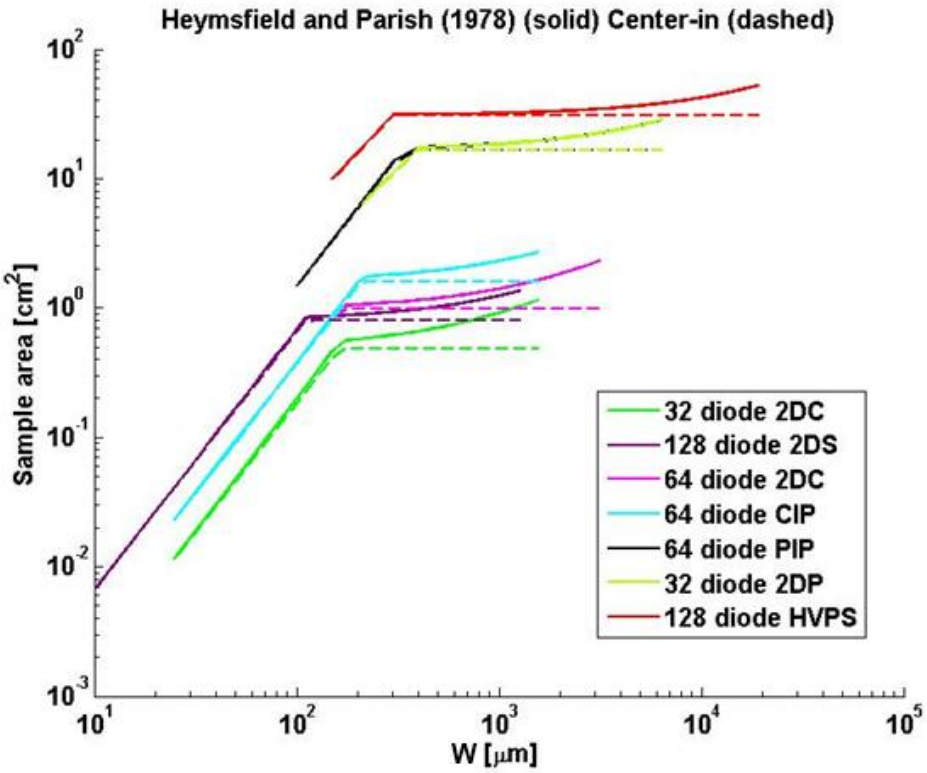


Figure A.1. Sample area as a function of W for various probes indicated in the legend. Solid lines indicate the sample area when the Heymsfield and Parish (1978) image reconstruction is used on every particle, and dashed lines indicate time periods when only particles with their centers inside the photodiode array are considered.

information are generated in the third step. Since several different versions of D_{max} can be generated, a final check for streakers and stuck bits is done in this third step since the area ratio $A_r = A_p/((\pi/4)D_{max}{}^2)$ is sentistive to the choice of D_{max}. If $A_r < 0.2$, then the particle is removed. In addition, if the sum of every Δt in a record is not within 40% of the record elapsed time, then

the record is ignored. Further, if any Δt is less than some specified threshold, the particle and the preceeding particle are identified as shattered artifacts and removed. The threshold is selected following the methodology outlined in Chapter 2, Section 3. After all artifacts have been rejected, the number of accepted particles in size bin *j* in a 1 s time interval is determined and can be represented as $N_{acc}(j)$. The 1 s number distribution function $N(D_j)$, where D_j is the midpoint of *D* in bin *j* with width ΔD_j is then calculated as

$$N(D_j) = \frac{N_{acc}(j)}{(SA)(TAS)(1-t_d)} \quad (1)$$

where *SA* is the probe sample area, *TAS* the true airspeed, and t_d is the deadtime of the probe in the 1 s time period. $1 - t_d$ is used instead of 1 s to account for time periods when the probe is not recording particles due to buffer overload. The *SA* is determined by *SA = (DOF)(EAW)*, where *DOF* is the depth of field and the effective area width *EAW = nd*, where *n* is the number of photodiodes in the array and *d* is the width of an individual diode given the magnificiation of the optical probe. If particles imgaged on one of the end photodiodes are reconstructed using the Heymsfield and Parish (1978) algorithm, then *EAW = nd + 0.72W*. The *DOF* is determined as the lesser of the formula introduced by Baumgardner and Korolev (1997) or the distance between the probe arms. The *SA* as a function of *W* for the probes currently incorporated in the University of Illinois software is given in Figure A.1. The size distribution is sorted into a number of different A_r bins as well as into a number of different bins according to D_{max}. Thus, $N(D_j, A_{rk})$, where *j* represents the j^{th} size bin with midpoint D_j and in the k^{th} A_r bin with area ratio midpoint A_{rk}. In order to determine $N(D_j, A_{rk})$, one simply replaces the numerator of Equation (1) with $N_{acc}(j,k)$, where $N_{acc}(j,k)$ represents the number of particles in size bin *j* and area ratio bin *k*.

The bulk extinction β can then be determined by

$$\beta = \sum_j \sum_k Q_{ext}(D_{j,} A_{r_k}) \frac{\pi}{4} {D_j}^2 A_{r_k} N(D_j, A_{r_k}) \Delta D_j \quad (2)$$

where $Q_{ext}(D_{j,} A_{r_k})$ is the extinction efficiency. For particles with D > 125 μm at visible wavelengths, $Q_{ext}(D_{j,} A_{r_k}) \approx 2$ since geometric optics applies (Um and McFarquhar 2007).

To estimate *IWC*, information about the relationship between crystal mass m and D_{max}, which depends on ice crystal habit, is required. There are two different methods for calculating the *IWC*. One method derives *IWC* using

$$IWC = \sum_j \sum_k f_k(D_j) \alpha_k D_j^{\beta_k} N(D_j) \Delta D_j \quad (3)$$

where $f_k(D_j)$ is the fraction of crystals in the bin centered at D_j having crystal habit k, α_k and β_k are habit-dependent coefficients listed in Table 3.2 that define the mass of an individual crystal $m = \alpha_k D_j^{\beta k}$,and $N(D_j)$ is the number distribution function for bin j with midpoint D_j and width ΔD_j. $N(D_j)$ is obtained by summing the contributions from all area ratio bins in that size bin with A_r > 0.2. The $f_k(D_j)$ can either be determined from a manual or automated habit classification scheme using data from the probe used to obtain $N(D_j)$ or from an alternate probe, such as the CPI, that provides higher resolution images making habit identification simpler and more accurate The CPI is not used to give $N(D_j)$ because it has a small and poorly defined sample volume so that statistically significant concentrations of particles cannot be determined. A second method for computing IWC_{BL06} uses the cross sectional mass-area relation of Baker and Lawson (2006), IWC_{BL06},to determine the mass of each particle from the area measured by the optical array probes. For particles touching an edge of a photodiode array, area calculated from Heymsfield and Parish (1978) image reconstruction is used. If the mass calculated by this method was greater than that of an ice sphere with the same maximum dimension, the ice sphere mass is used instead. IWC_{BL06} is given by

$$IWC = \sum_j \sum_k a\left(\frac{\pi}{4} {D_j}^2 A_{r_k}\right)^{1.218} N(D_j, A_{r_k})\, \Delta D_j \quad (4)$$

where a = 0.115 mg mm$^{-2.436}$. Mass-weighted terminal velocity is calculated using the method proposed by Heymsfield and Westbrook (2010) which parameterizes the terminal velocity as a function of particle mass and area ratio.

A.3. Tutorial

A.3.1 Downloading the code

This section outlines the instructions for downloading IOPPS and determining size distributions from a raw optical array probe data file. IOPPS is currently available in the MATLAB programming language. This code inputs raw optical array probe data recorded by a probe's controlling software and combines them with the aircraft state parameter data. The code outputs information on the morphology of individual particles as well as particle size distributions, and estimates of bulk extinction, mass weighted terminal velocity, and ice water content. IOPPS was originally based on a code processing 2DC and HVPS data written at the National Center for Atmospheric Research by Dr. Bill Hall in 1998. Originally written in IDL, it became a standalone code at the University of Illinois in 2001 and has subsequently undergone several revisions. These revisions include implementing several different technqiues to derive D_{max}, the removal of shattered artifacts, and the correction for diffraction effects in spherical hollow particles. The code has also been modified to process CIP, 2DP, and 2DS datasets.

IOPPS is version controlled using a Subversion client. This ensures that changes made by a user are automatically updated for all users, and allows reversion back to previous versions in the event that any problematic or erroneous changes are made. For Windows, TortiseSVN is used (http://tortoisesvn.net/). For Mac and Unix the "svn" command allows access to Subversion

trees. Documentation for the "svn" is available at http://svnbook.red-bean.com/nightly/en/index.html, and the subversion tree is located at https://subversion.assembla.com/svn/twopass/. New users must be invited by the administrator maintaining the Subversion tree to obtain a username and password.

A.3.2 Processing a raw data file

Table A.1 lists the separate files that make up the IOPPS code. These codes generate three files for every flight. The first file contains a series 4096 byte records that are generated for every frame recoded by the probe. The information about the variables in this file which are generated from the raw data file by *read_binary_image.m* are listed in Table A.3. After the particle image and timing information are output, *imgProc_dm.m* generates a netCDF file containing several descriptors (e.g. timing information, morphological description) of each particle as listed in Table A.2.

Table A.3. Variables generated for each 4096-byte frame

NetCDF Variable	Description
'data'	Raw image data of frame – uncompressed bytes with '0' representing shaded photodiodes and '1' representing unshaded photodiodes. At the end of the particle, there is a time slice in same format as specified in the probe manual
'day'	Day of frame
'hour'	Hour of frame [UTC]
'millisec'	Milliseconds of frame
'minute'	Minute of frame [UTC]
'month'	Month of frame
'second'	Second of frame [UTC]
'wkday'	Weekday of frame
'year'	Year of frame

The *imgProc_dm.m* also calls a routine *calculate_reject_unified.m* that accepts or rejects a particle based on a series of criteria designed to remove spurious stuck bits, splash artifacts, blank records, and streaker particles. Since *imgProc_dm.m* may take a few days to run for a single flight, a code is not placed to identify shattered artifacts at this stage so that the user can conduct sensitivity tests with the threshold interarrival times that are used to reject these artifacts when the size distributions are generated. After the netCDF file containing particle parameters is created, the routine *size_Dist.m* generates size distributions and the other bulk parameters listed in Table A.4. The rest of this section describes the manner in which the MATLAB routines are applied to generate the output files for a datafile.

A.3.3 Uncompressing the raw OAP data into a netCDF image file

There are many different formats that optical array probes use to store the data stream (i.e. time information and particle images). For the 2DC and 2DP, IOPPS can read the raw binary files generated by both Environment Canada and NCAR. For the 2DS, IOPPS can read in base*.2DS generated by the 2DS Playback software. For the HVPS, IOPPS can read in base *.HVPS files generated by the HVPS Playback software. For the CIP, IOPPS reads in the compressed files created by the Droplet Measurement Technologies software. In order routine *img_processing.m* to use these images, they must first be uncompressed into netCDF format readable by the.

This is done for a CIP file by typing the following command at the MATLAB prompt:

read_binary_image(*raw_file, output_file_name, project*)

where *raw_file* is the name of the raw OAP file, *output_file_name* is the name of the output uncompressed netCDF image file generated by the routine, and *project* is a flag that specifies the

manner in which the compressed images were stored, which varies according to the project being processed. For TWP-ICE, 0 is used. for ISDAC, 1, for RACORO and DYNAMO, 2. Additional projects that use compressed data formats other than those currently implemented can be incoroporated into *read_binary_image.m* by modifiying lines 69 to 123 of the code. *read_binary_image.m* can take anywhere from a few hours to a couple of days to complete depending on the number of particles, so it typically run overnight or through a batch job.

For converting raw 2DS files to a format readable by *img_processing.m*, the same steps as above are followed, except the program *read_binary_image_2ds.m* is used instead of *read_binary_image.m.* For HVPS data files, use *read_binary_image_hvps.m*. For 2DC and 2DP files supplied by NCAR, *read_binary_image_plows.m* is used. For raw 2DC and 2DP files supplied by Environment Canada, *read_binary_image_ec.m* is used. Any read_binary* file can be subsequently modified to read raw data formats encountered in field experiments that are not already implemented in the code in order to adapt IOPPS to read these formats.

A.3.4 Performing automated analysis of image netCDF files

The code that generates the particle parameters, *imgProc_dm.m*, has many inputs that depend on the field project and probe used, or on the manner in which the data are selected to be processed. An example of a file *run_img_processing.m* that calls *imgProc_dm.m* and sets the relevant processing options is listed below:

Example code: run_img_processing.m

```
% Specify the destination file name infilename and diodesize ds in
millimeters

ds = 0.025;                                    % Size of diode in
millimeters
infilename = 'caps.cdf';              % Input raw image netCDF file name
```

```
% Load netCDF file of uncompressed records
f = netcdf.open([infilename],'nowrite');
handles.f = f;
handles.hour = netcdf.getVar(f, netcdf.inqVarID(f,'hour'));
handles.minute = netcdf.getVar(f, netcdf.inqVarID(f,'minute'));
handles.second = netcdf.getVar(f, netcdf.inqVarID(f,'second'));
handles.millisec = netcdf.getVar(f, netcdf.inqVarID(f,'millisec'));

% Intialize handles structure - load necessary data from uncompressed netCDF
file
handles.img_count = length(handles.hour);
handles.diodesize = ds;
% Start at first image
handles.current_image = 1;

% Load true airspeed data - insert code here to load airspeeds and HHMMSS
% time into tas and timehhmmss from your data file

% Output file name
File = 'proc.cdf';          % Output image autoanalysis file
% What file stores the airspeeds?

% run the img_processing.m routine
handles = imgProc_dm(infile, File, 'CIP', 1, handles.img_count)
```

This file requires the uncompressed image netCDF as input. The *ds* variable stores the photodiode width in millimeters, so must be modified for the probe used in the relevant field experiment. If particle parameters from only a few frames are needed, *handles.img_count* can be modified to specify the frame which should be processed. Once the relevant fields in the *run_img_processing.m* and *img_processing.m* files have been modified, 'run_img_processing' is typed in the MATLAB command prompt to execute. The conversion process may take a day or two for very large files, so it is recommended that this script be run overnight or in batch mode on a capable supercomputer.

A.3.5 Deriving size distributions from the autoanalysis netCDF file

Once the particle properties have been determined, the final step in processing is to generate the particle size distributions. This is done using the *size_Dist.m* file. An input file giving the air speed, TAS, as a function of time at 1 Hz resolution is required by this code. A

number of different criteria are also applied to reject particles at this stage, and are specified in the good_particles variable on line 139 of the *size_Dist.m* file. For example, the following shows an example of the rejection criteria applied:

```
good_particles = (auto_reject == '0' | auto_reject == 'H' | auto_reject ==
'h' | auto_reject == 'u') & int_arr > 1e-4;
```

Here, *auto_reject* is the reject flag and *int_arr* is the threshold interarrival time in seconds that needs to be exceeded for a particle to be classified as naturally occurring. The methodology described in Chapter 2.3 should be followed to choose *int_arr* appropriate for each dataset. It is also possible to implement code that automates the selection of this threshold. This is done by adding a "for" loop before line 139 that applies the method used to remove shattered artifacts outline in Chapter 2 of this dissertation to every 5000 particles such as the code below.

```
      bins = logspace(-7, 0, 50);
% Specify range of normalized frequency histogram
      width = log(bins(2))-log(bins(1));
      num_particles = 5000;
% # of particles to factor into fit
      bimodalfit = @(tau, dt) (dt/tau(1)).*exp(-dt/tau(1));
% Equation from Chapter 2+3 for bimodal fit
for(i=1:num_particles:length(auto_reject))

   indicies = i:min([i+num_particles-1 length(int_arr)];
  arr = int_arr(indicies);
  rej = rej(indicies);
  h = hist(arr, bins(bins < 1));
  [tau_std] = abs(nlinfit(bins,h./sum(h)./width,bimodalfit,[1e-2 1e-5
0.5], statset('Robust', 'on')));        % Do fit to delta t distribution
    rej(arr < min(tau_std(1:2))*2) = 'S';
   auto_reject(indicies) = rej;
% Set shattered artifact flag to particles with dt < 2*tau2
end
```

The *auto_reject* flag provides options to accept particles if that were previously rejected if they meet certain criteria. Currerntly the following options are configured to include particles accepted by *calculate_reject_unified.m* (`auto_reject == '0'`) and both correctable ('h') and uncorrectable ('u') hollow particles (`auto_reject == 'H' | auto_reject == 'h' |`

`auto_reject == 'u'`). Other options can be configured in the code to include particles that were rejected due to aspect ratio (`auto_reject == 'a' | auto_reject == 't'`), rejected for having fewer than 25% of photodiodes shadowed in the rectangle encompassing the particle (`auto_reject == 'p'`), for being a split particle (`auto_reject == 's'`), or a particle with zero area (`auto_reject == 'z' | auto_reject == 'f'`).

Once the particle acceptance criteria are set the code is executed by calling the script:

```
sizeDist(infile, outfile, tas, timehhmmss, probename, d_choice,
SAmethod, Pres, Temp)
```

Here, *infile* is a string containing the path to the netCDF file containing the particle properties, *outfile* is the name of the netCDF output file containing the calculated size distribution, *tas* is a 1D array of true air speeds at the given UTC times, *probename* is the name of the probe being procseed (currently 'CIP', '2DS', 'HVPS', and '2DC.' are implemented), and *d_choice* is an option describing how particle maximum dimension is defined. The distributions in this thesis were generated using option 6 which means the maximum dimension was defined as maximum length chord through the particle. The other inputs are *pres*, is a 1D array of air pressure in hPA, and *temp*, a 1D array of temperatures in degrees Celsius. *SAmethod* specifies the method used to calculate sample area, where 0 assumes that only particles with their centers inside the photodiode array are accepted, 1 assumes that only particles entirely inside the photodiode array are accepted, and 2 assumes that the Heymsfield and Parish (1978) image reconstruction technique is used to estimate the maximum dimension of particles touching the edge of the photodiode array. It typically takes one to two hours to execute this step in IOPPS. A netCDF file that contains the size distributions that are described in Table A.4 is output by this step in IOPPS.

Table A.4. List of parameters generated in the size distribution data file

Table A.4. (cont.)

Parameter	Description
'ar'	Mean area ratio
'bin_dD'	Bin width in microns
'bin_max'	Maximum boundary for bin in microns
'bin_mid'	Midpoint for size bin in microns
'bin_min'	Minimum boundary for size bin in microns
'Calcd_area'	Cumulative particle area calculated from a-D relationships in mm^2
'conc_AreaR'	Size distribution in terms of area equivalent diameter in cm^{-3} μm^{-1}
'conc_minR'	Size distribution in terms of maximum dimension in cm^{-3} μm^{-1}
'habitmsd'	Mass distribution function in terms of maximum dimension and particle habit in g cm^{-3} μm^{-1}
'habitsd'	Size distribution in terms of maximum in dimension and particle habit cm^{-3} μm^{-1}
'mass'	IWC calculated from m-D relationships in g m^{-3}
'massBL'	IWC from Baker and Lawson [2006] g m^{-3}
'n'	Total number concentration cm^{-3}
'Prec_rate'	Precipitation rate mm h^{-1}
're'	Effective radius from Fu [1996] in μm
'reject_ratio'	% of particles rejected
'area'	Size distribution in terms of maximum dimension and area ratio in cm^{-3} μm^{-1}
'time'	UTC time [HHMMSS]
'total_area'	Total cross sectional area in mm^2
'vt'	Mass-weighted terminal velocity in m s^{-1}

A.4 Viewing size distributions and concentrations

Codes are also available to display size distributions and concentrations in various size ranges. To view a size distribution, the *conc_minR* or another equivalent size distribution variable which stores the 1 Hz size distributions normalized by bin width must be first loaded. The first dimension is time and the second dimension is the size bin. The bin minimum boundary

points *bin_min* array must also be loaded to use the MATLAB stairs routine to display the size distribution. A code to display the size distributon averaged over an entire flight is listed below:

```
%% Load data
f = netcdf.open('sd.cdf');                                          % Open SD file

conc = netcdf.getVar(f, netcdf.inqVarID(f, 'conc_minR'));          % Load SD
bin_min = netcdf.getVar(f, netcdf.inqVarID(f, 'bin_min'));        % Load bin
ninima
bin_max = netcdf.getVar(f, netcdf.inqVarID(f, 'bin_min'));         % Load bin
maxima

netcdf.close(f);     % Close netCDF

%% Display SD
figure;
stairs(bin_min, nanmean(conc,1), 'r', 'LineWidth', 2);
xlabel('D [\mum]');
ylabel('N(D) [cm^{-4}]');
```

Thereafter, the following code can be used to determine the total concentration within a given size range. In the example, *total_conc* is then a 1D array storing the 1 Hz concentration of particles with maximum dimension between 100 to 500 μm.

```
%% Calculate concentration of particles 0.1 to 0.5 mm
s = size(conc);
size_start = 0.1;
size_end = 0.5;
size_indicies = find(bin_min >= size_start && bin_max <= size_end);
total_conc = zeros(s(1),1);
for(i=1:length(size_indicies))
    total_conc = total_conc +
conc(:,size_indicies(i)).*bin_dD(size_indicies(i));
end
```

A.5. Viewing images

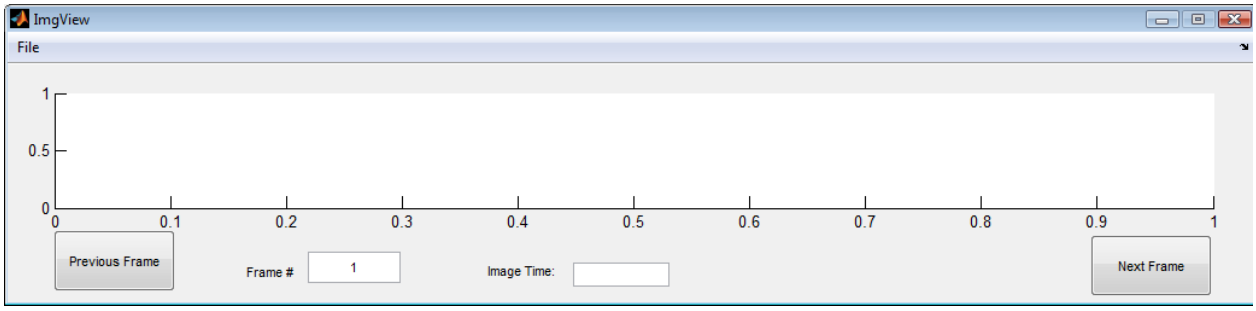


Figure A.2. Main screen of ImgView program

In addition to plotting concentrations and size distributions, it is important that there be a capability of viewing the two-dimensional particle images. The ImgView program. To execute the program, the working directory must be changed to the directory containing ImgView.fig Then, after typing 'ImgView' from the MATLAB prompt, a window that looks like Figure A.2 will be displayed on the screen. On the top left hand corner there is a menu "File" with three options "Open Image CDF," "Save Image As…," and "Exit." To view images from an

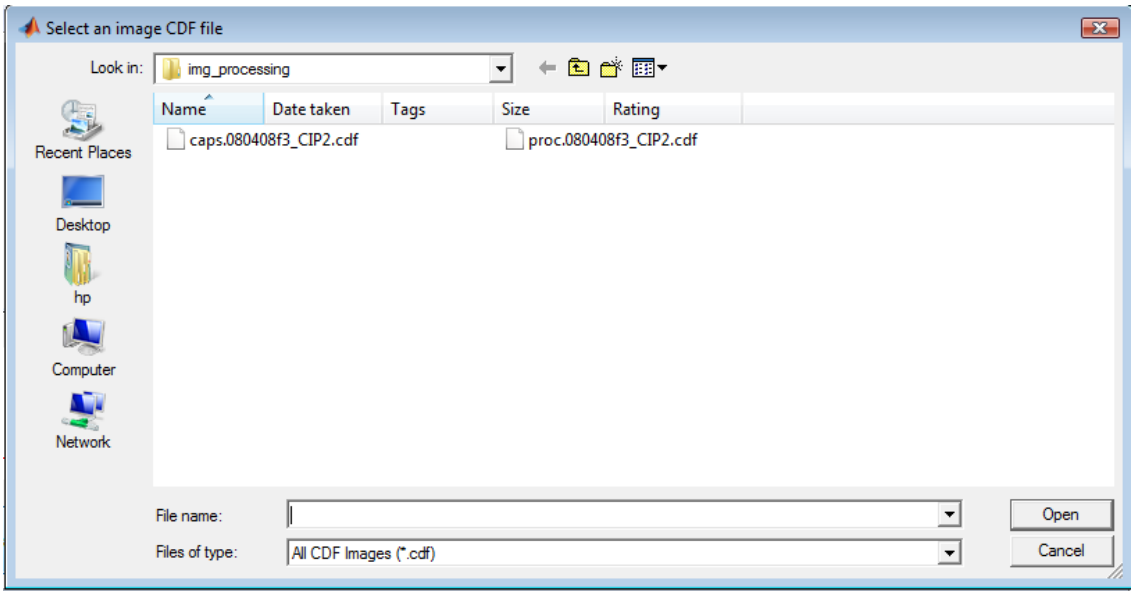


Figure A.3. Open file dialog box of ImgView.

uncompressed netCDF image file, "Open Image CDF" must be clicked. The program will then prompt for the name of the uncompressed netCDF image file in a window that looks like Figure A.3.

The name of the uncompressed netCDF image file must be clicked to open it. Then, the program will prompt for the name of the netCDF file that corresponds to the uncompressed netCDF image using a similar dialog box file. After selecting the appropriate filename, the images will be displayed on a screen that looks like Figure A.4.

Each vertical line indicates the end of a particle. Particles are displayed in black if they are not rejected, in red if they are rejected for any reason other than "correctable hollow," and in green if they are tagged as "correctable hollow." Advancing between particle frames is

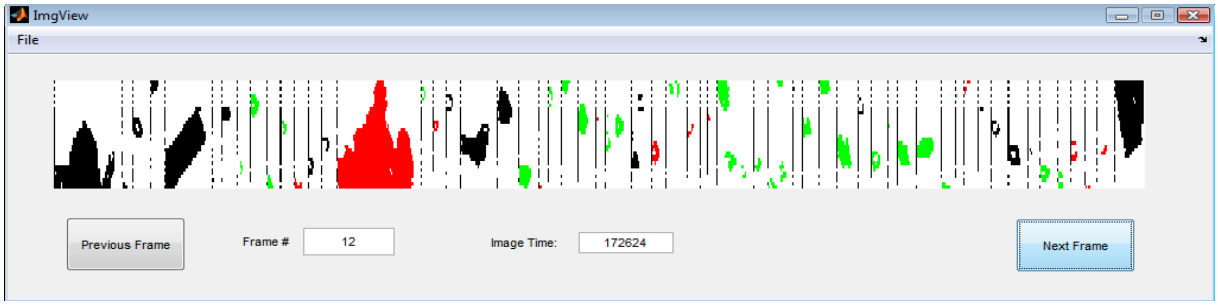


Figure A.4. ImgView main screen with a particle frame displayed.

accomplished by clicking "Next Frame." or "Previous Frame." The frame number to display can also be specified in the "Frame #" text box. A time to display can be specified in the "Image Time:" text box. The program will find the first image record on or after the time specified. To save the current image displayed as a JPEG file, click "File," then click "Save Image As…"

Sometimes, specifically designed plots of particle images are required. The imageview subroutine within ImgView.m can be used to create custom scripts to view 2D images to fit into specific figures as necessary. Imageview inputs in 5 parameters:

```
% imageview - Plot a 2D image record onto the given axis
%     axis_handle - handle pointing to axis you wish to draw the 2D image
%     on
%     imgcdf - netCDF file handle for uncompressed image netCDF
%     frame - # of record to draw
%     rec_num - 'parent_rec_num' from autoanalysis netCDF
%     reject - 'image_auto_reject' from autoanalysis netCDF
function imageview(axes_handle, imgcdf, frame, rec_num, reject)
```

axis_handle represents the MATLAB axis handle for displaying the image in. *imgcdf* identfies the corresponding to the uncompressed netCDF image file. *frame* is the frame number. *rec_num* stores the index of the 4,096 byte record that each individual particle is in. *reject* is an array containing the ASCII reject code for each individual particle. This example code below calls the

imageview routine to load the first image from a file and display it on the screen in a specific position:

```
imgcdf = netcdf.open('test.cdf');                          % Uncompressed
image netCDF file
proccdf = netcdf.open('proc.cdf');                         % Autoanalysis
netCDF file

% Load the 'image_auto_reject' and 'parent_rec_num' variables into memory
reject = netcdf.getVar(proccdf,netcdf.inqVarID(proccdf,
'image_auto_reject'));
rec_num = netcdf.getVar(proccdf,netcdf.inqVarID(proccdf, 'parent_rec_num'));

frame = 1;                                                 % Display first frame

figure('Position', [1 1 1700 100]);                        % Make the figure 1700 x
100 wide to fit entire record - adjust for your record sizes

set(gca, 'units', 'characters');                           % The current axis needs
to be set in characters so imageview can correctly adjust the axis size

imageview(gca, imgcdf, frame, rec_num, reject);            % Display the frame

% Close netCDF files
netcdf.close(imgcdf);
netcdf.close(proccdf);
```

After creating the image figure, the MATLAB figure editor can be used to make further changes.

A.6. MATLAB code

read_binary_2ds

```
function read_binary_2ds(infilename,outfilename)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%% Read the raw base*.2DS file, and then write into NETCDF file
%% Follow the SPEC manual
%%  by Will Wu, 08/01/2014
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

starpos = find(infilename == '*',1,'last');

if ~isempty(starpos)
    files = dir(infilename);
    filenums = length(files);
    filedir = infilename(1:starpos-1);
else
    filenums = 1;
end

for i = 1:filenums
    if filenums > 1
        infilename = [filedir,files(i).name];
    end

    if outfilename == '1'
        slashpos = find(infilename == '.',1,'last');
        outfilename = ['HVPS.',infilename(1:slashpos-1),'.cdf'];
    end

    outfilename1=[outfilename, '.2DS_H.cdf'];
    outfilename2=[outfilename, '.2DS_V.cdf'];

    fid=fopen(infilename,'r','l');

    f = netcdf.create(outfilename1, 'clobber');

    dimid0 = netcdf.defDim(f,'time',netcdf.getConstant('NC_UNLIMITED'));
    dimid1 = netcdf.defDim(f,'ImgRowlen',8);
    dimid2 = netcdf.defDim(f,'ImgBlocklen',1700);

    varid0 = netcdf.defVar(f,'year','short',dimid0);
    varid1 = netcdf.defVar(f,'month','byte',dimid0);
    varid2 = netcdf.defVar(f,'day','byte',dimid0);
    varid3 = netcdf.defVar(f,'hour','byte',dimid0);
    varid4 = netcdf.defVar(f,'minute','byte',dimid0);
    varid5 = netcdf.defVar(f,'second','byte',dimid0);
    varid6 = netcdf.defVar(f,'millisec','short',dimid0);
    varid7 = netcdf.defVar(f,'wkday','byte',dimid0);
    varid8 = netcdf.defVar(f,'data','int',[dimid1 dimid2 dimid0]);
    netcdf.endDef(f)
```

```
    f1 = netcdf.create(outfilename2, 'clobber');

    dimid01 = netcdf.defDim(f1,'time',netcdf.getConstant('NC_UNLIMITED'));
    dimid11 = netcdf.defDim(f1,'ImgRowlen',8);
    dimid21 = netcdf.defDim(f1,'ImgBlocklen',1700);

    varid01 = netcdf.defVar(f1,'year','short',dimid01);
    varid11 = netcdf.defVar(f1,'month','byte',dimid01);
    varid21 = netcdf.defVar(f1,'day','byte',dimid01);
    varid31 = netcdf.defVar(f1,'hour','byte',dimid01);
    varid41 = netcdf.defVar(f1,'minute','byte',dimid01);
    varid51 = netcdf.defVar(f1,'second','byte',dimid01);
    varid61 = netcdf.defVar(f1,'millisec','short',dimid01);
    varid71 = netcdf.defVar(f1,'wkday','byte',dimid01);
    varid81 = netcdf.defVar(f1,'data','int',[dimid11 dimid21 dimid01]);
    netcdf.endDef(f1)

    kk1=1;
    kk2=1;
    endfile = 0;
    nNext=1;
    dataprev=zeros(2048,1);

    %fseek(fid,4114*233191,'bof');
    while feof(fid)==0 && endfile == 0
        %tic
        [year,month, wkday,day, hour, minute, second, millisec, data,
discard]=readRecord(fid);
        [year,month,day, hour, minute, second, millisec]
        [year1,month1, wkday1,day1, hour1, minute1, second1, millisec1,
data1, discard1]=readRecord(fid);
        %fseek(fid,-4114,'cof');
        datan=[data' data1'];
        datan=datan';

        [imgH, imgV, nNext]=get_img(datan);
        sizeimg= size(imgH);
        if sizeimg(2)>1700
            imgH=imgH(:,1:1700);
            sizeimg(2)
        end

        sizeimg= size(imgV);
        if sizeimg(2)>1700
            imgV=imgV(:,1:1700);
            sizeimg(2)
        end

        if sum(sum(imgH))~=0
            for  mmm=1:8
                img1(mmm,1:1700)=sixteen2int(imgH((mmm-
1)*16+1:mmm*16,1:1700));
            end
```

```
            netcdf.putVar ( f, varid0, kk1-1, 1, year );
            netcdf.putVar ( f, varid1, kk1-1, 1, month );
            netcdf.putVar ( f, varid2, kk1-1, 1, day );
            netcdf.putVar ( f, varid3, kk1-1, 1, hour );
            netcdf.putVar ( f, varid4, kk1-1, 1, minute );
            netcdf.putVar ( f, varid5, kk1-1, 1, second );
            netcdf.putVar ( f, varid6, kk1-1, 1, millisec );
            netcdf.putVar ( f, varid7, kk1-1, 1, wkday );
            netcdf.putVar ( f, varid8, [0, 0, kk1-1], [8,1700,1], img1 );

            kk1=kk1+1;
            if mod(kk1,1000) == 0
                 ['kk1=' num2str(kk1) ', ' datestr(now)]
            end
        end

        if sum(sum(imgV))~=0
            for  mmm=1:8
                img2(mmm,1:1700)=sixteen2int(imgV((mmm-
1)*16+1:mmm*16,1:1700));
            end

            netcdf.putVar ( f1, varid01, kk2-1, 1, year );
            netcdf.putVar ( f1, varid11, kk2-1, 1, month );
            netcdf.putVar ( f1, varid21, kk2-1, 1, day );
            netcdf.putVar ( f1, varid31, kk2-1, 1, hour );
            netcdf.putVar ( f1, varid41, kk2-1, 1, minute );
            netcdf.putVar ( f1, varid51, kk2-1, 1, second );
            netcdf.putVar ( f1, varid61, kk2-1, 1, millisec );
            netcdf.putVar ( f1, varid71, kk2-1, 1, wkday );
            netcdf.putVar ( f1, varid81, [0, 0, kk2-1], [8,1700,1], img2 );

            kk2=kk2+1;
            if mod(kk2,1000) == 0
                 ['kk2=' num2str(kk2) ', ' datestr(now)]
            end
        end

        %for j=1:4115
        bb=fread(fid,1,'int8');
        if feof(fid) == 1
            endfile=1;
            break
        end
        %end
        fseek(fid,-4115,'cof');
        %kk
        %ftell(fid)
        %toc
    end

    fclose(fid);
    netcdf.close(f);
    netcdf.close(f1);
end
```

```
end

function [year,month, wkday,day, hour, minute, second, millisec, data,
discard]=readRecord(fid)

        year=fread(fid,1,'uint16');
        month=fread(fid,1,'uint16');
        wkday=fread(fid,1,'uint16');
        day=fread(fid,1,'uint16');
        hour=fread(fid,1,'uint16');
        minute=fread(fid,1,'uint16');
        second=fread(fid,1,'uint16');
        millisec=fread(fid,1,'uint16');
        data = fread(fid,2048,'uint16');
        discard=fread(fid,1,'uint16');
end

function [imgH, imgV, nNext]=get_img(buf)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%% Decompress the image
%% Follow the SPEC manual
%%  by Will Wu, 06/20/2013
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    imgH=zeros(128,1700);
    imgV=zeros(128,1700);
    iSlice=0;
    iii=1;
    while iii<=2048
        if 12883==buf(iii) %'0011001001010011'
              nH=bitand(buf(iii+1), 4095); %bin2dec('0000111111111111'));
              bHTiming=bitand(buf(iii+1), 4096);
%bin2dec('0001000000000000'))/2^13;
              nV=bitand(buf(iii+2), 4095); %bin2dec('0000111111111111'));
              bVTiming=bitand(buf(iii+2), 4096);
%bin2dec('0001000000000000'))/2^13;
              PC = buf(iii+3);
              nS = buf(iii+4);

              if nH~=0 && nV~=0
                  system(['echo ' num2str(nH)  ' ' num2str(nV) ' >>
output.txt']);
              end

              iii=iii+5;
              if bHTiming~=0 || bVTiming~=0
                  iii=iii+nH+nV;
              elseif nH~=0
                  jjj=1;
                  kkk=0;
                  while jjj<=nS && kkk<nH-2 % Last two slice is time
```

```
                    aa=bitand(buf(iii+kkk),16256)/2^7;
%bin2dec('0011111110000000')
                    bb=bitand(buf(iii+kkk),127);
%bin2dec('0000000001111111')
                    imgH(min(128,bb+1):min(aa+bb,128),iSlice+jjj)=1;
                    bBase=min(aa+bb,128);
                    kkk=kkk+1;
                    while( bitand(buf(iii+kkk),16384)==0  && kkk<nH-2) %
bin2dec('1000000000000000')
                        aa=bitand(buf(iii+kkk),16256)/2^7;
                        bb=bitand(buf(iii+kkk),127);

imgH(min(128,bBase+bb+1):min(bBase+aa+bb,128),iSlice+jjj)=1;
                        bBase=min(bBase+aa+bb,128);
                        kkk=kkk+1;
                    end
                    jjj=jjj+1;
                end
                iSlice=iSlice+nS+2;

                imgH(:,iSlice-
1)='1010101010101010101010101010101010101010101010101010101010101010101010101010
0101010101010101010101010101010101010101010101010101010101010101';
                imgH(:,iSlice-1)=imgH(:,iSlice-1)-48;
                imgH(:,iSlice)=1;
                tParticle=buf(iii+nH-2)*2^16+buf(iii+nH-1);
                imgH(:,iSlice)=dec2bin(tParticle,128)-48;
                iii=iii+nH;

            elseif nV~=0
                jjj=1;
                kkk=0;
                while jjj<=nS && kkk<nV-2 % Last two slice is time
                    aa=bitand(buf(iii+kkk),16256)/2^7;
%bin2dec('0011111110000000')
                    bb=bitand(buf(iii+kkk),127);
%bin2dec('0000000001111111')
                    imgV(min(128,bb+1):min(aa+bb,128),iSlice+jjj)=1;
                    bBase=min(aa+bb,128);
                    kkk=kkk+1;
                    while( bitand(buf(iii+kkk),16384)==0  && kkk<=nV-2) %
bin2dec('1000000000000000')
                        aa=bitand(buf(iii+kkk),16256)/2^7;
                        bb=bitand(buf(iii+kkk),127);

imgV(min(128,bBase+bb+1):min(bBase+aa+bb,128),iSlice+jjj)=1;
                        bBase=min(bBase+aa+bb,128);
                        kkk=kkk+1;
                    end
                    jjj=jjj+1;
                end
                iSlice=iSlice+nS+2;

                imgV(:,iSlice-
1)='1010101010101010101010101010101010101010101010101010101010101010101010101010
0101010101010101010101010101010101010101010101010101010101010101';
```

```
                imgV(:,iSlice-1)=imgV(:,iSlice-1)-48;
                imgV(:,iSlice)=1;
                tParticle=buf(iii+nV-2)*2^16+buf(iii+nV-1);
                imgV(:,iSlice)=dec2bin(tParticle,128)-48;
                iii=iii+nV;
            end

        else
            iii=iii+1;

        end
    end
    nNext=iii-2048;
end

function intres=sixteen2int(original)

intres=zeros(1,1700);
for i=1:16
    temp=original(i,:)*2^(16-i);
    intres=intres+temp;
end
end
```

MATLAB code – imgProc_dm

```
function imgProc_dm(infile, outfile, probename, n, nEvery)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  This function is the image processing part of OAP processing using
%  distributed memory parallisation. The function use one simple interface
%  for all probes.
%
%  Interface:
%    infile   :   The input file name
%    outfile  :   The output file name
%    probetype:   One of the following: '2DC','2DP','CIP','PIP','HVPS' and
'2DS'
%    n        :   The nth chuck to be processed.
%    nEvery   :   The individual chuck size. nChuck*nEvery shoudl equal the
%                 total frame number
%
%  Note other important variables used in the program
%    handles:  a structure to store information. It is convinient to use a
%          struture to store the global information rather than using
%          various varibles
%
%  Update Dates:
%    * Initially Written by Will Wu, 06/24/2013
%          imgprocdm(File,probetype,n)
%    * Updated by Will Wu, 10/11/2013
%          New function interface
%          imgprocdm(infile,outfile,probetype,n, nEvery) and updated
documentation.
%          This version is a major update to include all probes and simplify
%          the function interface significantly
%    * Updated by Will Wu, 07/10/2013
%          New function interface imgProc_dm(infile,outfile,probetype,n,
nEvery)
%          Output perimeter, rectangle length/width/angle and eclispe
%          length/width/angle
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Setting probe information according to probe type
%    use ProbeType to indicate three type of probes:
%       0: 2DC/2DP, 32 doides, boundary 85,
%       1: CIP/PIP, 64 doides, boundary 170
%       2: HVPS/2DS, 128 doides, boundary 170

switch probename
    case '2DC'
        boundary=[255 255 255 255];
        boundarytime=85;

        ds = 0.025;              % Size of diode in millimeters
        handles.diodesize = ds;
        handles.diodenum  = 32;  % Diode number
        handles.current_image = 1;
        probetype=0;
```

```
    case '2DP'
        boundary=[255 255 255 255];
        boundarytime=85;

        ds = 0.200;              % Size of diode in millimeters
        handles.diodesize = ds;
        handles.diodenum  = 32;  % Diode number
        handles.current_image = 1;
        probetype=0;

    case 'CIP'
        boundary=[170, 170, 170, 170, 170, 170, 170, 170];
        boundarytime=NaN;

        ds = 0.025;              % Size of diode in millimeters
        handles.diodesize = ds;
        handles.diodenum  = 64;  % Diode number
        handles.current_image = 1;
        probetype=1;

    case 'PIP'
        boundary=[170, 170, 170, 170, 170, 170, 170, 170];
        boundarytime=NaN;

        ds = 0.100;              % Size of diode in millimeters
        handles.diodesize = ds;
        handles.diodenum  = 64;  % Diode number
        handles.current_image = 1;
        probetype=1;

    case 'HVPS'
        boundary=[43690, 43690, 43690, 43690, 43690, 43690, 43690, 43690];
        boundarytime=0;

        ds = 0.150;              % Size of diode in millimeters
        handles.diodesize = ds;
        handles.diodenum  = 128; % Diode number
        handles.current_image = 1;
        probetype=2;

    case '2DS'
        boundary=[43690, 43690, 43690, 43690, 43690, 43690, 43690, 43690];
        boundarytime=0;

        ds = 0.010;              % Size of diode in millimeters
        handles.diodesize = ds;
        handles.diodenum  = 128; % Diode number
        handles.current_image = 1;
        probetype=2;
end

diodenum = handles.diodenum;
byteperslice = diodenum/8;
handles.disagree = 0;
```

```
%% Read the particle image files
handles.f = netcdf.open(infile,'nowrite');
[~, dimlen] = netcdf.inqDim(handles.f,2);
[~, handles.img_count] = netcdf.inqDim(handles.f,0);
size_mat = dimlen;
warning off all
diode_stats = zeros(1,diodenum);


%% Create output NETCDF file and variables
f = netcdf.create(outfile, 'clobber');
dimid0 = netcdf.defDim(f,'time',netcdf.getConstant('NC_UNLIMITED'));
dimid1 = netcdf.defDim(f,'pos_count',2);
dimid2 = netcdf.defDim(f,'bin_count',diodenum);

varid1 = netcdf.defVar(f,'Date','double',dimid0);
varid0  = netcdf.defVar(f,'Time','double',dimid0);
varid2  = netcdf.defVar(f,'msec','double',dimid0);
%varid3 = netcdf.defVar(f,'wkday','double',dimid0);
varid4  = netcdf.defVar(f,'position','double',[dimid1 dimid0]);
varid5  = netcdf.defVar(f,'particle_time','double',dimid0);
varid6  = netcdf.defVar(f,'particle_millisec','double',dimid0);
varid7  = netcdf.defVar(f,'particle_microsec','double',dimid0);
varid8  = netcdf.defVar(f,'parent_rec_num','double',dimid0);
varid9  = netcdf.defVar(f,'particle_num','double',dimid0);
varid10 = netcdf.defVar(f,'image_length','double',dimid0);
varid11 = netcdf.defVar(f,'image_width','double',dimid0);
varid12 = netcdf.defVar(f,'image_area','double',dimid0);
varid13 = netcdf.defVar(f,'image_longest_y','double',dimid0);
varid14 = netcdf.defVar(f,'image_max_top_edge_touching','double',dimid0);
varid15 = netcdf.defVar(f,'image_max_bottom_edge_touching','double',dimid0);
varid16 = netcdf.defVar(f,'image_touching_edge','double',dimid0);
varid17 = netcdf.defVar(f,'image_auto_reject','double',dimid0);
varid18 = netcdf.defVar(f,'image_hollow','double',dimid0);
varid19 = netcdf.defVar(f,'image_center_in','double',dimid0);
varid20 = netcdf.defVar(f,'image_axis_ratio','double',dimid0);
varid21 = netcdf.defVar(f,'image_diam_circle_fit','double',dimid0);
varid22 = netcdf.defVar(f,'image_diam_horiz_chord','double',dimid0);
varid23 = netcdf.defVar(f,'image_diam_horiz_chord_corr','double',dimid0);
varid24 = netcdf.defVar(f,'image_diam_following_bamex_code','double',dimid0);
varid25 = netcdf.defVar(f,'image_diam_vert_chord','double',dimid0);
varid26 = netcdf.defVar(f,'image_diam_minR','double',dimid0);
varid27 = netcdf.defVar(f,'image_diam_AreaR','double',dimid0);
varid45 = netcdf.defVar(f,'image_perimeter','double',dimid0);
varid46 = netcdf.defVar(f,'image_RectangleL','double',dimid0);
varid47 = netcdf.defVar(f,'image_RectangleW','double',dimid0);
varid67 = netcdf.defVar(f,'image_RectangleAngle','double',dimid0);
varid48 = netcdf.defVar(f,'image_EllipseL','double',dimid0);
varid49 = netcdf.defVar(f,'image_EllipseW','double',dimid0);
varid69 = netcdf.defVar(f,'image_EllipseAngle','double',dimid0);
varid28 = netcdf.defVar(f,'percent_shadow_area','double',dimid0);
varid29 = netcdf.defVar(f,'edge_at_max_hole','double',dimid0);
varid30 = netcdf.defVar(f,'max_hole_diameter','double',dimid0);
varid31 = netcdf.defVar(f,'part_z','double',dimid0);
varid32 = netcdf.defVar(f,'size_factor','double',dimid0);
varid33 = netcdf.defVar(f,'holroyd_habit','double',dimid0);
```

```
varid34 = netcdf.defVar(f,'area_hole_ratio','double',dimid0);
varid35 = netcdf.defVar(f,'inter_arrival','double',dimid0);
varid36 = netcdf.defVar(f,'bin_stats','double',dimid2);
netcdf.endDef(f)

%% Variabels initialization
kk=1;
w=-1;
wstart = 0;

time_offset_hr = 0;
time_offset_mn = 0;
time_offset_sec = 0;
time_offset_ms = 0;
timeset_flag = 0;



%% Processing nth chuck. Every chuck is nEvery frame
%% Analyze each individual particle images and Outpu the particle by particle
information
for i=((n-1)*nEvery+1):min(n*nEvery,handles.img_count)

    handles.year     =
netcdf.getVar(handles.f,netcdf.inqVarID(handles.f,'year'    ),i-1,1);
    handles.month    =
netcdf.getVar(handles.f,netcdf.inqVarID(handles.f,'month'  ),i-1,1);
    handles.day      =
netcdf.getVar(handles.f,netcdf.inqVarID(handles.f,'day'  ),i-1,1);
    handles.hour     =
netcdf.getVar(handles.f,netcdf.inqVarID(handles.f,'hour'    ),i-1,1);
    handles.minute   =
netcdf.getVar(handles.f,netcdf.inqVarID(handles.f,'minute'  ),i-1,1);
    handles.second   =
netcdf.getVar(handles.f,netcdf.inqVarID(handles.f,'second'  ),i-1,1);
    handles.millisec =
netcdf.getVar(handles.f,netcdf.inqVarID(handles.f,'millisec'),i-1,1);

    if mod(i,100) == 0
        [num2str(i),'/',num2str(handles.img_count), ', ',datestr(now)]
    end
    varid = netcdf.inqVarID(handles.f,'data');

    if probetype==0
        temp = netcdf.getVar(handles.f,varid,[0, 0, i-1], [4,1024,1]);
    else
        temp = netcdf.getVar(handles.f,varid,[0, 0, i-1], [8,1700,1]);
    end
    data(:,:) = temp';

    j=1;
    start=0;
    firstpart = 1;
```

```
%c=[dec2bin(data(:,1),8),dec2bin(data(:,2),8),dec2bin(data(:,3),8),dec2bin(da
ta(:,4),8)];

    while data(j,1) ~= -1 && j < size(data,1)
        % Calculate every particles
        if (isequal(data(j,:), boundary) && ( (isequal(data(j+1,1),
boundarytime) || probetype==1) ) )

            if start ==0
                start=j+2;
            else

                if probetype==0
                    if start+1 > (j-1)  % Remove Corrupted Data
                     break;
                    end
                else
                    if start > (j-1)  % Remove Corrupted Data
                     break;
                    end
                end

                 header_loc = j+1;
                 w=w+1;
                 %% Create binary image according to probe type

                 if probetype==0
                     ind_matrix(1:j-start-1,:) = data(start+1:j-1,:);  % 2DC
has 3 slices between particles (sync word timing word and end of particle
words)

c=[dec2bin(ind_matrix(:,1),8),dec2bin(ind_matrix(:,2),8),dec2bin(ind_matrix(:
,3),8),dec2bin(ind_matrix(:,4),8)];
                 elseif probetype==1
                     ind_matrix(1:j-start,:) = data(start:j-1,:);
                     c=[dec2bin(ind_matrix(:,1),8),
dec2bin(ind_matrix(:,2),8),dec2bin(ind_matrix(:,3),8),dec2bin(ind_matrix(:,4)
,8), ...
                     dec2bin(ind_matrix(:,5),8),
dec2bin(ind_matrix(:,6),8),dec2bin(ind_matrix(:,7),8),dec2bin(ind_matrix(:,8)
,8)];
                 elseif probetype==2
                     ind_matrix(1:j-start,:) = 65535 - data(start:j-1,:); % I
used 1 to indicate the illuminated doides for HVPS and 2DS
                     c=[dec2bin(ind_matrix(:,1),16),
dec2bin(ind_matrix(:,2),16),dec2bin(ind_matrix(:,3),16),dec2bin(ind_matrix(:,
4),16), ...
                     dec2bin(ind_matrix(:,5),16),
dec2bin(ind_matrix(:,6),16),dec2bin(ind_matrix(:,7),16),dec2bin(ind_matrix(:,
8),16)];
                 end

                 % Just to test if there is bad images, usually 0 area images
                 figsize = size(c);
                 if figsize(2)~=diodenum
```

```
                    disp('Not equal to doide number');
                    return
                end


                images.position(kk,:) = [start, j-1];
                parent_rec_num(kk)=i;
                particle_num(kk) = mod(kk,66536);
%hex2dec([dec2hex(data(start-1,7)),dec2hex(data(start-1,8))]);

                %  Get the particle time
                if probetype==0
                    bin_convert =
[dec2bin(data(header_loc,2),8),dec2bin(data(header_loc,3),8),dec2bin(data(hea
der_loc,4),8)];
                    part_time = bin2dec(bin_convert);       % Interarrival
time in tas clock cycles
                    tas2d =
netcdf.getVar(handles.f,netcdf.inqVarID(handles.f,'tas'),i-1, 1);
                    part_time = part_time/tas2d*0.025*0.001;

                    images.int_arrival(kk) = part_time;

                    if(firstpart == 1)
                        firstpart = 0;
                        start_hour = handles.hour;
                        start_minute = handles.minute;
                        start_second = handles.second;
                        start_msec = handles.millisec*10;
                        % First, we get the hours....
                        start_msec = start_msec;
                        start_microsec = 0;
                        time_offset_hr = 0;
                        time_offset_mn = 0;
                        time_offset_sec = 0;
                        time_offset_ms = 0;

                        part_hour(kk) = start_hour;
                        part_min(kk) = start_minute;
                        part_sec(kk) = start_second;
                        part_mil(kk) = start_msec;
                        part_micro(kk) = 0;
                    else
                        frac_time = part_time - floor(part_time);
                        frac_time = frac_time * 1000;
                        part_micro(kk) = part_micro(kk-1) + (frac_time -
floor(frac_time))*1000;
                        part_mil(kk) = part_mil(kk-1) + floor(frac_time);
                        part_sec(kk) = part_sec(kk-1) + floor(part_time);
                        part_min(kk) = part_min(kk-1);
                        part_hour(kk) = part_hour(kk-1);
                    end

                    part_sec(part_mil >= 1000) = part_sec(part_mil >= 1000) +
1;
```

```
                    part_mil(part_mil >= 1000) = part_mil(part_mil >= 1000) -
1000;

                    part_min(part_sec >= 60) = part_min(part_sec >= 60) + 1;
                    part_sec(part_sec >= 60) = part_sec(part_sec >= 60) - 60;

                    part_hour(part_min >= 60) = part_hour(part_min >= 60) +
1;
                    part_min(part_min >= 60) = part_min(part_min >= 60) - 60;
                    part_hour(part_hour >= 24) = part_hour(part_hour >= 24) -
24;
                elseif probetype==1
                    bin_convert = [dec2bin(data(start-
1,2),8),dec2bin(data(start-1,3),8),dec2bin(data(start-1,4),8), ...
                        dec2bin(data(start-1,5),8), dec2bin(data(start-
1,6),8)];

                    part_hour(kk) = bin2dec(bin_convert(1:5));
                    part_min(kk) = bin2dec(bin_convert(6:11));
                    part_sec(kk) = bin2dec(bin_convert(12:17));
                    part_mil(kk) = bin2dec(bin_convert(18:27));
                    part_micro(kk) = bin2dec(bin_convert(28:40))*125e-9;


                    time_in_seconds(kk) = part_hour(kk) * 3600 + part_min(kk)
* 60 + part_sec(kk) + part_mil(kk)/1000 + part_micro(kk);
                    if kk > 1
                        images.int_arrival(kk) = time_in_seconds(kk) -
time_in_seconds(kk-1);
                    else
                        images.int_arrival(kk) = time_in_seconds(kk);
                    end


                elseif probetype==2
                    part_time =
double(data(header_loc,7))*2^16+double(data(header_loc,8));       %
Interarrival time in tas clock cycles
                    part_micro(kk) = part_time;
                    part_mil(kk)   = 0;
                    part_sec(kk)   = 0;
                    part_min(kk)   = 0;
                    part_hour(kk)  = 0;
                    if(kk>1)
                        images.int_arrival(kk) = part_time-part_micro(kk-1);
                    else
                        images.int_arrival(kk) = 0;
                    end
                end

                slices_ver = length(start:j-1);

rec_time(kk)=double(handles.hour)*10000+double(handles.minute)*100+double(han
dles.second);
```

```
rec_date(kk)=double(handles.year)*10000+double(handles.month)*100+double(hand
les.day);
                rec_millisec(kk)=handles.millisec;
                %                 rec_wkday(kk)=handles.wkday(i);

                %% Determine the Particle Habit
                %  We use the Holroyd algorithm here
                handles.bits_per_slice = diodenum;
                diode_stats = diode_stats + sum(c=='1',1);
                csum = sum(c=='1',1);

                images.holroyd_habit(kk) = holroyd(handles,c);

                %% Determine if the particle is rejected or not
                %  Calculate the Particle Length, Width, Area, Auto Reject
                %  Status And more... See calculate_reject_unified()
                %  funtion for more information


[images.image_length(kk),images.image_width(kk),images.image_area(kk), ...

images.longest_y_within_a_slice(kk),images.max_top_edge_touching(kk),images.m
ax_bottom_edge_touching(kk),...
                    images.image_touching_edge(kk),
images.auto_reject(kk),images.is_hollow(kk),images.percent_shadow(kk),images.
part_z(kk),...

images.sf(kk),images.area_hole_ratio(kk),handles]=calculate_reject_unified(c,
handles,images.holroyd_habit(kk));

                images.max_hole_diameter(kk) = handles.max_hole_diameter;
                images.edge_at_max_hole(kk) = handles.edge_at_max_hole;

                max_horizontal_length = images.image_length(kk);
                max_vertical_length = images.longest_y_within_a_slice(kk);
                image_area = images.image_area(kk);

                diode_size= handles.diodesize;
                corrected_horizontal_diode_size = handles.diodesize;
                largest_edge_touching  =
max(images.max_top_edge_touching(kk), images.max_bottom_edge_touching(kk));
                smallest_edge_touching =
min(images.max_top_edge_touching(kk), images.max_bottom_edge_touching(kk));

                %% Calculate more size deciptor using more advanced
techniques
                %  See dropsize for more information

[images.center_in(kk),images.axis_ratio(kk),images.diam_circle_fit(kk),images
.diam_horiz_chord(kk),images.diam_vert_chord(kk),...
                    images.diam_horiz_mean(kk),
images.diam_spheroid(kk)]=dropsize(max_horizontal_length,max_vertical_length,
image_area...
```

```
,largest_edge_touching,smallest_edge_touching,diode_size,corrected_horizontal
_diode_size, diodenum);

                %% Calculate size deciptor using bamex code
                %  See dropsize_new for more information
                % images.diam_bamex(kk) = dropsize_new(c,
largest_edge_touching, smallest_edge_touching, diodenum,
corrected_horizontal_diode_size, handles.diodesize, max_vertical_length);

                %% Using OpenCV C program to calculate length, width and
radius. This
                %% Get diameter of the smallest-enclosing circle, rectangle
and ellipse
                %images.minR(kk)=particlesize_cgal(c);
                images.minR(kk)=CGAL_minR(c);
                images.AreaR(kk)=2*sqrt(images.image_area(kk)/3.1415926);  %
Calculate the Darea (area-equivalent diameter)
                images.Perimeter(kk)=ParticlePerimeter(c);
                [images.RectangleL(kk), images.RectangleW(kk),
images.RectangleAngle(kk)] = CGAL_RectSize(c);
                [images.EllipseL(kk), images.EllipseW(kk),
images.EllipseAngle(kk)]       = CGAL_EllipseSize(c);

                %% Get the area ratio using the DL=max(DT,DP), only observed
area are used
                if images.image_length(kk) > images.image_width(kk)
                    images.percent_shadow(kk) = images.image_area(kk) / (pi *
images.image_length(kk).^ 2 / 4);
                elseif images.image_width(kk) ~= 0
                    images.percent_shadow(kk) = images.image_area(kk) / (pi *
images.image_width(kk).^ 2 / 4);
                else
                    images.percent_shadow(kk) = 0;
                end

                start = j + 2;
                kk = kk + 1;
                clear c ind_matrix
            end
        end

        j = j + 1;
    end

    %% Write out the processed information on NETCDF
    if kk > 1

        netcdf.putVar ( f, varid0, wstart, w-wstart+1, rec_time(:) );
        netcdf.putVar ( f, varid1, wstart, w-wstart+1, rec_date(:) );
        netcdf.putVar ( f, varid2, wstart, w-wstart+1, rec_millisec(:) );
        %netcdf.putVar ( f, varid3, wstart, w-wstart+1, rec_wkday(:) );
        netcdf.putVar ( f, varid4, [0 wstart], [2 w-wstart+1],
images.position' );
```

```
        netcdf.putVar ( f, varid5, wstart, w-wstart+1,
part_hour(:)*10000+part_min(:)*100+part_sec(:) );
        netcdf.putVar ( f, varid6, wstart, w-wstart+1, part_mil(:) );
        netcdf.putVar ( f, varid7, wstart, w-wstart+1, part_micro(:) );
        netcdf.putVar ( f, varid8, wstart, w-wstart+1, parent_rec_num );
        netcdf.putVar ( f, varid9, wstart, w-wstart+1, particle_num(:) );
        netcdf.putVar ( f, varid10, wstart, w-wstart+1, images.image_length);
        netcdf.putVar ( f, varid11, wstart, w-wstart+1, images.image_width);
        netcdf.putVar ( f, varid12, wstart, w-wstart+1,
images.image_area*diode_size*diode_size);
        netcdf.putVar ( f, varid13, wstart, w-wstart+1,
images.longest_y_within_a_slice);
        netcdf.putVar ( f, varid14, wstart, w-wstart+1,
images.max_top_edge_touching);
        netcdf.putVar ( f, varid15, wstart, w-wstart+1,
images.max_bottom_edge_touching);
        netcdf.putVar ( f, varid16, wstart, w-wstart+1,
images.image_touching_edge-'0');
        netcdf.putVar ( f, varid17, wstart, w-wstart+1,
double(images.auto_reject));
        netcdf.putVar ( f, varid18, wstart, w-wstart+1, images.is_hollow);
        netcdf.putVar ( f, varid19, wstart, w-wstart+1, images.center_in);
        netcdf.putVar ( f, varid20, wstart, w-wstart+1, images.axis_ratio);
        netcdf.putVar ( f, varid21, wstart, w-wstart+1,
images.diam_circle_fit);
        netcdf.putVar ( f, varid22, wstart, w-wstart+1,
images.diam_horiz_chord);
        netcdf.putVar ( f, varid23, wstart, w-wstart+1,
images.diam_horiz_chord ./ images.sf);
        netcdf.putVar ( f, varid24, wstart, w-wstart+1,
images.diam_horiz_mean);
        netcdf.putVar ( f, varid25, wstart, w-wstart+1,
images.diam_vert_chord);
        netcdf.putVar ( f, varid26, wstart, w-wstart+1,
images.minR*diode_size);
        netcdf.putVar ( f, varid27, wstart, w-wstart+1,
images.AreaR*diode_size);
        netcdf.putVar ( f, varid45, wstart, w-wstart+1,
images.Perimeter*diode_size);
        netcdf.putVar ( f, varid46, wstart, w-wstart+1,
images.RectangleL*diode_size);
        netcdf.putVar ( f, varid47, wstart, w-wstart+1,
images.RectangleW*diode_size);
        netcdf.putVar ( f, varid67, wstart, w-wstart+1,
images.RectangleAngle);
        netcdf.putVar ( f, varid48, wstart, w-wstart+1,
images.EllipseL*diode_size);
        netcdf.putVar ( f, varid49, wstart, w-wstart+1,
images.EllipseW*diode_size);
        netcdf.putVar ( f, varid69, wstart, w-wstart+1, images.EllipseAngle);
        netcdf.putVar ( f, varid28, wstart, w-wstart+1,
images.percent_shadow);
        netcdf.putVar ( f, varid29, wstart, w-wstart+1,
images.max_hole_diameter);
        netcdf.putVar ( f, varid30, wstart, w-wstart+1,
images.edge_at_max_hole);
        netcdf.putVar ( f, varid31, wstart, w-wstart+1, images.part_z);
```

```
        netcdf.putVar ( f, varid32, wstart, w-wstart+1, images.sf);
        netcdf.putVar ( f, varid33, wstart, w-wstart+1,
double(images.holroyd_habit));
        netcdf.putVar ( f, varid34, wstart, w-wstart+1,
images.area_hole_ratio);
        netcdf.putVar ( f, varid35, wstart, w-wstart+1, images.int_arrival);
        netcdf.putVar ( f, varid36, diode_stats );

        wstart = w+1;
        kk = 1;
        clear rec_time rec_date rec_millisec part_hour part_min part_sec
part_mil part_micro parent_rec_num particle_num images
    end
    clear images
end
warning on all

netcdf.close(f);
end
```

MATLAB code – read_binary_hvps

```
function read_binary_hvps(infilename,outfilename)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%% Read the raw base*.HVPS file, and then write into NETCDF file
%% Follow the SPEC manual
%%  by Will Wu, 06/20/2013
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

starpos = find(infilename == '*',1,'last');

if ~isempty(starpos)
    files = dir(infilename);
    filenums = length(files);
    filedir = infilename(1:starpos-1);
else
    filenums = 1;
end

for i = 1:filenums
    if filenums > 1
        infilename = [filedir,files(i).name];
    end

    if outfilename == '1'
        slashpos = find(infilename == '.',1,'last');
        outfilename = ['HVPS.',infilename(1:slashpos-1),'.cdf'];
    end

    fid=fopen(infilename,'r','l');
    f = netcdf.create(outfilename, 'clobber');

    dimid0 = netcdf.defDim(f,'time',netcdf.getConstant('NC_UNLIMITED'));
    dimid1 = netcdf.defDim(f,'ImgRowlen',8);
    dimid2 = netcdf.defDim(f,'ImgBlocklen',1700);

    varid0 = netcdf.defVar(f,'year','short',dimid0);
    varid1 = netcdf.defVar(f,'month','byte',dimid0);
    varid2 = netcdf.defVar(f,'day','byte',dimid0);
    varid3 = netcdf.defVar(f,'hour','byte',dimid0);
    varid4 = netcdf.defVar(f,'minute','byte',dimid0);
    varid5 = netcdf.defVar(f,'second','byte',dimid0);
    varid6 = netcdf.defVar(f,'millisec','short',dimid0);
    varid7 = netcdf.defVar(f,'wkday','byte',dimid0);
    varid8 = netcdf.defVar(f,'data','int',[dimid1 dimid2 dimid0]);
    netcdf.endDef(f)

    kk=1;
    endfile = 0;
    nNext=1;
    dataprev=zeros(2048,1);
```

```
    %fseek(fid,4114*233191,'bof');
    while feof(fid)==0 && endfile == 0
        %tic
        [year,month, wkday,day, hour, minute, second, millisec, data,
discard]=readRecord(fid);
        [year,month,day, hour, minute, second, millisec]
        [year1,month1, wkday1,day1, hour1, minute1, second1, millisec1,
data1, discard1]=readRecord(fid);
        %fseek(fid,-4114,'cof');
        datan=[data' data1'];
        datan=datan';

        [img, nNext]=get_img(datan);
        sizeimg= size(img);
        if sizeimg(2)>1700
            img=img(:,1:1700);
            sizeimg(2)
        end

        if sum(sum(img))~=0
            for  mmm=1:8
                img1(mmm,1:1700)=sixteen2int(img((mmm-
1)*16+1:mmm*16,1:1700));
            end

            netcdf.putVar ( f, varid0, kk-1, 1, year );
            netcdf.putVar ( f, varid1, kk-1, 1, month );
            netcdf.putVar ( f, varid2, kk-1, 1, day );
            netcdf.putVar ( f, varid3, kk-1, 1, hour );
            netcdf.putVar ( f, varid4, kk-1, 1, minute );
            netcdf.putVar ( f, varid5, kk-1, 1, second );
            netcdf.putVar ( f, varid6, kk-1, 1, millisec );
            netcdf.putVar ( f, varid7, kk-1, 1, wkday );
            netcdf.putVar ( f, varid8, [0, 0, kk-1], [8,1700,1], img1 );

            kk=kk+1;
            if mod(kk,1000) == 0
                 ['kk=' num2str(kk) ', ' datestr(now)]
            end
        end

        %for j=1:4115
        bb=fread(fid,1,'int8');
        if feof(fid) == 1
            endfile=1;
            break
        end
        %end
        fseek(fid,-4115,'cof');
        kk
        ftell(fid)
        %toc
    end

    fclose(fid);
    netcdf.close(f);
```

```
end


end

function [year,month, wkday,day, hour, minute, second, millisec, data,
discard]=readRecord(fid)

        year=fread(fid,1,'uint16');
        month=fread(fid,1,'uint16');
        wkday=fread(fid,1,'uint16');
        day=fread(fid,1,'uint16');
        hour=fread(fid,1,'uint16');
        minute=fread(fid,1,'uint16');
        second=fread(fid,1,'uint16');
        millisec=fread(fid,1,'uint16');
        data = fread(fid,2048,'uint16');
        discard=fread(fid,1,'uint16');
end

function [img, nNext]=get_img(buf)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%% Decompress the image
%% Follow the SPEC manual
%%  by Will Wu, 06/20/2013
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    img=zeros(128,1700);
    iSlice=0;
    iii=1;
    while iii<=2048
        if 12883==buf(iii) %'0011001001010011'
              nH=bitand(buf(iii+1), 4095); %bin2dec('0000111111111111'));
              bHTiming=bitand(buf(iii+1), 4096);
%bin2dec('0001000000000000'))/2^13;
              nV=bitand(buf(iii+2), 4095); %bin2dec('0000111111111111'));
              bVTiming=bitand(buf(iii+2), 4096);
%bin2dec('0001000000000000'))/2^13;
              PC = buf(iii+3);
              nS = buf(iii+4);

              if bVTiming~=0 || nV==0
                  iii=iii+nV+5;
              else
                  jjj=1;
                  kkk=1;
                  while jjj<=nS && kkk<=nV-2 % Last two slice is time
                      aa=bitand(buf(iii+4+kkk),16256)/2^7;
%bin2dec('0011111110000000')
                      bb=bitand(buf(iii+4+kkk),127);
%bin2dec('0000000001111111')
                      img(min(128,bb+1):min(aa+bb,128),iSlice+jjj)=1;
                      bBase=min(aa+bb,128);
```

```
                    kkk=kkk+1;
                    while( bitand(buf(iii+4+kkk),16384)==0  && kkk<=nV-2) %
bin2dec('1000000000000000')
                        aa=bitand(buf(iii+4+kkk),16256)/2^7;
                        bb=bitand(buf(iii+4+kkk),127);

img(min(128,bBase+bb+1):min(bBase+aa+bb,128),iSlice+jjj)=1;
                        bBase=min(bBase+aa+bb,128);
                        kkk=kkk+1;
                    end
                    jjj=jjj+1;
                end
                iSlice=iSlice+nS+2;

                img(:,iSlice-
1)='1010101010101010101010101010101010101010101010101010101010101010101
0101010101010101010101010101010101010101010101010101010101010101010';
                img(:,iSlice-1)=img(:,iSlice-1)-48;
                img(:,iSlice)=1;
                tParticle=buf(iii+nV+3)*2^16+buf(iii+nV+4);
                img(:,iSlice)=dec2bin(tParticle,128)-48;
                iii=iii+nV+5;
            end

        else
            iii=iii+1;

        end
    end
    nNext=iii-2048;
end

function intres=sixteen2int(original)

intres=zeros(1,1700);
for i=1:16
    temp=original(i,:)*2^(16-i);
    intres=intres+temp;
end
end
```

MATLAB code – lwc_calc

```
function lwc = lwc_calc(conc,bins)

% calculates lwc in g/m3, needs conc in cm^-3 and bins in microns

%rho_i = 0.91;
lwc = sum(conc(:,:) .* pi/6 * (bins(:)*(1e-4)) .^3,2)*1e6;
```

MATLAB code – calculate_reject_unified

```
function
[p_length,width,area,longest_y,max_top,max_bottom,touching_edge,reject_status
,is_hollow,percent_shadow_area,part_z,size_factor,area_hole_ratio,handles]=ca
lculate_reject_unified(image_buffer,handles,habit)

% /*  RETURN CODE                                                 */
% /* 0 = not rejected                                             */
% /* 'a' = reject max. aspect ratio                              */
% /* 't' = reject max. aspect ratio touch edg                        */
% /* 'p' = reject percent shadowed area                          */
% /*    'h' = reject Hollow image                                 */
% /*    's' = reject split image                                  */
% /*    'z' = reject 0 area image                                 */
% /*    'f' = reject fake 0 area image                            */


z_d = 0 : .05 : 8.15;


part_z = -1;
size_factor = 1;


edge_0 = [1.000, 1.054, 1.083, 1.101, 1.095, 1.110, 1.148, 1.162, 1.155,
1.123, ...
    1.182, 1.121, 1.162, 1.210, 1.242, 1.134, 1.166, 1.202, 1.238, 1.270, ...
    1.294, 1.278, 1.130, 1.148, 1.170, 1.194, 1.218, 1.242, 1.265, 1.288, ...
    1.310, 1.331, 1.351, 1.369, 1.386, 1.400, 1.411, 1.416, 1.407, 1.074, ...
    1.080, 1.087, 1.096, 1.106, 1.117, 1.127, 1.139, 1.150, 1.162, 1.173, ...
    1.185, 1.197, 1.208, 1.220, 1.232, 1.243, 1.255, 1.266, 1.277, 1.289, ...
    1.300, 1.311, 1.322, 1.333, 1.344, 1.355, 1.366, 1.376, 1.387, 1.397, ...
    1.407, 1.418, 1.428, 1.438, 1.448, 1.458, 1.467, 1.477, 1.486, 1.496, ...
    1.505, 1.515, 1.524, 1.533, 1.542, 1.551, 1.559, 1.568, 1.577, 1.585, ...
    1.594, 1.602, 1.610, 1.618, 1.626, 1.634, 1.642, 1.650, 1.657, 1.665, ...
    1.673, 1.680, 1.687, 1.694, 1.702, 1.709, 1.716, 1.722, 1.729, 1.736, ...
    1.742, 1.749, 1.755, 1.761, 1.768, 1.774, 1.780, 1.786, 1.791, 1.797, ...
    1.803, 1.808, 1.813, 1.819, 1.824, 1.829, 1.834, 1.839, 1.843, 1.848, ...
    1.852, 1.857, 1.861, 1.865, 1.869, 1.872, 1.876, 1.880, 1.883, 1.886, ...
    1.889, 1.892, 1.895, 1.897, 1.899, 1.901, 1.903, 1.905, 1.906, 1.907, ...
    1.908, 1.908, 1.908, 1.908, 1.907, 1.905, 1.903, 1.900, 1.897, 1.892, ...
    1.885, 1.877, 1.865, 1.845];


spot_edge = [0.003, 0.008, 0.017, 0.024, 0.033, 0.040, 0.047, 0.054, 0.062,
0.072, ...
    0.076, 0.088, 0.093, 0.096, 0.101, 0.119, 0.123, 0.127, 0.130, 0.134, ...
    0.139, 0.148, 0.175, 0.180, 0.184, 0.188, 0.192, 0.195, 0.199, 0.202, ...
    0.206, 0.209, 0.213, 0.217, 0.221, 0.225, 0.230, 0.235, 0.243, 0.327, ...
    0.334, 0.340, 0.345, 0.351, 0.355, 0.360, 0.365, 0.369, 0.373, 0.377, ...
    0.381, 0.385, 0.389, 0.393, 0.397, 0.400, 0.404, 0.408, 0.411, 0.415, ...
    0.419, 0.422, 0.426, 0.429, 0.433, 0.436, 0.439, 0.443, 0.446, 0.450, ...
    0.453, 0.457, 0.460, 0.463, 0.467, 0.470, 0.473, 0.477, 0.480, 0.484, ...
    0.487, 0.490, 0.494, 0.497, 0.501, 0.504, 0.507, 0.511, 0.514, 0.518, ...
    0.521, 0.525, 0.528, 0.532, 0.535, 0.539, 0.543, 0.547, 0.550, 0.554, ...
```

```
    0.558, 0.562, 0.566, 0.569, 0.572, 0.575, 0.578, 0.581, 0.584, 0.587, ...
    0.590, 0.593, 0.596, 0.598, 0.601, 0.605, 0.610, 0.614, 0.618, 0.623, ...
    0.627, 0.631, 0.635, 0.640, 0.644, 0.648, 0.653, 0.657, 0.662, 0.666, ...
    0.671, 0.676, 0.680, 0.685, 0.690, 0.695, 0.700, 0.705, 0.711, 0.716, ...
    0.721, 0.727, 0.733, 0.738, 0.744, 0.751, 0757, 0.763, 0.770, 0.777, ...
    0.784, 0.792, 0.800, 0.808, 0.817, 0.826, 0.836, 0.846, 0.858, 0.870, ...
    0.884, 0.901, 0.921, 0.950];



% temp =
[dec2bin(image_buffer(:,1),16),dec2bin(image_buffer(:,2),16),dec2bin(image_bu
ffer(:,3),16),...
%
dec2bin(image_buffer(:,4),16),dec2bin(image_buffer(:,5),16),dec2bin(image_buf
fer(:,6),16),...
%     dec2bin(image_buffer(:,7),16),dec2bin(image_buffer(:,8),16)];
% clear image_buffer
% image_buffer(:,:)=temp(:,:);

n_size=size(image_buffer);
n_slices=n_size(1);


handles.rej_zero_area = 1;
handles.rej_split = 1;
handles.rej_hollow = 1;
handles.bits_per_slice = n_size(2);
handles.shadowed_area = 25;
handles.max_edge_img_ar = 6;
handles.max_comp_img_ar = 5;

handles.max_hole_diameter = 0;
handles.edge_at_max_hole = 0;



min_length=-1;
max_length=-1;
max_width=1;
min_width=n_size(2);

total_area=0;

touch=0;
width = 0;
ndrops=0;
split=0;
hollow=0;
met_image=0;
is_hollow=0;

aspect_ratio=0;
```

```
percent_shadow_area=0;
area_hole_ratio = 0;


area=0;
max_top=0;
max_bottom=0;
longest_y=0;
touching_edge=0;

top_min_x=-1;
top_max_x=-1;
bottom_min_x=-1;
bottom_max_x=-1;

if n_slices==0
    p_length=0;
    area=0;



    if handles.rej_zero_area==1
        reject_status='z';
        return;
    else
        reject_status='0';
    end

else

    for i=1:n_slices


[min_pos_lite,max_pos_lite,n_lite]=scan_slice(image_buffer(i,:),handles);

        if longest_y < n_lite
            longest_y=n_lite;
        end

        if i>1
            vertical_split=vertical_split & image_buffer(i,:);
        else
            vertical_split=image_buffer(i,:);
        end

        if n_lite>0
            if touch==0
                if max_pos_lite==handles.bits_per_slice || min_pos_lite==1
                    touch=1;
                end
            end

            if min_pos_lite==1
                if bottom_min_x==-1
```

```
                bottom_min_x=i;
            end
            if bottom_max_x<i
                bottom_max_x=i;
            end
        end

        if max_pos_lite==handles.bits_per_slice
            if top_min_x==-1
                top_min_x=i;
            end
            if top_max_x<i
                top_max_x=i;
            end

        end

        if max_pos_lite > max_width
            max_width=max_pos_lite;
        end
        if min_pos_lite < min_width
            min_width=min_pos_lite;
        end

        if min_length == -1
            min_length = i;
            max_length = i;
        else
            max_length = i;
        end


        total_area=n_lite+total_area;
    end

    if met_image == 0 & n_lite > 0
        met_image=1;
        ndrops=ndrops+1;
        if ndrops > 1 & handles.rej_split==1
            split=1;
        end
    elseif met_image==1 & n_lite == 0
        met_image=0;
    end

end


area=total_area;
if top_min_x == -1
    max_top = 0;
else
    max_top = (top_max_x - top_min_x) + 1;
end
if bottom_min_x == -1
```

```
        max_bottom = 0;
    else
        max_bottom = (bottom_max_x - bottom_min_x) + 1;
    end
    if touch == 1
        touching_edge = 't';
    else
        touching_edge = '0';
    end



    if total_area == 0;
        p_length = 0;
        width = 0;
    else
        p_length = max_length - min_length + 1;
        width = max_width - min_width + 1;
    end

    if total_area > .8 * handles.bits_per_slice * n_slices
        reject_status = 'A';
        return;
    end

    if split == 1
        reject_status = 's';
        return;
    end

    if exist('vertical_split') == 1

[min_pos_lite,max_pos_lite,n_lite]=scan_slice(vertical_split,handles);
    else
        min_pos_lite=0;
        max_pos_lite=0;
        n_lite=0;
    end
    if n_lite == 1 & n_lite ~= max_pos_lite - min_pos_lite + 1 &
handles.rej_split == 0
        reject_status = 's';
        return;
    end

    if total_area > 0
        if p_length > 0 & width > 0
            aspect_ratio =  p_length / width;
            percent_shadow_area = total_area / (p_length * width ) * 100;
        else
            aspect_ratio = 0;
            percent_shadow_area = 0;
        end
    else
        aspect_ratio = 0;
        percent_shadow_area = 0;
```

```
    end

    if total_area == 0 && handles.rej_zero_area == 1
        reject_status = 'z';
        return;
    elseif total_area == 0 && handles.rej_zero_area == 0
        reject_status = '0';
        return;
    elseif ( aspect_ratio > handles.max_comp_img_ar || aspect_ratio <
1/handles.max_comp_img_ar ) % Second critirion added on Dec 2nd, 2013 by Will
for small aspect ratio
        reject_status = 'a';
        return;
    elseif touch == 1 && aspect_ratio > handles.max_edge_img_ar
        reject_status = 't';
        return;
    elseif percent_shadow_area < handles.shadowed_area
        reject_status = 'p';
        return;
    elseif handles.rej_hollow == 1

[hollow_status,edge_at_max_hole,max_hole_diameter]=is_it_hollow(image_buffer(
1:n_slices,:),n_slices,handles);

[hollow_status2,edge_at_max_hole2,max_hole_diameter2]=is_it_hollow(image_buff
er(n_slices:-1:1,:),n_slices,handles);


[hollow_status_side1,edge_at_max_hole_side1,max_hole_diameter_side1]=is_it_ho
llow_sidescan(image_buffer(1:n_slices,:)',n_slices,handles);

[hollow_status_side2,edge_at_max_hole_side2,max_hole_diameter_side2]=is_it_ho
llow_sidescan(image_buffer(1:n_slices,32:-1:1)',n_slices,handles);



        if hollow_status ~= hollow_status2
            hollow_status;
%             handles.disagree = handles.disagree + 1;
        end
%
%         if hollow_status == 1 & hollow_status2 == 0
%             hollow_status = 0;
%         end



if hollow_status + hollow_status2 == 1 & (habit == 's' | habit == 'h' | habit
== 'i' | habit == 't')
%if hollow_status + hollow_status2 == 1
    hollow_status = 1;
elseif habit == 'd' & percent_shadow_area < 35 & hollow_status +
hollow_status2 == 1
    hollow_status = 1;
elseif hollow_status == 1 & hollow_status2 == 0
    hollow_status = 0;
```

```
end

if percent_shadow_area < 30
    hollow_status = 0;
end


        if hollow_status == 1
            %         ratio = max_hole_diameter./(max_width-min_width+1);
            if edge_at_max_hole <= 0
                ratio = 0;
            else
                ratio = max_hole_diameter./edge_at_max_hole;
            end
            if ratio == 0
                part_z = 0;
                size_factor = 1;
                reject_status = 'h';
                area_hole_ratio = 0;
            elseif max_hole_diameter <= 1
                part_z = 0;
                size_factor = 1;
                area_hole_ratio = 0;
                reject_status = '0';
            else
                part_z = z_d(find(spot_edge < ratio,1,'last'));
                size_factor = edge_0(find(z_d <= part_z,1,'last'));
                reject_status = 'H';
                area_hole_ratio = area/max_hole_diameter;

                if hollow_status_side1 + hollow_status_side2 < 1
                    part_z = 0;
                    size_factor = 1;
                    reject_status = 'i';
                end

                if area_hole_ratio > 20 & habit == 'i'
                    part_z = 0;
                    size_factor = 1;
                    reject_status = 'u';
                elseif area_hole_ratio > 35 & habit == 'h'
                    part_z = 0;
                    size_factor = 1;
                    reject_status = 'u';
                elseif area_hole_ratio > 40
                    part_z = 0;
                    size_factor = 1;
                    reject_status = 'u';
                end

            end
            handles.edge_at_max_hole = edge_at_max_hole;
            handles.max_hole_diameter = max_hole_diameter;
            return
```

```
        else
            ratio = -1;
            part_z = -1;
            size_factor = 1;
            area_hole_ratio = 0;

        end
%         if hollow_status ==1
%             reject_status='h';
%             return
%         end
    end
    reject_status = '0';
    return


end

function [min_pos_lite,max_pos_lite,n_lite]=scan_slice(image_buf,handles)

n_lite=0;
max_pos_lite=0;
min_pos_lite=0;


zeros = find(image_buf == '0');
n_lite = length(zeros);
if n_lite == 0
    return
else
    min_pos_lite = zeros(1);
    max_pos_lite = zeros(n_lite);
end

% for i=1:handles.bits_per_slice
%     if image_buf(i) == '0'
%         n_lite=n_lite+1;
%         if min_pos_lite==0
%             min_pos_lite=i;
%             max_pos_lite=i;
%         else
%             max_pos_lite=i;
%         end
%     end
% end
return

function [status,edge_at_max_hole,max_hole_diameter] =
is_it_hollow(image_buf,slices,handles)

current = 0;
old = 0;
new = 0;

possibly_hollow = 0;
```

```
max_hole_diameter = 0;
edge_at_max_hole = 0;
status = 0;

start_img = 0;
end_img = 0;
i = 1;
while end_img == 0
    zero_amt = sum(image_buf(i,:) == '0');
    if zero_amt > 0 & start_img == 0
        start_img = i;
    end
    if zero_amt == 0 & start_img > 0
        end_img = i;
    end
    i = i + 1;
    if i > slices
        if start_img == 0
            start_img = 1;
        end
        if end_img == 0
            end_img = slices;
        end
    end

end

slices = end_img-start_img+1;

for i=start_img:end_img

    [min_pos_lite,max_pos_lite,n_lite]=scan_slice(image_buf(i,:),handles);

    num_empty = max_pos_lite - min_pos_lite + 1 - n_lite;

    if slices > 6
        slices_third = floor(slices/3);
    else
        slices_third = 1;
    end


    if num_empty > max_hole_diameter & i > slices_third & i < slices -
slices_third
        max_hole_diameter = num_empty;
        edge_at_max_hole = max_pos_lite - min_pos_lite + 1;
    end

    if possibly_hollow == 1 & status == 0
        if n_lite > 0 & n_lite ~= max_pos_lite - min_pos_lite + 1
            new = bin2dec(image_buf(i,17:32)) +
bitshift(bin2dec(image_buf(i,1:16)),16);
            olddec = bin2dec(old(17:32)) + bitshift(bin2dec(old(1:16)),16);
```

```
            newandold = bitand(new , olddec);
            if newandold == zeros

                status=1;
%                 return
            else
                old = mask_start_end(max_pos_lite, min_pos_lite,
image_buf(i,:),handles.bits_per_slice);
            end

        elseif n_lite > 0
            bufdec = bin2dec(image_buf(i,17:32)) +
bitshift(bin2dec(image_buf(i,1:16)),16);
            olddec = bin2dec(old(17:32)) + bitshift(bin2dec(old(1:16)),16);

            bufdec1 = bin2dec(image_buf(i,1:16));
            olddec1 = bin2dec(old(1:16));

            bufdec2 = bin2dec(image_buf(i,17:32));
            olddec2 = bin2dec(old(17:32));

            bufandold1 = bitand(bufdec1,olddec1);
            bufandold2 = bitand(bufdec2,olddec2);

            hole_size = length(find(old == '1'));
            cover_size = length(find(dec2bin(bufandold1) == '1')) +
length(find(dec2bin(bufandold2) == '1'));



%
%             bufandold = bitand(bufdec , olddec);
%             hole_size = length(find(old == '1'));
%             cover_size = length(find(dec2bin(bufandold) == '1'));
%
%



            if bufandold1 + bufandold2 > 0
%                 if cover_size <= 2 & hole_size ~=1
if cover_size <= .65*hole_size
                    status = 1;
%                     return;
                end
                possibly_hollow = 0;
                old = 0;
            elseif i > 1
                status = 1;
%                 return;
            else
                possibly_hollow = 0;
                old = 0;
            end
```

```
        else
            possibly_hollow = 0;

        end
    elseif status == 0
        if n_lite > 0 & n_lite ~= max_pos_lite - min_pos_lite + 1
            old = mask_start_end(max_pos_lite, min_pos_lite,
image_buf(i,:),handles.bits_per_slice);

            possibly_hollow = 1;
        end
    end
end
return;

function [status,edge_at_max_hole,max_hole_diameter] =
is_it_hollow_sidescan(image_buf,slices,handles)

current = 0;
old = 0;
new = 0;

possibly_hollow = 0;

max_hole_diameter = 0;
edge_at_max_hole = 0;
status = 0;

im_width = size(image_buf);

if im_width(2) > 32
    im_width(2) = 32;
end


start_img = 0;
end_img = 0;
i = 1;


while end_img == 0
    zero_amt = sum(image_buf(i,:) == '0');
    if zero_amt > 0 && start_img == 0
        start_img = i;
    end
    if zero_amt == 0 && start_img > 0
        end_img = i;
    end
    i = i + 1;
    if i > 32
        if start_img == 0
            start_img = 1;
        end
        if end_img == 0
```

```
            end_img = 32;
        end
    end

end

slices = end_img-start_img+1;

for i=start_img:end_img

    [min_pos_lite,max_pos_lite,n_lite]=scan_slice(image_buf(i,:),handles);

    num_empty = max_pos_lite - min_pos_lite + 1 - n_lite;

    if slices > 6
        slices_third = floor(slices/3);
    else
        slices_third = 1;
    end


    if num_empty > max_hole_diameter & i > slices_third & i < slices -
slices_third
        max_hole_diameter = num_empty;
        edge_at_max_hole = max_pos_lite - min_pos_lite + 1;
    end

    if possibly_hollow == 1 & status == 0
        if n_lite > 0 & n_lite ~= max_pos_lite - min_pos_lite + 1
            if im_width(2) <= 16
                new = bin2dec(image_buf(i,1:im_width(2)));
                olddec = bin2dec(old(1:im_width(2)));

            else
                new = bin2dec(image_buf(i,17:im_width(2))) +
bitshift(bin2dec(image_buf(i,1:16)),16);
                olddec = bin2dec(old(17:im_width(2))) +
bitshift(bin2dec(old(1:16)),16);
            end

            newandold = bitand(new , olddec);
            if newandold == zeros

                status=1;
%                 return
            else
                old = mask_start_end(max_pos_lite, min_pos_lite,
image_buf(i,:),im_width(2));
            end

        elseif n_lite > 0

            %             bufdec = bin2dec(image_buf(i,33:64)) +
bitshift(bin2dec(image_buf(i,1:32)),32);
```

```
            %             olddec = bin2dec(old(33:64)) +
bitshift(bin2dec(old(1:32)),32);


            if im_width(2) <= 16

                bufdec1 = bin2dec(image_buf(i,1:im_width(2)));
                olddec1 = bin2dec(old(1:im_width(2)));

                bufdec2 = 0;
                olddec2 = 0;

            else
                bufdec1 = bin2dec(image_buf(i,1:16));
                olddec1 = bin2dec(old(1:16));

                bufdec2 = bin2dec(image_buf(i,17:im_width(2)));
                olddec2 = bin2dec(old(17:im_width(2)));
            end

            bufandold1 = bitand(bufdec1,olddec1);
            bufandold2 = bitand(bufdec2,olddec2);

            hole_size = length(find(old == '1'));
            cover_size = length(find(dec2bin(bufandold1) == '1')) +
length(find(dec2bin(bufandold2) == '1'));


%
%             bufandold = bitand(bufdec , olddec);
%             hole_size = length(find(old == '1'));
%             cover_size = length(find(dec2bin(bufandold) == '1'));
%
%



            if bufandold1 + bufandold2 > 0
%                 if cover_size <= 2 & hole_size ~=1
if cover_size <= .65*hole_size
                    status = 1;
%                     return;
                end
                possibly_hollow = 0;
                old = 0;
            elseif i > 1
                status = 1;
%                 return;
            else
                possibly_hollow = 0;
                old = 0;
            end

        else
            possibly_hollow = 0;
```

```
        end
    elseif status == 0
        if n_lite > 0 & n_lite ~= max_pos_lite - min_pos_lite + 1
            old = mask_start_end(max_pos_lite, min_pos_lite,
image_buf(i,:),im_width(2));

            possibly_hollow = 1;
        end
    end
end
return;

function old = mask_start_end(end_mask, start_mask, to_mask,bits_per_slice)

result=0;

if start_mask == 0 & end_mask == 0
    result = to_mask;

else
    %     to_mask_dec = bin2dec(to_mask(33:64)) +
bitshift(bin2dec(to_mask(1:32)),32);
    %     result = bitshift(bitshift(to_mask_dec,start_mask),-start_mask);
    %     result = bitshift(bitshift(result,-(bits_per_slice - end_mask) +
1),(bits_per_slice - end_mask)+1);
    %     result = dec2bin(result,bits_per_slice);

    result(1:bits_per_slice) = '0';
    result(start_mask:end_mask) = to_mask(start_mask:end_mask);

end

old=char(result);
return;
```

MATLAB code – sizeDist.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Derive the area and size distribution for entire-in particles
%  Include the IWC calculation
%  Include the effective radius
%               Created by Will Wu, 09/18/2013
%  * Modified to use the new maximum size and derive both maximum size
distribution
%    and area-equivalent size distribution.  Will Wu, 10/26/2013
%  * Modified to calculate terminal velocity using Heymsfield and Westbrook
(2010) method
%    and precipitation rate.     Will Wu, 01/15/2014
%  * Modified to include mass size distribution with habit info.
%         Will Wu, 02/09/2014
%  * Modified to include particle area using A-D relations. Will Wu, Feb 14,
2014
%  * Special Edition for Boston Cloud workshop, Wei Wu, 04/01/2014
%  * Gneralized as a new sorting function for all probes, Wei Wu, July 25th,
2014
%
%  Usage:
%    infile:   Input filename, string
%    outfile:       Output filename, string
%    tas:           True air speed, double array
%    timehhmmss:    Time in hhmmss format, double array
%    probename:     Should be one of 'HVPS', 'CIP', 'PIP', '2DC', '2DP',
'F2DC'
%    d_choice:      the definition of Dmax, should use 6 usually. [1-6]
%    SAmethod:      0: Center in; 1: Entire in; 2: With Correction
%    Pres:          1 second pressure data
%    Temp:          1 second temperature data
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function sizeDist(infile, outfile, tas, timehhmmss, probename, d_choice,
SAmethod, Pres, Temp)

%% Probe information
switch probename
    case 'HVPS'
        % For the HVPS
        num_diodes =128;
        num_bins = 28;
        diodesize = .150;
        armdst=161.;
        kk=[200.0   400.0   600.0   800.0  1000.0  1200.0  1400.0  1600.0
1800.0  2200.0  2600.0 ...
             3000.0  3400.0  3800.0  4200.0  4600.0  5000.0  6000.0  7000.0
8000.0  9000.0 10000.0 ...
             12000.0 14000.0 16000.0 18000.0 20000.0 25000.0 30000.0]/1000;
        %num_bins =128;
        %kk=diodesize/2:diodesize:(num_bins+0.5)*diodesize;
    case 'CIP'
        % For the CIP
        num_diodes =64;
        diodesize = .025;
```

```
        armdst=100.;
        %num_bins = 64;
        %kk=diodesize/2:diodesize:(num_bins+0.5)*diodesize;
        num_bins=19;
        kk=[50.0   100.0   150.0   200.0   250.0   300.0   350.0   400.0
475.0   550.0   625.0 ...
            700.0   800.0   900.0  1000.0  1200.0  1400.0  1600.0  1800.0
2000.0]/1000;

    case 'PIP'
        % For the PIP
        num_diodes =64;
        diodesize = .1;
        armdst=400.; %70.;
        %num_bins = 64;
        %kk=diodesize/2:diodesize:(num_bins+0.5)*diodesize;
        num_bins=19;
        kk=[50.0   100.0   150.0   200.0   250.0   300.0   350.0   400.0
475.0   550.0   625.0 ...
            700.0   800.0   900.0  1000.0  1200.0  1400.0  1600.0  1800.0
2000.0]*4/1000;


    case '2DC'
        % For the 2DC
        num_diodes =32;
        diodesize = .025; %.025;
        armdst=61.; %61.;
        num_bins = 32;
        kk=diodesize/2:diodesize:(num_bins+0.5)*diodesize;
        %num_bins=19;
        %kk=[50.0   100.0   150.0   200.0   250.0   300.0   350.0   400.0
475.0   550.0   625.0 ...
        %    700.0   800.0   900.0  1000.0  1200.0  1400.0  1600.0  1800.0
2000.0]/1000;

    case '2DP'
        % For the 2DP
        num_diodes =32;
        diodesize = .200; %.025;
        armdst=260.; %75.77; %61.;
        %num_bins = 32;
        %kk=diodesize/2:diodesize:(num_bins+0.5)*diodesize;
        num_bins=19;
        kk=[50.0   100.0   150.0   200.0   250.0   300.0   350.0   400.0
475.0   550.0   625.0 ...
            700.0   800.0   900.0  1000.0  1200.0  1400.0  1600.0  1800.0
2000.0]*8/1000;

    case 'F2DC'
        % For the 2DC
        num_diodes =64;
        diodesize = .025; %.025;
        armdst=61.; %60; %
        %num_bins = 32;
        %kk=diodesize/2:diodesize:(num_bins+0.5)*diodesize;
```

```
        num_bins=19;
        kk=[50.0   100.0   150.0   200.0   250.0   300.0   350.0   400.0
475.0   550.0   625.0 ...
            700.0   800.0   900.0  1000.0  1200.0  1400.0  1600.0  1800.0
2000.0]/1000;


end

res=diodesize*1000;
binwidth=diff(kk);
%SAmethod = 2;
% for i=1:num_bins+1
%     kk(i)=  (diodesize*i)^2*3.1415926/4;
% end

%% Load input and output file
f = netcdf.open(infile,'nowrite');
mainf = netcdf.create(outfile, 'clobber');

tas_char = num2str(timehhmmss);
tas_time =
floor(timehhmmss/10000)*3600+floor(mod(timehhmmss,10000)/100)*60+floor(mod(ti
mehhmmss,100));
averaging_time = 1;

%% Define Variables
particle_dist2 = zeros(length(tas)/averaging_time,num_bins)*NaN;
particle_dist_minR  = zeros(length(tas)/averaging_time,num_bins)*NaN;
particle_dist_AreaR = zeros(length(tas)/averaging_time,num_bins)*NaN;
area_dist2 = zeros(length(tas)/averaging_time,num_bins,10)*NaN;
particle_area = zeros(length(tas)/averaging_time,num_bins)*NaN;
% time_interval1 = zeros(length(tas)/averaging_time, 1);
% cip2_ar = zeros(1,length(tas)/averaging_time)*NaN;
cip2_iwc = zeros(length(tas)/averaging_time,num_bins)*NaN;
cip2_iwcbl = zeros(length(tas)/averaging_time,num_bins)*NaN;
cip2_vt = zeros(length(tas)/averaging_time,num_bins)*NaN;
cip2_pr = zeros(length(tas)/averaging_time,num_bins)*NaN;
cip2_partarea = zeros(length(tas)/averaging_time,num_bins)*NaN;

cip2_re = zeros(1,length(tas)/averaging_time)*NaN;
cip2_habitsd = zeros(length(tas)/averaging_time,num_bins,10)*NaN;
cip2_habitmsd = zeros(length(tas)/averaging_time,num_bins,10)*NaN;
goodpartpercent=zeros(length(tas)/averaging_time,1);
rejectpercentbycriterion=zeros(length(tas)/averaging_time,12);
goodintpercent=zeros(length(tas)/averaging_time,1);
throwoutpercent=zeros(length(tas)/averaging_time,1);
totalint=zeros(length(tas)/averaging_time,1);
intsum=zeros(length(tas)/averaging_time,1);
intparticles=zeros(length(tas)/averaging_time,1);

area_bins = 0:.1:1.01;
one_sec_times = tas_time;
one_sec_dur = length(one_sec_times);
total_one_sec_locs(1:one_sec_dur) = 0;
```

```
start_time = floor(tas_time(1));
end_time = ceil(tas_time(end));
one_sec_tas(1:one_sec_dur) = 0;
one_sec_tas_entire(1:one_sec_dur) = 0;
deadtime(1:one_sec_dur) = 0;
warning off all
one_sec_times=[one_sec_times;one_sec_times(one_sec_dur)+1];
time_interval2 = zeros(one_sec_dur,1);



%% Load particles for each second, and then process them.
%  Only for large files cannot be processed at once
[~, NumofPart] = netcdf.inqDim(f,0); % Check the number of particles

image_time_hhmmssall = netcdf.getVar(f,netcdf.inqVarID(f,'Time')); % 'Time'?
startindex=[true;diff(image_time_hhmmssall)>0];
starttime=image_time_hhmmssall(startindex);
clearvars image_time_hhmmssall
start_all=find(startindex>0)-1;
count_all=[diff(start_all); NumofPart-start_all(end)];

starttime = starttime(count_all>10);
start_all = start_all(count_all>10);
count_all = count_all(count_all>10);

length(starttime)-length(unique(starttime))
[starttime, ia, ic] = unique(starttime);
start_all = start_all(ia);
count_all = count_all(ia);


nNoTAS=0;
for i=1:length(starttime)
    if isempty(timehhmmss(timehhmmss == starttime(i)))
        starttime(i)=500000;
        nNoTAS=nNoTAS+1;
    end
end

nNoTAS
start_all = start_all(starttime<500000);
count_all = count_all(starttime<500000);
starttime = starttime(starttime<500000);


if (int32(timehhmmss(1))>int32(starttime(1)))
    error('Watch Out for less TAS time from begining!')
end

jjj=1;
nThrow11=0;
maxRecNum=1;
for i=1:length(tas)
```

```
%     if i>=10400
%         disp('2wewe');
%     end

    if (int32(timehhmmss(i))==int32(starttime(jjj)))

        start=start_all(jjj);
        count=count_all(jjj);
        jjj=min(jjj+1,length(start_all));

        %% Load autoanalysis parameters
        msec =
netcdf.getVar(f,netcdf.inqVarID(f,'particle_millisec'),start,count);
        microsec =
netcdf.getVar(f,netcdf.inqVarID(f,'particle_microsec'),start,count);
        auto_reject =
netcdf.getVar(f,netcdf.inqVarID(f,'image_auto_reject'),start,count);
        im_width =
netcdf.getVar(f,netcdf.inqVarID(f,'image_width'),start,count);
        im_length =
netcdf.getVar(f,netcdf.inqVarID(f,'image_length'),start,count);
        area = netcdf.getVar(f,netcdf.inqVarID(f,'image_area'),start,count);
        int_arr =
netcdf.getVar(f,netcdf.inqVarID(f,'inter_arrival'),start,count);
        rec_nums =
netcdf.getVar(f,netcdf.inqVarID(f,'parent_rec_num'),start,count);
        top_edges =
netcdf.getVar(f,netcdf.inqVarID(f,'image_max_top_edge_touching'),start,count)
;
        bot_edges =
netcdf.getVar(f,netcdf.inqVarID(f,'image_max_bottom_edge_touching'),start,cou
nt);
        longest_y = netcdf.getVar(f,netcdf.inqVarID(f,'image_longest_y'));
        size_factor = netcdf.getVar(f,netcdf.inqVarID(f,'size_factor'));
        habit1 =
netcdf.getVar(f,netcdf.inqVarID(f,'holroyd_habit'),start,count);
        centerin =
netcdf.getVar(f,netcdf.inqVarID(f,'image_center_in'),start,count);
        entirein =
netcdf.getVar(f,netcdf.inqVarID(f,'image_touching_edge'),start,count);



        % max_dimension = im_width;
        % max_dimension(im_length>im_width)=im_length(im_length>im_width);

        %particle_diameter_minR =
netcdf.getVar(f,netcdf.inqVarID(f,'image_diam_minR'),start,count);
        particle_diameter_AreaR =
netcdf.getVar(f,netcdf.inqVarID(f,'image_diam_AreaR'),start,count);
        %% Change this line to choose which size definition to use

        if 1==d_choice
            particle_diameter_minR = im_length * diodesize; %(im_length+
```

```
        elseif 2==d_choice
            particle_diameter_minR = im_width * diodesize; %(im_length+
        elseif 3==d_choice
            particle_diameter_minR = (im_length + im_width)/2 * diodesize;
%(im_length+
        elseif 4==d_choice
            particle_diameter_minR = sqrt(im_width.^2+im_length.^2) *
diodesize; %(im_length+
        elseif 5==d_choice
            particle_diameter_minR = max(im_width, im_length) * diodesize;
%(im_length+
        elseif 6==d_choice
            particle_diameter_minR =
netcdf.getVar(f,netcdf.inqVarID(f,'image_diam_minR'),start,count); % *
diodesize
        end
        particle_diameter_AreaR = particle_diameter_AreaR * diodesize;

        image_time_hhmmss =
netcdf.getVar(f,netcdf.inqVarID(f,'particle_time'),start,count);
        image_time_hhmmssnew =
netcdf.getVar(f,netcdf.inqVarID(f,'particle_time'),start,count); % Remember
to change back to Time
        fracseccc= netcdf.getVar(f,netcdf.inqVarID(f,'msec'),start,count);


        %% Calculate area of particle according to image reconstruction,
airspeed.
        %disp('Correcting areas for TAS > 125 m s-1 and reconstructing
areas...')

        if(tas(i) > 125)                                % Set to threshold as
necessary - stretch area of particle
           area = area*tas(i)/125;
        end


        particle_mass = area*0;
        calcd_area = area*0;
        for iiii=1:length(area)
           particle_mass(iiii)=single_mass(particle_diameter_minR(iiii)/10,
habit1(iiii));  % in unit of gram
           calcd_area(iiii)=single_area(particle_diameter_minR(iiii)/10,
habit1(iiii));  % in unit of gram
        end
        particle_massbl=0.115/1000*area.^(1.218); % in unit of gram



        %% Added by Robert Jackson -- old version did not have area ratio
code

        area_ratio = area./(pi/4.*particle_diameter_minR.^2);
        auto_reject(area_ratio < .2) = 'z';
```

```
        %% Added by Will to calculate terminal velocity and precipitation
rate
        particle_vt = area*0;
        for iiii=1:length(area)
           particle_vt(iiii)=single_vt(particle_diameter_minR(iiii)/1000,
area_ratio(iiii), particle_mass(iiii)/1000,Pres(i),Temp(i));  % in unit of
gram
        end
        particle_pr=particle_mass.*particle_vt;

        %% Time and interarrival calculation. Modified by Will Wu 11/12/2013
        if strcmp(probename,'2DC')==1 || strcmp(probename,'2DP')==1 ||
strcmp(probename,'F2DC')==1
            image_timeia = floor(image_time_hhmmssnew/10000)*3600 +
mod(floor(image_time_hhmmssnew/100),100)*60 +
mod(image_time_hhmmssnew,100)+fracseccc*1e-2; % for 2DC
        elseif strcmp(probename,'CIP')==1 || strcmp(probename,'PIP')==1
            image_timeia = floor(image_time_hhmmss/10000)*3600 +
mod(floor(image_time_hhmmss/100),100)*60 +
mod(image_time_hhmmss,100)+msec*1e-3+microsec; % for CIP
        else
            image_timeia = floor(image_time_hhmmss/10000)*3600 +
mod(floor(image_time_hhmmss/100),100)*60 +
mod(image_time_hhmmss,100)+msec*1e-3+microsec/10^6; % for HVPS
        end
        image_time = floor(image_time_hhmmssnew/10000)*3600 +
mod(floor(image_time_hhmmssnew/100),100)*60 + mod(image_time_hhmmssnew,100);

        if strcmp(probename,'HVPS')==1
           int_arr=int_arr/10^6;
           int_arr(int_arr<0)=0;
           int_arr(int_arr>0.001)=0;
        end

        %% Verify integrity of interarrival times.
        %disp('Checking Interarrival Times')

        nThrow=0;
        for(itemp=min(rec_nums):max(rec_nums))
           rec_particles = find(rec_nums == itemp);
           rej = auto_reject(rec_particles);
           arr = int_arr(rec_particles);
           sum_arr = sum(arr(2:end));
           if(~isempty(rec_particles) && length(rec_particles) > 1)
               int_arr(rec_particles(1)) = int_arr(rec_particles(2));
           elseif(length(rec_particles) == 1)
               int_arr(rec_particles(1)) = 0;
           end

           if (strcmp(probename,'CIP')==1 || strcmp(probename,'PIP')==1 ||
strcmp(probename,'HVPS')==1 ) % 2DC use the interarrival time for every
particles, not absolute time

               if(isempty(rec_particles))
                 sum_int_arr_good = 0;
               else
```

```
                sum_int_arr_good = image_timeia(rec_particles(end))-
image_timeia(rec_particles(1));
            end
            if ~(sum_int_arr_good >= .6*sum_arr && sum_int_arr_good <=
1.4*sum_arr)
                auto_reject(rec_particles) = 'I';
                %disp(['Record ' num2str(itemp) ' thrown out: Accepted time
= ' num2str(sum_int_arr_good) ' total time = ' num2str(sum_arr)]);
                nThrow=nThrow+1;
                nThrow11=nThrow11+1;
            end

        end
    end
    %disp([num2str(100*nThrow/(max(rec_nums)-min(rec_nums)+1)),'% is
thrown out']);
    throwoutpercent(i)=100*nThrow/(max(rec_nums)-min(rec_nums)+1);
    maxRecNum=max(max(rec_nums),maxRecNum);
    %totalint(i)=sum_int_arr_good;
    %intsum(i)=sum_arr;
    %save('intarrhvps.mat','int_arr')

    %% Apply rejection criteria. Modify the next line to include/exclude
any particles you see fit. This line
    %% currently accepts any non-rejected or hollow particle with
interarrival time > 1e-4 s.
    %disp('Begin to Sort');
    if strcmp(probename,'CIP')==1 || strcmp(probename,'F2DC')==1
        good_particles = (auto_reject == '0' | auto_reject == 'H' |
auto_reject == 'h' | auto_reject == 'u'); %  &  centerin==1; % & int_arr >
1e-5 int_arr > 1e-5 &
    else
        good_particles = (auto_reject == '0' | auto_reject == 'H' |
auto_reject == 'h' | auto_reject == 'u'); % & centerin==1; % & int_arr > 1e-
5;
    end
    if SAmethod==0
        good_particles = good_particles & centerin==1;
    elseif SAmethod==1
        good_particles = good_particles & entirein==0;
    end
    goodpartpercent(i)=sum(good_particles)/length(good_particles);


rejectpercentbycriterion(i,1)=sum(centerin==1)/length(good_particles);
    rejectpercentbycriterion(i,2)=sum(auto_reject ==
'0')/length(good_particles);
    rejectpercentbycriterion(i,3)=sum(auto_reject ==
'H')/length(good_particles);
    rejectpercentbycriterion(i,4)=sum(auto_reject ==
'h')/length(good_particles);
    rejectpercentbycriterion(i,5)=sum(auto_reject ==
'u')/length(good_particles);
    rejectpercentbycriterion(i,6)=sum(auto_reject ==
'a')/length(good_particles);
```

```
        rejectpercentbycriterion(i,7)=sum(auto_reject ==
't')/length(good_particles);
        rejectpercentbycriterion(i,8)=sum(auto_reject ==
'p')/length(good_particles);
        rejectpercentbycriterion(i,9)=sum(auto_reject ==
's')/length(good_particles);
        rejectpercentbycriterion(i,10)=sum(auto_reject ==
'z')/length(good_particles);
        rejectpercentbycriterion(i,11)=sum(auto_reject ==
'i')/length(good_particles);
        rejectpercentbycriterion(i,12)=sum(auto_reject ==
'A')/length(good_particles);
        intparticles(i)=length(good_particles);

disp([sum(good_particles),length(good_particles),length(good_particles)-
sum(good_particles)]);

        good_image_times = image_time(good_particles);
        good_particle_diameter_minR = particle_diameter_minR(good_particles);
        good_particle_diameter_AreaR =
particle_diameter_AreaR(good_particles);
        good_int_arr=int_arr(good_particles);
        good_ar = area_ratio(good_particles);
        good_area = area(good_particles);
        good_iwc=particle_mass(good_particles);
        good_partarea=calcd_area(good_particles);
        good_iwcbl=particle_massbl(good_particles);
        good_vt=particle_vt(good_particles);
        good_pr=particle_pr(good_particles);
        habit=habit1(good_particles);

        good_particle_diameter=good_particle_diameter_minR;

        %% Sort particles into size distributions

        %for i = 1 : one_sec_dur

           if mod(i,1000) == 0
               disp([[num2str(i),'/',num2str(one_sec_dur)],' |
',datestr(now)])
           end

           good_one_sec_locs = find(good_image_times >= one_sec_times(i) &
good_image_times < one_sec_times(i+1));
           total_one_sec_locs(i) = length(find(image_time >= one_sec_times(i)
& image_time < one_sec_times(i+1)));
           time_interval2(i) = sum(int_arr(image_time >= one_sec_times(i) &
image_time < one_sec_times(i+1)));
           if sum(image_time >= one_sec_times(i) & image_time <
one_sec_times(i+1)) ~= length(image_time)
               disp([num2str(sum(image_time >= one_sec_times(i) & image_time
< one_sec_times(i+1))) ,' / ' , num2str(length(image_time)), '  Error on
sizing']);
           end
           goodintpercent(i) = sum(good_int_arr(good_image_times >=
one_sec_times(i) & good_image_times < one_sec_times(i+1)))/time_interval2(i);
```

```
            if(total_one_sec_locs(i) == 0)
              time_interval2(i) = 1;
            end
            one_sec_ar(i) = mean(good_ar(good_one_sec_locs));

            if ~isempty(good_one_sec_locs)

                for j = 1:num_bins
                    particle_dist_minR(i,j)  =
length(find(good_particle_diameter_minR(good_one_sec_locs) >= kk(j) &...
                        good_particle_diameter_minR(good_one_sec_locs) <
kk(j+1)));
                    particle_dist_AreaR(i,j) =
length(find(good_particle_diameter_AreaR(good_one_sec_locs) >= kk(j) &...
                        good_particle_diameter_AreaR(good_one_sec_locs) <
kk(j+1)));

                    % Create Habit Number Size Distribution
                    cip2_habitsd(i,j,1) =
length(find(habit(good_one_sec_locs)=='s' &
good_particle_diameter(good_one_sec_locs) >= kk(j) &...
                        good_particle_diameter(good_one_sec_locs) < kk(j+1)));
                    cip2_habitsd(i,j,2) =
length(find(habit(good_one_sec_locs)=='l' &
good_particle_diameter(good_one_sec_locs) >= kk(j) &...
                        good_particle_diameter(good_one_sec_locs) < kk(j+1)));
                    cip2_habitsd(i,j,3) =
length(find(habit(good_one_sec_locs)=='o' &
good_particle_diameter(good_one_sec_locs) >= kk(j) &...
                        good_particle_diameter(good_one_sec_locs) < kk(j+1)));
                    cip2_habitsd(i,j,4) =
length(find(habit(good_one_sec_locs)=='t' &
good_particle_diameter(good_one_sec_locs) >= kk(j) &...
                        good_particle_diameter(good_one_sec_locs) < kk(j+1)));
                    cip2_habitsd(i,j,5) =
length(find(habit(good_one_sec_locs)=='h' &
good_particle_diameter(good_one_sec_locs) >= kk(j) &...
                        good_particle_diameter(good_one_sec_locs) < kk(j+1)));
                    cip2_habitsd(i,j,6) =
length(find(habit(good_one_sec_locs)=='i' &
good_particle_diameter(good_one_sec_locs) >= kk(j) &...
                        good_particle_diameter(good_one_sec_locs) < kk(j+1)));
                    cip2_habitsd(i,j,7) =
length(find(habit(good_one_sec_locs)=='g' &
good_particle_diameter(good_one_sec_locs) >= kk(j) &...
                        good_particle_diameter(good_one_sec_locs) < kk(j+1)));
                    cip2_habitsd(i,j,8) =
length(find(habit(good_one_sec_locs)=='d' &
good_particle_diameter(good_one_sec_locs) >= kk(j) &...
                        good_particle_diameter(good_one_sec_locs) < kk(j+1)));
                    cip2_habitsd(i,j,9) =
length(find(habit(good_one_sec_locs)=='a' &
good_particle_diameter(good_one_sec_locs) >= kk(j) &...
                        good_particle_diameter(good_one_sec_locs) < kk(j+1)));
```

```
                    cip2_habitsd(i,j,10) =
length(find(habit(good_one_sec_locs)=='I' &
good_particle_diameter(good_one_sec_locs) >= kk(j) &...
                        good_particle_diameter(good_one_sec_locs) < kk(j+1)));

                    % Create Habit Mass Size Distribution
                    cip2_habitmsd(i,j,1) =
sum(good_iwc(habit(good_one_sec_locs)=='s' &
good_particle_diameter(good_one_sec_locs) >= kk(j) &...
                        good_particle_diameter(good_one_sec_locs) < kk(j+1)));
                    cip2_habitmsd(i,j,2) =
sum(good_iwc(habit(good_one_sec_locs)=='l' &
good_particle_diameter(good_one_sec_locs) >= kk(j) &...
                        good_particle_diameter(good_one_sec_locs) < kk(j+1)));
                    cip2_habitmsd(i,j,3) =
sum(good_iwc(habit(good_one_sec_locs)=='o' &
good_particle_diameter(good_one_sec_locs) >= kk(j) &...
                        good_particle_diameter(good_one_sec_locs) < kk(j+1)));
                    cip2_habitmsd(i,j,4) =
sum(good_iwc(habit(good_one_sec_locs)=='t' &
good_particle_diameter(good_one_sec_locs) >= kk(j) &...
                        good_particle_diameter(good_one_sec_locs) < kk(j+1)));
                    cip2_habitmsd(i,j,5) =
sum(good_iwc(habit(good_one_sec_locs)=='h' &
good_particle_diameter(good_one_sec_locs) >= kk(j) &...
                        good_particle_diameter(good_one_sec_locs) < kk(j+1)));
                    cip2_habitmsd(i,j,6) =
sum(good_iwc(habit(good_one_sec_locs)=='i' &
good_particle_diameter(good_one_sec_locs) >= kk(j) &...
                        good_particle_diameter(good_one_sec_locs) < kk(j+1)));
                    cip2_habitmsd(i,j,7) =
sum(good_iwc(habit(good_one_sec_locs)=='g' &
good_particle_diameter(good_one_sec_locs) >= kk(j) &...
                        good_particle_diameter(good_one_sec_locs) < kk(j+1)));
                    cip2_habitmsd(i,j,8) =
sum(good_iwc(habit(good_one_sec_locs)=='d' &
good_particle_diameter(good_one_sec_locs) >= kk(j) &...
                        good_particle_diameter(good_one_sec_locs) < kk(j+1)));
                    cip2_habitmsd(i,j,9) =
sum(good_iwc(habit(good_one_sec_locs)=='a' &
good_particle_diameter(good_one_sec_locs) >= kk(j) &...
                        good_particle_diameter(good_one_sec_locs) < kk(j+1)));
                    cip2_habitmsd(i,j,10) =
sum(good_iwc(habit(good_one_sec_locs)=='I' &
good_particle_diameter(good_one_sec_locs) >= kk(j) &...
                        good_particle_diameter(good_one_sec_locs) < kk(j+1)));


                    particle_area(i,j) =
sum(good_area(good_one_sec_locs(good_particle_diameter(good_one_sec_locs) >=
kk(j) &...
                        good_particle_diameter(good_one_sec_locs) <
kk(j+1))));
```

```
                    cip2_iwc(i,j) =
sum(good_iwc(good_one_sec_locs(good_particle_diameter(good_one_sec_locs) >=
kk(j) &...
                        good_particle_diameter(good_one_sec_locs) <
kk(j+1))));

                    cip2_partarea(i,j) =
sum(good_partarea(good_one_sec_locs(good_particle_diameter(good_one_sec_locs)
>= kk(j) &...
                        good_particle_diameter(good_one_sec_locs) <
kk(j+1))));

                    cip2_iwcbl(i,j) =
sum(good_iwcbl(good_one_sec_locs(good_particle_diameter(good_one_sec_locs) >=
kk(j) &...
                        good_particle_diameter(good_one_sec_locs) <
kk(j+1))));

                    cip2_vt(i,j) =
sum(good_vt(good_one_sec_locs(good_particle_diameter(good_one_sec_locs) >=
kk(j) &...
                        good_particle_diameter(good_one_sec_locs) <
kk(j+1))));

                    cip2_pr(i,j) =
sum(good_pr(good_one_sec_locs(good_particle_diameter(good_one_sec_locs) >=
kk(j) &...
                        good_particle_diameter(good_one_sec_locs) <
kk(j+1))));


                    for k = 1:length(area_bins)-1
                        area_dist2(i,j,k) =
length(find(good_ar(good_one_sec_locs) >= area_bins(k) & ...
                            good_ar(good_one_sec_locs) < area_bins(k+1) &
good_particle_diameter(good_one_sec_locs) >= kk(j) &...
                            good_particle_diameter(good_one_sec_locs) <
kk(j+1)));
                    end
                end
                particle_dist_minR(i,:)=particle_dist_minR(i,:)./binwidth*10;

particle_dist_AreaR(i,:)=particle_dist_AreaR(i,:)./binwidth*10;
                cip2_iwc(i,:)=cip2_iwc(i,:)./binwidth*10;
                cip2_iwcbl(i,:)=cip2_iwcbl(i,:)./binwidth*10;
                cip2_vt(i,:)=cip2_vt(i,:)./binwidth*10;
                cip2_pr(i,:)=cip2_pr(i,:)./binwidth*10;
                cip2_partarea(i,:)=cip2_partarea(i,:)./binwidth*10;

                particle_area(i,:)=particle_area(i,:)./binwidth*10;
                for mmmmmm=1:10

cip2_habitsd(i,:,mmmmmm)=cip2_habitsd(i,:,mmmmmm)./binwidth*10;

cip2_habitmsd(i,:,mmmmmm)=cip2_habitmsd(i,:,mmmmmm)./binwidth*10;
```

```
                end
                for mmmmmm = 1:length(area_bins)-1
                    area_dist2(i,:,mmmmmm)
=area_dist2(i,:,mmmmmm)./binwidth*10 ;
                end
                cip2_re(i) =
sqrt(3)/3/0.91*1000*sum(cip2_iwc(i,:)./binwidth,2)/sum(particle_area(i,:)./bi
nwidth,2)*1000; % in unit of um
            else
                particle_dist_minR(i,1:num_bins) = 0;
                particle_dist_AreaR(i,1:num_bins) = 0;
                area_dist2(i,1:num_bins,1:length(area_bins)-1) = 0;
                cip2_partarea(i,:) = 0;
                cip2_iwc(i,:) = 0;
                cip2_iwcbl(i,:) = 0;
                cip2_vt(i,:) = 0;
                cip2_pr(i,:) = 0;
                cip2_re(i) = 0;
                cip2_habitsd(i,:,:) = 0;
                cip2_habitmsd(i,:,:) = 0;
                time_interval2(i) = 1;
            end
         %end

         warning on all
    else
       particle_dist_minR(i,1:num_bins) = 0; %NaN;
       particle_dist_AreaR(i,1:num_bins) = 0;
       area_dist2(i,1:num_bins,1:length(area_bins)-1) = NaN; %int16(0);
       cip2_partarea(i,:) = NaN;
       cip2_iwc(i,:) = NaN;
       cip2_iwcbl(i,:) = NaN;
       cip2_vt(i,:) = NaN;
       cip2_pr(i,:) = NaN;
       cip2_re(i) = NaN;
       cip2_habitsd(i,:,:) = NaN;
       cip2_habitmsd(i,:,:) = NaN;
       one_sec_ar(i) = NaN;
       goodpartpercent(i)=1;
       rejectpercentbycriterion(i,:)=NaN;
       intparticles(i)=NaN;
       time_interval2(i) = 1;
    end

end

% Finished Sorting and close input file.
netcdf.close(f);

%% Check TAS length, should be the same
if (jjj~=length(start_all))
    error('Watch Out for less TAS time at the end!')
end

disp([num2str(100*nThrow11/maxRecNum),'% is thrown out IN TOTAL']);
```

```
%% Combine - calculate sample volumes, and divide by sample volumes
% Modified by Will, Nov 27th, 2013. For flexible bins
cip2_binmin = kk(1:end-1); %diodesize/2:diodesize:(num_bins-0.5)*diodesize;
%(12.5:25:(num_bins-0.5)*25);
cip2_binmax = kk(2:end); %3*diodesize/2:diodesize:(num_bins+0.5)*diodesize;
%(37.5:25:(num_bins+0.5)*25);
cip2_binmid = (cip2_binmin+cip2_binmax)/2; %
diodesize:diodesize:num_bins*diodesize; %(25:25:num_bins*25);
cip2_bindD = diff(kk); %diodesize*ones(1,num_bins);

% cip2_binmin = diodesize/2:diodesize:(num_bins-0.5)*diodesize;
%(12.5:25:(num_bins-0.5)*25);
% cip2_binmax = 3*diodesize/2:diodesize:(num_bins+0.5)*diodesize;
%(37.5:25:(num_bins+0.5)*25);
% cip2_binmid = diodesize:diodesize:num_bins*diodesize; %(25:25:num_bins*25);
% cip2_bindD = diodesize*ones(1,num_bins);

%sa2 = calc_sa(num_bins,res,armdst,num_bins);  %mm2
sa2 = calc_sa_randombins(cip2_binmid,res,armdst,num_diodes, SAmethod);
%(bins_mid,res,armdst,num_diodes)
vol_scale_factor = tas/125;  %% Clocking problem correction
vol_scale_factor(vol_scale_factor < 1) = 1;
for j=1:num_bins
    % c  Sample volume is in m-3
    %svol_old(j,:)=dof/100.*sa/100.*tas;
    svol2(j,:) = sa2(j)*(1e-3)^2*time_interval2.*tas.*vol_scale_factor; %m3
end

svol2 = svol2*100^3; %cm3

cip2_conc_minR  = particle_dist_minR./svol2';
cip2_conc_AreaR = particle_dist_AreaR./svol2';
for j = 1:10
    svol2a(:,:,j) = svol2';
end
cip2_conc_areaDist = permute(double(area_dist2)./svol2a, [3 2 1]);

cip2_n = nansum(cip2_conc_minR,2);
cip2_lwc = lwc_calc(cip2_conc_minR,cip2_binmid);
cip2_area = particle_area./svol2';
cip2_partarea = cip2_partarea./svol2';
cip2_iwc = cip2_iwc./svol2';
cip2_iwcbl = cip2_iwcbl./svol2';
cip2_vt = cip2_vt./svol2';
cip2_pr = cip2_pr./svol2';

%% Output results into NETCDF files (mainf)
save allareadistvforfuture.mat % Temporarily output for debug
% Define Dimensions
dimid0 = netcdf.defDim(mainf,'CIPcorrlen',num_bins);
dimid1 = netcdf.defDim(mainf,'CIParealen',10);
dimid2 = netcdf.defDim(mainf,'Time',length(timehhmmss));
dimid3 = netcdf.defDim(mainf,'Habit',10);
```

```
% Define Variables
varid0 = netcdf.defVar(mainf,'time','double',dimid2);
netcdf.putAtt(mainf, varid0,'units','HHMMSS');
netcdf.putAtt(mainf, varid0,'name','Time');
varid2 = netcdf.defVar(mainf,'bin_min','double',dimid0);
netcdf.putAtt(mainf, varid2,'units','millimeter');
netcdf.putAtt(mainf, varid2,'long_name','bin minimum size');
netcdf.putAtt(mainf, varid2,'short_name','bin min');
varid4 = netcdf.defVar(mainf,'bin_mid','double',dimid0);
netcdf.putAtt(mainf, varid4,'units','millimeter');
netcdf.putAtt(mainf, varid4,'long_name','bin midpoint size');
netcdf.putAtt(mainf, varid4,'short_name','bin mid');
varid3 = netcdf.defVar(mainf,'bin_max','double',dimid0);
netcdf.putAtt(mainf, varid3,'units','millimeter');
netcdf.putAtt(mainf, varid3,'long_name','bin maximum size');
netcdf.putAtt(mainf, varid3,'short_name','bin max');
varid5 = netcdf.defVar(mainf,'bin_dD','double',dimid0);
netcdf.putAtt(mainf, varid5,'units','millimeter');
netcdf.putAtt(mainf, varid5,'long_name','bin size');
netcdf.putAtt(mainf, varid5,'short_name','bin size');
varid1 = netcdf.defVar(mainf,'conc_minR','double',[dimid0 dimid2]);
netcdf.putAtt(mainf, varid1,'units','cm-4');
netcdf.putAtt(mainf, varid1,'long_name','Size distribution using Dmax');
netcdf.putAtt(mainf, varid1,'short_name','N(Dmax)');
varid7 = netcdf.defVar(mainf,'conc_AreaR','double',[dimid0 dimid2]);
netcdf.putAtt(mainf, varid7,'units','cm-4');
netcdf.putAtt(mainf, varid7,'long_name','Size distribution using area-
equivalent Diameter');
netcdf.putAtt(mainf, varid7,'short_name','N(Darea)');
varid6 = netcdf.defVar(mainf,'area','double',[dimid1 dimid0 dimid2]);
netcdf.putAtt(mainf, varid6,'units','cm-4');
netcdf.putAtt(mainf, varid6,'long_name','binned area ratio');
netcdf.putAtt(mainf, varid6,'short_name','binned area ratio');
varid8 = netcdf.defVar(mainf,'n','double',dimid2);
netcdf.putAtt(mainf, varid8,'units','cm-3');
netcdf.putAtt(mainf, varid8,'long_name','number concentration');
netcdf.putAtt(mainf, varid8,'short_name','N');
varid9 = netcdf.defVar(mainf,'total_area','double',[dimid0 dimid2]);
netcdf.putAtt(mainf, varid9,'units','mm2/cm4');
netcdf.putAtt(mainf, varid9,'long_name','projected area (extinction)');
netcdf.putAtt(mainf, varid9,'short_name','Ac');
varid10 = netcdf.defVar(mainf,'mass','double',[dimid0 dimid2]);
netcdf.putAtt(mainf, varid10,'units','g/cm4');
netcdf.putAtt(mainf, varid10,'long_name','mass using m-D relations');
netcdf.putAtt(mainf, varid10,'short_name','mass');
varid11 = netcdf.defVar(mainf,'habitsd','double',[dimid3 dimid0 dimid2]);
netcdf.putAtt(mainf, varid11,'units','cm-4');
netcdf.putAtt(mainf, varid11,'long_name','Size Distribution with Habit');
netcdf.putAtt(mainf, varid11,'short_name','habit SD');
varid12 = netcdf.defVar(mainf,'re','double',dimid2);
netcdf.putAtt(mainf, varid12,'units','mm');
netcdf.putAtt(mainf, varid12,'long_name','effective radius');
netcdf.putAtt(mainf, varid12,'short_name','Re');
varid13 = netcdf.defVar(mainf,'ar','double',dimid2);
netcdf.putAtt(mainf, varid13,'units','100/100');
netcdf.putAtt(mainf, varid13,'long_name','Area Ratio');
netcdf.putAtt(mainf, varid13,'short_name','AR');
```

```
varid14 = netcdf.defVar(mainf,'massBL','double',[dimid0 dimid2]);
netcdf.putAtt(mainf, varid14,'units','g/cm4');
netcdf.putAtt(mainf, varid14,'long_name','mass using Baker and Lawson
method');
netcdf.putAtt(mainf, varid14,'short_name','mass_BL');
varid19 = netcdf.defVar(mainf,'Calcd_area','double',[dimid0 dimid2]);
netcdf.putAtt(mainf, varid19,'units','cm^2/cm4');
netcdf.putAtt(mainf, varid19,'long_name','Particle Area Calculated using A-D
realtions');
netcdf.putAtt(mainf, varid19,'short_name','Ac_calc');
varid15 = netcdf.defVar(mainf,'Reject_ratio','double',dimid2);
netcdf.putAtt(mainf, varid15,'units','100/100');
netcdf.putAtt(mainf, varid15,'long_name','Reject Ratio');
varid16 = netcdf.defVar(mainf,'vt','double',[dimid0 dimid2]);
netcdf.putAtt(mainf, varid16,'units','g/cm4');
netcdf.putAtt(mainf, varid16,'long_name','Mass-weighted terminal velocity');
varid17 = netcdf.defVar(mainf,'Prec_rate','double',[dimid0 dimid2]);
netcdf.putAtt(mainf, varid17,'units','mm/hr');
netcdf.putAtt(mainf, varid17,'long_name','Precipitation Rate');
varid18 = netcdf.defVar(mainf,'habitmsd','double',[dimid3 dimid0 dimid2]);
netcdf.putAtt(mainf, varid18,'units','g/cm-4');
netcdf.putAtt(mainf, varid18,'long_name','Mass Size Distribution with
Habit');
netcdf.putAtt(mainf, varid18,'short_name','Habit Mass SD');
netcdf.endDef(mainf)

% Output Variables
netcdf.putVar ( mainf, varid0, timehhmmss );
netcdf.putVar ( mainf, varid2, cip2_binmin );
netcdf.putVar ( mainf, varid3, cip2_binmax );
netcdf.putVar ( mainf, varid4, cip2_binmid );
netcdf.putVar ( mainf, varid5, cip2_bindD );
netcdf.putVar ( mainf, varid1, cip2_conc_minR' );
netcdf.putVar ( mainf, varid7, cip2_conc_AreaR' );
netcdf.putVar ( mainf, varid6, cip2_conc_areaDist);
netcdf.putVar ( mainf, varid8, cip2_n);
netcdf.putVar ( mainf, varid9, cip2_area');
netcdf.putVar ( mainf, varid10, cip2_iwc');
netcdf.putVar ( mainf, varid11, permute(double(cip2_habitsd)./svol2a, [3 2
1]) );
netcdf.putVar ( mainf, varid12, cip2_re );
netcdf.putVar ( mainf, varid13, one_sec_ar );
netcdf.putVar ( mainf, varid14, cip2_iwcbl' );
netcdf.putVar ( mainf, varid15, 1-goodpartpercent );
netcdf.putVar ( mainf, varid16, cip2_vt' );
netcdf.putVar ( mainf, varid17, cip2_pr' );
netcdf.putVar ( mainf, varid18, permute(double(cip2_habitmsd)./svol2a, [3 2
1]) );
netcdf.putVar ( mainf, varid19, cip2_partarea');

netcdf.close(mainf) % Close output NETCDF file


end
```

MATLAB code – dropsize.m

```
function
[center_in,axis_ratio,diameter_circle_fit,diameter_horiz_chord,diameter_vert_
chord,diameter_horiz_mean, diameter_spheroid]=...

dropsize(max_horizontal_length,max_vertical_length,image_area,largest_edge_to
uching,...

smallest_edge_touching,diode_size,corrected_horizontal_diode_size,number_diod
es_in_array)

global max_vertical_chords max_horizontal_chords vertical_chords
vertical_chord_equivalent_spherical_diameters...
    horizontal_chords horizontal_chord_equivalent_spherical_diameters
%
%
%   DROP SIZING FOR 2DP PROBES (equiv sph vol diam)
%       one diode added to max_horizontal_length_in_pixels for missing first
slice
%       uncalculated size appears as zero; oversize as 9.1
%
%
%   diameter_horz_chord
%       d0 from horz chord (max_horizontal_length_in_pixels + 1)
%       designed for sideways-looking probe
%       but can be used for any probe orientation
%       with center-in image of equil shape
%
%
%   diameter_circle_fit
%       Heimsfield-Parish CIRCLE FIT SIZES FOR 2-EDGE & 1-EDGE(CENTER OUT)
IMAGES:
%       d0 from horz chord (circle fit)
%           designed for downward-looking probe
%
%
%   diameter_vert_chord
%       max_vertical_length_in_pixels & 2-CHORD SIZES FOR ENTIRE-IN IMAGES:
%       d0 from vert chord (max_vertical_length_in_pixels)
%           designed for sideways-looking probe
%           optional size for entire-in images
%
%
%   [If diam_vchord differs from diam_hchord, then drop is not equil shape]
%
%       diameter_horz_mean
%           d0 from mean horz chord [(xmax+1)*ymax]^0.5
%           designed for downward-looking probe
%           optional size for distorted entire-in images
%
%       diameter_spheroid
%           d0 from spheroid assumption (hchord*hchord*vchord)^1/3
%           designed for sideways-looking probe
%           optional size for distorted entire-in images
%
```

```
%
%   AXIS RATIO
%       axis_ratio = max_vertical_length_in_pixels /
max_horizontal_length_in_pixels
%       for entire-in image (no smoothing)
%       center_in = (0 if particle is not center in, 1 if particle is center
%   in)

max_horizontal_chords = 117;
max_vertical_chords = 55;

diodes_added_to_length = 0;
diodes_added_to_height = 1.0;

horizontal_chords = [...
     0.0000,  0.1000,  0.2000,  0.3000,  0.4000,  0.5000,  0.6000,  0.7000,
0.8000,  0.9000,...
     1.0000,  1.1000,  1.2000,  1.3000,  1.4000,  1.5000,  1.6000,  1.7000,
1.8000,  1.9000,...
     2.0000,  2.1000,  2.2000,  2.3000,  2.4000,  2.5000,  2.6000,  2.7000,
2.8000,  2.9000,...
     3.0000,  3.1000,  3.2000,  3.3000,  3.4000,  3.5000,  3.6000,  3.7000,
3.8000,  3.9000,...
     4.0000,  4.1000,  4.2000,  4.3000,  4.4000,  4.5000,  4.6000,  4.7000,
4.8000,  4.9000,...
     5.0000,  5.1000,  5.2000,  5.3000,  5.4000,  5.5000,  5.6000,  5.7000,
5.8000,  5.9000,...
     6.0000,  6.1000,  6.2000,  6.3000,  6.4000,  6.5000,  6.6000,  6.7000,
6.8000,  6.9000,...
     7.0000,  7.1000,  7.2000,  7.3000,  7.4000,  7.5000,  7.6000,  7.7000,
7.8000,  7.9000,...
     8.0000,  8.1000,  8.2000,  8.3000,  8.4000,  8.5000,  8.6000,  8.7000,
8.8000,  8.9000,...
     9.0000,  9.1000,  9.2000,  9.3000,  9.4000,  9.5000,  9.6000,  9.7000,
9.8000,  9.9000,...
    10.0000, 10.1000, 10.2000, 10.3000, 10.4000, 10.5000, 10.6000, 10.7000,
10.8000, 10.9000,...
    11.0000, 11.1000, 11.2000, 11.3000, 11.4000, 11.5000, 11.6000];

horizontal_chord_equivalent_spherical_diameters = [...
    0.0000, 0.1000, 0.2000, 0.3000, 0.3998, 0.4996, 0.5992, 0.6986, 0.7976,
0.8964,...
    0.9947, 1.0927, 1.1903, 1.2875, 1.3842, 1.4804, 1.5761, 1.6711, 1.7657,
1.8597,...
    1.9531, 2.0460, 2.1385, 2.2304, 2.3217, 2.4124, 2.5026, 2.5921, 2.6812,
2.7696,...
    2.8574, 2.9448, 3.0319, 3.1185, 3.2046, 3.2903, 3.3755, 3.4603, 3.5446,
3.6286,...
    3.7120, 3.7950, 3.8776, 3.9598, 4.0416, 4.1229, 4.2036, 4.2841, 4.3642,
4.4440,...
    4.5233, 4.6024, 4.6811, 4.7597, 4.8378, 4.9152, 4.9924, 5.0690, 5.1452,
5.2210,...
    5.2961, 5.3711, 5.4457, 5.5201, 5.5942, 5.6681, 5.7416, 5.8148, 5.8877,
5.9602,...
    6.0323, 6.1041, 6.1756, 6.2467, 6.3175, 6.3879, 6.4580, 6.5278, 6.5973,
6.6664,...
```

```
    6.7353, 6.8040, 6.8720, 6.9399, 7.0075, 7.0744, 7.1411, 7.2074, 7.2733,
7.3388,...
    7.4040, 7.4688, 7.5332, 7.5973, 7.6610, 7.7241, 7.7864, 7.8490, 7.9117,
7.9739,...
    8.0359, 8.0976, 8.1591, 8.2203, 8.2813, 8.3421, 8.4027, 8.4631, 8.5233,
8.5834,...
    8.6433, 8.7030, 8.7625, 8.8220, 8.8814, 8.9407, 8.9998];

vertical_chords = [...
    0.0000,  0.1000,  0.2000,  0.3000,  0.4000,  0.5000,  0.6000,  0.7000,
0.8000,  0.9000,...
    1.0000,  1.1000,  1.2000,  1.3000,  1.4000,  1.5000,  1.6000,  1.7000,
1.8000,  1.9000,...
    2.0000,  2.1000,  2.2000,  2.3000,  2.4000,  2.5000,  2.6000,  2.7000,
2.8000,  2.9000,...
    3.0000,  3.1000,  3.2000,  3.3000,  3.4000,  3.5000,  3.6000,  3.7000,
3.8000,  3.9000,...
    4.0000,  4.1000,  4.2000,  4.3000,  4.4000,  4.5000,  4.6000,  4.7000,
4.8000,  4.9000,...
    5.0000,  5.1000,  5.2000,  5.3000,  5.4000];
vertical_chord_equivalent_spherical_diameters = [...
    0.0000,  0.1000,  0.2000,  0.3001,  0.4003,  0.5008,  0.6016,  0.7028,
0.8048,  0.9075,...
    1.0110,  1.1155,  1.2208,  1.3271,  1.4348,  1.5438,  1.6545,  1.7668,
1.8809,  1.9967,...
    2.1142,  2.2336,  2.3553,  2.4792,  2.6059,  2.7353,  2.8677,  3.0023,
3.1394,  3.2792,...
    3.4219,  3.5678,  3.7169,  3.8695,  4.0259,  4.1865,  4.3512,  4.5201,
4.6922,  4.8687,...
    5.0525,  5.2438,  5.4432,  5.6474,  5.8580,  6.0775,  6.3073,  6.5485,
6.8018,  7.0729,...
    7.3699,  7.7007,  8.0781,  8.4868,  8.9230];


diameter_circle_fit = 0.0;
diameter_horiz_chord = 0.0;
diameter_vert_chord = 0.0;
diameter_horiz_mean = 0.0;
diameter_spheroid = 0.0;
axis_ratio = 0.0;
center_in = 0;

scaling_factor_for_horizontal_lengths = corrected_horizontal_diode_size /
diode_size;


corrected_diodes_added_to_length = 0;

if(image_area < 1)
    diameter_circle_fit = corrected_diodes_added_to_length;
    diameter_horiz_chord = corrected_diodes_added_to_length;
    diameter_vert_chord = corrected_diodes_added_to_length;
    diameter_horiz_mean = corrected_diodes_added_to_length;
    diameter_spheroid = corrected_diodes_added_to_length;
    center_in=0;
```

```
    axis_ratio = 1.0;
    return;
end



largest_edge_touching_length =  largest_edge_touching *
scaling_factor_for_horizontal_lengths;
smallest_edge_touching_length = smallest_edge_touching *
scaling_factor_for_horizontal_lengths;
max_horizontal_length = max_horizontal_length *
scaling_factor_for_horizontal_lengths;
corrected_diodes_added_to_length = diodes_added_to_length *
scaling_factor_for_horizontal_lengths;



% determine no. of edges
number_edges_touching = 0;
if largest_edge_touching > 0
    number_edges_touching = number_edges_touching + 1;
    if smallest_edge_touching > 0
        number_edges_touching = number_edges_touching + 1;
    end
elseif smallest_edge_touching > 0
    number_edges_touching = number_edges_touching + 1;
end



center_in = 1;

if max_horizontal_length <= largest_edge_touching
    center_in = 0;
end

if number_edges_touching == 2 & center_in == 1
    temp = number_diodes_in_array + (largest_edge_touching_length^2 -
smallest_edge_touching_length^2 ) / ( 4 * number_diodes_in_array); % + is
replaced by -, Will 10/17/2013
    horizontal_size = sqrt(temp^2 + smallest_edge_touching_length^2);
    horizontal_chord = horizontal_size * diode_size;
    diameter_circle_fit =
horizontal_chord_to_spherical_dia(horizontal_chord);
    diameter_horiz_chord =
horizontal_chord_to_spherical_dia(horizontal_chord);
elseif number_edges_touching == 2
    largest_edge_touching_length = largest_edge_touching_length +
corrected_diodes_added_to_length;
    temp = number_diodes_in_array + (largest_edge_touching_length^2 -
smallest_edge_touching_length^2 ) / ( 4 * number_diodes_in_array);  % + is
replaced by -, Will 10/17/2013
    horizontal_size = sqrt(temp^2 + smallest_edge_touching_length^2);
    horizontal_chord = horizontal_size * diode_size;
    diameter_circle_fit =
horizontal_chord_to_spherical_dia(horizontal_chord);
```

```
elseif number_edges_touching == 1 & center_in == 1
    horizontal_chord = (max_horizontal_length + diodes_added_to_length) *
corrected_horizontal_diode_size;
    diameter_horiz_chord =
horizontal_chord_to_spherical_dia(horizontal_chord);
    diameter_circle_fit =
horizontal_chord_to_spherical_dia(horizontal_chord);
elseif number_edges_touching == 1
    largest_edge_touching_length = largest_edge_touching_length +
corrected_diodes_added_to_length;
    max_vertical_length = max_vertical_length + diodes_added_to_height * 0.5;
    horizontal_size = (0.25 * largest_edge_touching_length^2 +
max_vertical_length^2)/(max_vertical_length);
    horizontal_chord = horizontal_size * diode_size;
    diameter_circle_fit =
horizontal_chord_to_spherical_dia(horizontal_chord);
else
    horizontal_chord = (max_horizontal_length + diodes_added_to_length) *
corrected_horizontal_diode_size;
    diameter_horiz_chord =
horizontal_chord_to_spherical_dia(horizontal_chord);
    diameter_circle_fit =
horizontal_chord_to_spherical_dia(horizontal_chord);
    vertical_chord = (max_vertical_length + diodes_added_to_length) *
diode_size;
    axis_ratio = vertical_chord / horizontal_chord;
    diameter_vert_chord = vertical_chord_to_spherical_dia(vertical_chord);
    horizontal_mean_chord = sqrt(horizontal_chord * vertical_chord);
    diameter_horiz_mean =
horizontal_chord_to_spherical_dia(horizontal_mean_chord);
    diameter_spheroid = exp(log(horizontal_chord^2 * vertical_chord)/3);
end
end
```

```
function diameter=vertical_chord_to_spherical_dia(vertical_chord)
global max_vertical_chords max_horizontal_chords vertical_chords
vertical_chord_equivalent_spherical_diameters...
    horizontal_chords horizontal_chord_equivalent_spherical_diameters

    delta_vertical_chord = .1;

    i = round(vertical_chord * 10);
    if i+1 < max_vertical_chords & i ~=0
        delta_diameter = vertical_chord_equivalent_spherical_diameters(i+1) -
vertical_chord_equivalent_spherical_diameters(i);
        diameter = vertical_chord_equivalent_spherical_diameters(i) +
(delta_diameter / delta_vertical_chord) * (vertical_chord -
vertical_chords(i));
    elseif i == 0
        diameter = 0;
    else
        diameter = 9.1;
    end
```

```
end


function diameter = horizontal_chord_to_spherical_dia(horizontal_chord)
global max_vertical_chords max_horizontal_chords vertical_chords
vertical_chord_equivalent_spherical_diameters...
    horizontal_chords horizontal_chord_equivalent_spherical_diameters


    delta_horizontal_chord = .1;

    i = round(horizontal_chord * 10);
    if i+1 < max_horizontal_chords & i ~= 0
        delta_diameter = horizontal_chord_equivalent_spherical_diameters(i+1)
- horizontal_chord_equivalent_spherical_diameters(i);
        diameter = horizontal_chord_equivalent_spherical_diameters(i) +
(delta_diameter / delta_horizontal_chord) * (horizontal_chord -
horizontal_chords(i));
    elseif i == 0
        diameter = 0;
    else
        diameter = 9.1;
    end

end
```

MATLAB code – Holroyd.m

```
% holroyd - identified particle habit according to Holroyd  (1987)
% inputs:
%    handles - handles structure outlined in run_img_processing.m
%    image_buffer - n x photodiodes/8 raw image buffer without timestamps
% outputs:
%    holroyd_habit - habit code as listed below
function [holroyd_habit] = holroyd(handles, image_buffer)

%/*************************************************************/%
%/* Return code                                               */
%/*                                                           */
%/*                                                           */
%/* reference: J. Atmos. and Oceanic Tech. Vol 4, Sept. '87    */
%/*            pages 498- 511.                                */
%/*                                                           */
%/* 'M' = not calculated, zero image                          */
%/* 'C' = not calculated, center is out                       */
%/* 't' = tiny                                                */
%/* 'o' = oriented                                            */
%/* 'l' = linear                                              */
%/* 'a' = aggregate                                           */
%/* 'g' = graupel                                             */
%/* 's' = spherel                                             */
%/* 'h' = hexagonal                                           */
%/* 'i' = irregular                                           */
%/* 'd' = dendrite                                            */
%/*                                                           */
%/*************************************************************/


    image_size = size(image_buffer);
    probe_resolution = .025;
    n_slices  = image_size(1);


    if (n_slices == 0)
        holroyd_habit = 'M';
        return;
    else
        if (parabola_fit_center_is_in(image_buffer, n_slices) == 1)
            [x_length, y_length, d_length,
w_width,a_angle,area,r2_correlation, F_fine_detail, S_ratio] =
calc_stat(handles,image_buffer, n_slices);



            if (area == 0 )
                holroyd_habit = 'M';
                return;
            elseif (area < 25)
                holroyd_habit = 't';
                return;
            elseif  (r2_correlation >= .4) | ( (d_length < 64) & ( (x_length
>= 4*y_length) |  (y_length >= 4*x_length)))
```

```
                if ((a_angle> 30.0) & (a_angle < 60.0))
                    holroyd_habit = 'o';
                    return;
                else
                    holroyd_habit = 'l';
                    return;
                end
            elseif ( (d_length * probe_resolution > 6.4 ) | (d_length >
160.0))
                holroyd_habit = 'a';
                return;
            elseif (S_ratio >= .7)
                holroyd_habit = 'g';
                return;
            elseif (d_length >= 64)
                if (F_fine_detail <= 13)
                    holroyd_habit = 'g';
                    return;
                else
                    holroyd_habit = 'a';
                    return;
                end
            elseif (F_fine_detail < 5.5)
                holroyd_habit = 's';
                return;
            elseif (F_fine_detail < 10.0)
                if (d_length >= 32)
                    holroyd_habit = 'g';
                    return;
                else
                    holroyd_habit = 'h';
                    return;
                end
            elseif ((F_fine_detail < 16.0) | (x_length <= 7.0))
                holroyd_habit = 'i';
                return;
            else
                holroyd_habit = 'd';
                return;
            end

        else
            holroyd_habit = 'C';
            return;
        end
    end
end

%/*********************************************************************/



%/*********************************************************************/
function [x_length, y_length, d_length, w_width,a_angle,area,r2_correlation,
F_fine_detail, S_ratio] = calc_stat(handles, image_buffer, n_slices)
```

```
BITS_PER_SLICE = handles.bits_per_slice;
MAX_TWOD_DATA_LENGTH = 6000;



area = 0.0;
n_count = 0;
sum_x2= 0.0;
sum_y2= 0.0;
sum_x = 0.0;
sum_y = 0.0;
sum_xy= 0.0;
cross_x2= 0.0;
cross_y2= 0.0;
cross_xy= 0.0;
p_perimeter_change = 0;
min_x = MAX_TWOD_DATA_LENGTH*3;
min_y = BITS_PER_SLICE;
max_x = 0;
max_y = 0;

spot_on_off = 0;
fully_on_count = 0;
partial_on_count =0;

if (n_slices <= 0)
    return;
end

for i=1:n_slices
    fully_on_temp = 0;
    for j=1:BITS_PER_SLICE
        if ((image_buffer(i,j)) == '0')

            tx = i;
            ty =  j;
            if (tx > max_x)
                max_x = tx;
            end

            if (tx < min_x)
                min_x = tx;
            end
            if (ty > max_y)
                max_y = ty;
            end
            if (ty < min_y)
                min_y = ty;
            end
            sum_x2 = sum_x2 + tx * tx;
            sum_y2 = sum_y2 + ty * ty;
            sum_x  = sum_x  + tx;
            sum_y  = sum_y  + ty;
            sum_xy = sum_xy + tx * ty;
```

```
                n_count = n_count + 1;
                p(n_count).x = tx;
                p(n_count).y = ty;



                fully_on_temp = fully_on_temp + 1;

                if (spot_on_off == 0)
                    spot_on_off = 1;
                    p_perimeter_change = p_perimeter_change + 1;
                end
            else
                if spot_on_off == 1
                    spot_on_off = 0;
                    p_perimeter_change = p_perimeter_change + 1;
                end
            end
        end

        if (fully_on_temp == BITS_PER_SLICE)
            fully_on_count = fully_on_count + 1;
        end
        if (fully_on_temp ~= 0)
            partial_on_count = partial_on_count + 1;
        end
    end
    area = n_count;

%/*** scan the other way for perimeter change ****/

    spot_on_off = 0;
    for j=1:BITS_PER_SLICE
        for i=1:n_slices
            if ((image_buffer(i,j)) == '0')
                if (spot_on_off == 0)
                    spot_on_off = 1;
                    p_perimeter_change = p_perimeter_change + 1;
                end
            else
                if (spot_on_off == 1)
                    spot_on_off = 0;
                    p_perimeter_change = p_perimeter_change + 1;
                end
            end
        end
    end

    if (max_x >= min_x)
        x_length = max_x - min_x +1;
    else
        x_length = 0.0;
    end
```

```
if (max_y >= min_y)
    y_length = max_y - min_y +1;
else
    y_length = 0.0;
end

cross_xy = sum_xy - (sum_x * sum_y / area);
cross_x2 = sum_x2 - (sum_x * sum_x / area);
cross_y2 = sum_y2 - (sum_y * sum_y / area);

slope = cross_xy / cross_x2;
intercept = (sum_y/(area)) - slope * (sum_x/(area));

angle_radian = atan(slope);
a_angle = atan(slope) * (180.0/pi);

if (a_angle < 0)
    a_angle = a_angle + 180.0;
    angle_radian = angle_radian + pi;
end



dmin_x = MAX_TWOD_DATA_LENGTH*3;
dmin_y = BITS_PER_SLICE;
dmax_x = 0;
dmax_y = 0;

if ( (angle_radian > (pi/2.0)) & (angle_radian <= (pi)))
    angle_radian = (pi - angle_radian);
elseif ( angle_radian > pi)
    ['HEY: something is wrong here  a_angle = ', num2str(a_angle)];
    return
end
for i=1:n_count
    new_x = (p(i).x * cos(angle_radian)) + (p(i).y * sin(angle_radian));
    new_y = (p(i).y * cos(angle_radian)) - (p(i).x * sin(angle_radian));
    if (new_x > dmax_x)
        dmax_x = new_x;
    end
    if (new_y > dmax_y)
        dmax_y = new_y;
    end
    if (new_x < dmin_x)
        dmin_x = new_x;
    end
    if (new_y < dmin_y)
        dmin_y = new_y;
    end
end

d_length = (dmax_x - dmin_x) +1;
w_width = (dmax_y - dmin_y) +1;

r2_correlation = (cross_xy) / (sqrt( cross_x2 * cross_y2));
```

```
    F_fine_detail = p_perimeter_change * (d_length)/ area;
    if (partial_on_count ~=0 )
        S_ratio = fully_on_count / partial_on_count;
    else
        S_ratio = 0.0;
    end
end
%/*************************************************************************/


%/*************************************************************************/
function result = parabola_fit_center_is_in(image_buffer, n_slices)


    result = 1;

    return;

end
```

MATLAB code – calc_sa_randombins.m

```
% Calculate image sample area assuming Heymsfield and Parish (1978)
%   bins_mid - mid-point of each bins in doide number
%   res - photodiode resolution, bin width in microns
%   armdst - distance between probe arms in millimeters
%   num_diodes - number of photodiodes (does not need to equal number of
bins)
%   SAmethod - method to calculate SA. Could be 0: center in, 1: entire in
and 2 with correction
%
% ** Created to replace calc_sa to include different bin setup, and three
choices of SA
%    calculation methods. Notice the parameter differences.        Will,
2014/06/04
function sa = calc_sa_randombins(bins_mid,res,armdst,num_diodes, SAmethod)

% calculates OAP SA in mm^2 with res in um, armdist in mm


res = res * 1e-3; %mm
%max_diameter = max_diameter*.5e-3;


radius = bins_mid/2; %radius = bins_mid .* res/2;  You can use this one if
you provide midpoint in doide number
diameter = 2 * radius;

% Calculate the width
switch SAmethod
    case 0
        % Center in
        EAWci = num_diodes*res;
        EAWri = EAWci;
    case 1
        % Entire in
        EAWci = (num_diodes-(bins_mid/res)+1)*res;
        EAWci(EAWci<0)=0;
        EAWri = EAWci;
    case 2
        % With Correction
        EAWci = num_diodes*res;
        EAWri = EAWci + 0.72*diameter;
end

% Calculate the DOF
lambda = 680 * 1e-6;              % mm,laser wavelength
DOF = 6*radius.^2/lambda;         % Using
DOF(DOF > armdst) = armdst;

sa = DOF .* EAWri;
```

MATLAB code – read_binary_image.m

```
function read_binary_image(infilename,outfilename,project)

%%%
% project codes:
% 0 = TWP-ICE
% 1 = ISDAC
% 2 = RACORO

starpos = find(infilename == '*',1,'last');

if ~isempty(starpos)
    files = dir(infilename);
    filenums = length(files);
    filedir = infilename(1:starpos-1);
else
    filenums = 1;
end

for i = 1:filenums
    if filenums > 1
        infilename = [filedir,files(i).name];
    end

    if outfilename == '1'
        slashpos = find(infilename == '/',1,'last');
        outfilename = ['caps.',infilename(slashpos+1:end),'.cdf'];
    end

    fid=fopen(infilename,'r','l');
    infilename

    %%% Updated for new MATLAB NETCDF interface
    f = netcdf.create(outfilename, 'clobber');

    dimid0 = netcdf.defDim(f,'time',netcdf.getConstant('NC_UNLIMITED'));
    dimid1 = netcdf.defDim(f,'ImgRowlen',8);
    dimid2 = netcdf.defDim(f,'ImgBlocklen',1700);

    varid0 = netcdf.defVar(f,'year','double',dimid0);
    varid1 = netcdf.defVar(f,'month','double',dimid0);
    varid2 = netcdf.defVar(f,'day','double',dimid0);
    varid3 = netcdf.defVar(f,'hour','double',dimid0);
    varid4 = netcdf.defVar(f,'minute','double',dimid0);
    varid5 = netcdf.defVar(f,'second','double',dimid0);
    varid6 = netcdf.defVar(f,'millisec','double',dimid0);
    varid7 = netcdf.defVar(f,'wkday','double',dimid0);
    varid8 = netcdf.defVar(f,'data','double',[dimid1 dimid2 dimid0]);
    netcdf.endDef(f)

%     f = netcdf(outfilename,'clobber');
%     %f = netcdf('clobber');
%
%     f('time') = 0;
```

```
%     f('ImgRowlen') = 8;
%     f('ImgBlocklen')= 1700;
%
%     f{'year'} = 'time';
%     f{'month'} = 'time';
%     f{'day'} = 'time';
%     f{'hour'} = 'time';
%     f{'minute'} = 'time';
%     f{'second'} = 'time';
%     f{'millisec'} = 'time';
%     f{'wkday'} = 'time';
%
%     f{'data'} = {'time','ImgBlocklen','ImgRowlen'};

    % recsize = f('time');

%     autoscale(f{'data'},1);
%     autonan(f{'data'},1);



    %
    %
bytes=['00';'C0';'43';'00';'01';'81';'00';'F0';'41';'0F';'AA';'AA';'AA';'AA';
'AA';'AA';'AA';'AA';'89';'9E';'91';'AA';'3C';'66';'6C';'67';'41';'03';'7F'...
    %     ;'00';'FC';'FF';'43';'03'];

    kk=1;

    endfile = 0;

    if project == 1
        discard = fread(fid,1,'uint16'); %
    end

    % while feof(fid)==0 & kk <= 3000
    while feof(fid)==0 & endfile == 0
        if project == 0

            year=fread(fid,1,'uint16');
            month=fread(fid,1,'uint16');
            day=fread(fid,1,'uint16');
            hour=fread(fid,1,'uint16');
            minute=fread(fid,1,'uint16');
            second=fread(fid,1,'uint16');
            millisec=fread(fid,1,'uint16');
            wkday=fread(fid,1,'uint16');

        elseif project == 1
            %%%% added for isdac files

            year=fread(fid,1,'uint16');
            while(year ~= 2008)
                year = fread(fid,1,'uint16');
            end
            month=fread(fid,1,'uint16');
```

```
            day=fread(fid,1,'uint16');
            hour=fread(fid,1,'uint16');
            minute=fread(fid,1,'uint16');
            second=fread(fid,1,'uint16');
            millisec=fread(fid,1,'uint16');
            wkday=fread(fid,1,'uint16');
            discard = fread(fid,1,'uint16'); %

            %test = fread(fid,1,'uint16');
            %if test == 2008
            %    month=fread(fid,1,'uint16');
            %    day=fread(fid,1,'uint16');
            %    hour=fread(fid,1,'uint16');
            %    minute=fread(fid,1,'uint16');
            %    second=fread(fid,1,'uint16');
            %    millisec=fread(fid,1,'uint16');
            %    wkday=fread(fid,1,'uint16');
            %    discard = fread(fid,1,'uint16'); %
            %else
            %    fseek(fid,-16,'cof');
            %end
        elseif project == 2

            year=fread(fid,1,'uint16');
            month=fread(fid,1,'uint16');
            day=fread(fid,1,'uint16');
            hour=fread(fid,1,'uint16');
            minute=fread(fid,1,'uint16');
            second=fread(fid,1,'uint16');
            millisec=fread(fid,1,'uint16');
            wkday=fread(fid,1,'uint16');

        else
            return
        end
        %%%%

        data = fread(fid,4096,'uchar');
        %     data=reshape(fread(fid,4096*8,'ubit1'),4096,8);
        %     b1 =
[num2str(data(:,1)),num2str(data(:,2)),num2str(data(:,3)),num2str(data(:,4)),
num2str(data(:,5)),...
        %         num2str(data(:,6)),num2str(data(:,7)),num2str(data(:,8))];

        bytes=dec2hex(data,2);
        kk;

        i=1;
        ii=1;
        b1full=dec2bin(hex2dec(bytes(:,:)),8);
        b2 = bin2dec(b1full(:,4:8));


        while i<4096
            b1 = b1full(i,:);
```

```
    curi = i;
    i=i+1;
    if b1(3) == '1'
        %             i=i+1;
    elseif b1(1) == '0' & b1(2) == '0'
        %            b2=bin2dec(b1(4:8));
        for k=1:b2(curi)+1;
            if i < length(bytes)
                decomp(ii,:)=bytes(i,:);
            else break
            end
            ii=ii+1;
            i=i+1;
        end
    elseif b1(1) == '1' & b1(2) == '0'
        %            b2=bin2dec(b1(4:8));
        for k=1:b2(curi)+1;
            decomp(ii,:)='00';
            ii=ii+1;
        end
    elseif b1(2) == '1' & b1(1) == '0'
        %            b2=bin2dec(b1(4:8));
        for k=1:b2(curi)+1;
            decomp(ii,:)='FF';
            ii=ii+1;
        end
    else
        kk
    end
end



found = 0;
i=1;
count=0;
while found == 0
    if decomp(i)=='AA'
        count=count+1;
    else
        count=0;
    end

    if count == 8
        found=1;
        dd=i+1:8:length(decomp)-7;
    end
    i=i+1;

end

if project == 0
    discard=fread(fid,6,'uint16');  %%%%%%%% twpice
elseif project == 1
    discard=fread(fid,2,'uint16');  %%%%% for isdac proc
elseif project == 2
```

```
        end

        %
        %
decomp_convert=[hex2dec(decomp(dd,:)),hex2dec(decomp(dd+1,:)),hex2dec(decomp(
dd+2,:)),hex2dec(decomp(dd+3,:)),...
        %
hex2dec(decomp(dd+4,:)),hex2dec(decomp(dd+5,:)),hex2dec(decomp(dd+6,:)),hex2d
ec(decomp(dd+7,:))];

decomp_convert=[hex2dec(decomp(dd+7,:)),hex2dec(decomp(dd+6,:)),hex2dec(decom
p(dd+5,:)),hex2dec(decomp(dd+4,:)),...

hex2dec(decomp(dd+3,:)),hex2dec(decomp(dd+2,:)),hex2dec(decomp(dd+1,:)),hex2d
ec(decomp(dd,:))];

k2=[decomp(dd,:),decomp(dd+1,:),decomp(dd+2,:),decomp(dd+3,:),decomp(dd+4,:),
decomp(dd+5,:),decomp(dd+6,:),decomp(dd+7,:)];

        %             length_diff=length(decomp_convert) -
length(handles.matrix(kk-1,:,:));
        %             matrix_size(kk)=length(decomp_convert);
        %             if length_diff > 0
        %                 handles.matrix(1:kk-1,length(handles.matrix(kk-
1,:,:)):length(decomp_convert),:)=-1;
        %             elseif length_diff < 0
        %
decomp_convert(length(decomp_convert):length(handles.matrix(kk-1,:,:)),:)=-1;
        %             end
        if length(decomp_convert) < 1700
            decomp_convert(length(decomp_convert):1700,:)=-1;
        end
        %     recsize(:) = kk;
%         f{'year'}(kk) = year;
%         f{'month'}(kk) = month;
%         f{'day'}(kk) = day;
%         f{'hour'}(kk) = hour;
%         f{'minute'}(kk) = minute;
%         f{'second'}(kk) = second;
%         f{'millisec'}(kk) = millisec;
%         f{'wkday'}(kk) = wkday;
%
%
%         f{'data'}(kk,:,:)=decomp_convert(:,:);

        netcdf.putVar ( f, varid0, kk-1, 1, year );
        netcdf.putVar ( f, varid1, kk-1, 1, month );
        netcdf.putVar ( f, varid2, kk-1, 1, day );
        netcdf.putVar ( f, varid3, kk-1, 1, hour );
        netcdf.putVar ( f, varid4, kk-1, 1, minute );
        netcdf.putVar ( f, varid5, kk-1, 1, second );
        netcdf.putVar ( f, varid6, kk-1, 1, millisec );
        netcdf.putVar ( f, varid7, kk-1, 1, wkday );
        netcdf.putVar ( f, varid8, [0, 0, kk-1], [8,1700,1], decomp_convert'
);
```

```
        kk=kk+1;
        if mod(kk,100) == 0
            kk
            datestr(now)
        end
        clear decomp dd k2 b1 b2

        for j=1:4132
            bb=fread(fid,1,'int8');
            if feof(fid) == 1
                endfile=1;
                break
            end
        end
        fseek(fid,-4132,'cof');

    end

    fclose(fid);
    % close(f);
    netcdf.close(f);  % New interface by Will
end
```

MATLAB code – single_area.m

```
function iwc=single_area(diameter, habit)
%
% Used to calculate the particle area using A-D relations
% This is in cgs units
%
% Created on Feb 14, 2014 by Will Wu
%
iwc = 0;
switch char(habit)
    %case 't'  % Sphere and Tiny
        %iwc=0.91*3.1415926/6*diameter^3;
        %a=0.049;
        %b=2.8;
        %iwc = a *diameter^b;
    case {'l','o'}  % Linear and Oriented
        if diameter<0.03
            a=0.0696;
            b=1.5;
        else
            a=0.0512; %0.000907;
            b=1.414;
        end
        iwc = a *diameter^b;
    case 'h'  % Plate
        a=0.65;
        b=2.0;
        iwc = a *diameter^b;
    case {'i','a','s','t'}       % Inregular
        a=0.2285;
        b=1.88;
        iwc = a *diameter^b;
    case {'g'}       % Graupel
        a=0.5;
        b=2.0;
        iwc = a *diameter^b;
    case 'd'       % Dendrite
        a=0.21;
        b=1.76;
        iwc = a *diameter^b;
%     case 'a'       % Aggregate
%         a=0.0033;
%         b=2.2;
%         iwc = a *diameter^b;
end

end
```

MATLAB code – calc_vt.m

```
%% Returns terminal velocity for a single particle
%  Both options to calculate the terminal velocity
%  Default is to use the Heymsfield and Westbrook (2010) method,
%  but you can also choose to to use Mitchel (1996)
%  Created by Will Wu, 2014/01/15
%  - Mass and Diameter uses metric system
%  - Pressure use hPa
%  - Temperature use Celsius
function vt = single_vt(diameter, area_ratio, mass, P, T)

usingMithcell=0; % Setting 0 to use Heymsfield method, other value for
Mitchell method
g=9.8;
pi=3.1415926;

% Calculate environmental conditions
T = T + 273.15;
P = P*100;
rho_a = P/(287.15*T);
eta = 18.27*(291.15+120)./(T+120)*(T/291.15)^(3/2)/10^(6);  % Sutherland's
formula to calculate the dynamic viscosity
nu = eta/rho_a;  % kinectic viscosity

if 0==usingMithcell
    % Calculate modified Best Number using Heymsfield and Westbrook (2010).
    % using drag C=0.35 and epsilon=8.0
    X=rho_a/(eta^2)*8*mass*g/(pi*area_ratio^0.5);
    ReynoldN=16*(sqrt(1+4*sqrt(X)/64/sqrt(0.35))-1)^2;
else
    % Calculate modified Best Number using Mitchell (1996).
    % This is actually a special case of Heymsfield and Westbrook (2010),
with
    % k=0, and here we use drag C=0.6 and epsilon=5.83
    % We calculate from the original equations without using power law
    % approximation
    X=rho_a/(eta^2)*8*mass*g/(pi*area_ratio);
    ReynoldN=5.83^2/4*(sqrt(1+4*sqrt(X)/(5.83^2)/sqrt(0.6))-1)^2;
end

vt=nu/diameter*ReynoldN;

end
```

MATLAB code – dropsize_new.m

```
% largest edge, smallest edge, array_dim_length in pixels
%% image_buffer - raw 2DC record of single image - time by # of bytes in
array
%% largest_edge - # of pixels shaded in edge with greatest number of pixels
shaded
%% smallest_edge - # of pixels shaded in edge with least number of pixels
shaded
%% num_diodes_in_array - # of diodes in the array
%% time_diode_size - length of diode in time (X) direction. May not be equal
to that in array direction if clocking problem present.
%% array_diode_size - length of diode in array (Y) direction.
function max_dim = dropsize_new(image_buffer, largest_edge, smallest_edge,
num_diodes_in_array, time_diode_size, array_diode_size, array_dim_length)


%% The assumed buffers are either 32 or 64 diodes wide, standard 2DC format

%% Convert decimal record to binary

if(num_diodes_in_array == 32)
   temp =
[dec2bin(image_buffer(:,1),8),dec2bin(image_buffer(:,2),8),dec2bin(image_buff
er(:,3),8),...
    dec2bin(image_buffer(:,4),8)];
elseif(num_diodes_in_array == 64)
   temp =
[dec2bin(image_buffer(:,1),8),dec2bin(image_buffer(:,2),8),dec2bin(image_buff
er(:,3),8),...
    dec2bin(image_buffer(:,4),8), dec2bin(image_buffer(:,5),8),
dec2bin(image_buffer(:,6),8), dec2bin(image_buffer(:,7),8),...
    dec2bin(image_buffer(:,8),8)];
end

s = size(temp);
max_dim = 0;
xdim = 0;                                   %
ydim = 0;

%% Apply the Pythagorean theorem, using all possible X (time) and Y (array)
lengths as legs to get all possible hypotenuses - longest hypotenuse is
maximum dimension
for(i=1:s(1))
   for(j=1:s(2))
      %
      [min_pos_lite, max_pos_lite, time_length] = scan_slice(temp(:,j));
% Get positions, maximum length of particle in X direction at point (i,j)
      time_length = max_pos_lite-min_pos_lite+1;
% Length = position of last shadowed pixel - position of first shadowed pixel
+ 1
      [min_pos_lite, max_pos_lite, array_length] = scan_slice(temp(i,:));
% Get positions, maximum length of particle in Y direction at point (i,j)
      array_length = max_pos_lite-min_pos_lite+1;
      d = (time_length*time_diode_size)^2 +
(array_length*array_diode_size)^2;
```

```
      max_dim = max([max_dim d]);

   end
end

%% Square root only done once to minimize calcluation time - above method is
mathematically equivalent whether sqrt is done here or above

max_dim = sqrt(max_dim);

%% Heymsfield and Parish 1978 image reconstruction

d2 = 0;                                                 % If image not touching
edge, reconstructed diameter = 0 --> use maximum dimension above
if(smallest_edge+largest_edge > 0)                      % If image touches edge -
calculate Heymsfield and Parish diameter
   xa = smallest_edge*time_diode_size;
   xb = largest_edge*time_diode_size;
   d2 = sqrt((array_dim_length*array_diode_size+.25*(xb^2-
xa^2)/(array_dim_length*array_diode_size))^2+xa^2);
end

max_dim = max([d2 max_dim]);                          %% Use greater of
reconstructed diameter, maximum dimension

return;

%% Scans a slice for the position of last, first lit pixel
function [min_pos_lite,max_pos_lite,n_lite]=scan_slice(image_buf)

n_lite=0;
max_pos_lite=0;
min_pos_lite=0;


zeros = find(image_buf == '0');
n_lite = length(zeros);
if n_lite == 0
    return
else
    min_pos_lite = zeros(1);
    max_pos_lite = zeros(n_lite);
end

return
```

MATLAB code – ImgView.m

```
function varargout = ImgView(varargin)
% IMGVIEW M-file for ImgView.fig
%      IMGVIEW, by itself, creates a new IMGVIEW or raises the existing
%      singleton*.
%
%      H = IMGVIEW returns the handle to a new IMGVIEW or the handle to
%      the existing singleton*.
%
%      IMGVIEW('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in IMGVIEW.M with the given input arguments.
%
%      IMGVIEW('Property','Value',...) creates a new IMGVIEW or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before ImgView_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to ImgView_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help ImgView

% Last Modified by GUIDE v2.5 09-Jan-2013 11:33:04

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @ImgView_OpeningFcn, ...
                   'gui_OutputFcn',  @ImgView_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before ImgView is made visible.
function ImgView_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to ImgView (see VARARGIN)
```

```
% Choose default command line output for ImgView
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes ImgView wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = ImgView_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in PrevButton.
function PrevButton_Callback(hObject, eventdata, handles)
% hObject    handle to PrevButton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
if(handles.frame > 1)
    handles.frame = handles.frame - 1;
    imageview(handles.ImgAxes, handles.imgcdf, handles.frame,
handles.rec_num, handles.reject, handles.procdf);

    set(handles.FrameNo, 'String', num2str(handles.frame));
%     hour = handles.imgcdf{'hour'}(handles.frame);
%     minute = handles.imgcdf{'minute'}(handles.frame);
%     second = handles.imgcdf{'second'}(handles.frame);
    hour =
netcdf.getVar(handles.imgcdf,netcdf.inqVarID(handles.imgcdf,'hour'),handles.f
rame-1,1);
    minute =
netcdf.getVar(handles.imgcdf,netcdf.inqVarID(handles.imgcdf,'minute'),handles
.frame-1,1);
    second =
netcdf.getVar(handles.imgcdf,netcdf.inqVarID(handles.imgcdf,'second'),handles
.frame-1,1);
    set(handles.TimeBox, 'String', [num2str(hour, '%02d') num2str(minute,
'%02d') num2str(second, '%02d')]);
    set(handles.FrameNo, 'String', num2str(handles.frame));
    guidata(hObject, handles);
end


% --- Executes on button press in NextButton.
function NextButton_Callback(hObject, eventdata, handles)
% hObject    handle to NextButton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
if(handles.frame < handles.noFrames)
    handles.frame = handles.frame + 1;
    imageview(handles.ImgAxes, handles.imgcdf, handles.frame,
handles.rec_num, handles.reject, handles.procdf);

%     hour = handles.imgcdf{'hour'}(handles.frame);
%     minute = handles.imgcdf{'minute'}(handles.frame);
%     second = handles.imgcdf{'second'}(handles.frame);
    hour =
netcdf.getVar(handles.imgcdf,netcdf.inqVarID(handles.imgcdf,'hour'),handles.f
rame-1,1);
    minute =
netcdf.getVar(handles.imgcdf,netcdf.inqVarID(handles.imgcdf,'minute'),handles
.frame-1,1);
    second =
netcdf.getVar(handles.imgcdf,netcdf.inqVarID(handles.imgcdf,'second'),handles
.frame-1,1);

    set(handles.TimeBox, 'String', [num2str(hour, '%02d') num2str(minute,
'%02d') num2str(second, '%02d')]);
    set(handles.FrameNo, 'String', num2str(handles.frame));
    guidata(hObject, handles);
end



function FrameNo_Callback(hObject, eventdata, handles)
% hObject    handle to FrameNo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of FrameNo as text
%        str2double(get(hObject,'String')) returns contents of FrameNo as a
double
no = str2double(get(hObject, 'String'));
if(no > 0 && no < handles.noFrames+1)
    handles.frame = no;
    imageview(handles.ImgAxes, handles.imgcdf, handles.frame,
handles.rec_num, handles.reject, handles.procdf);
    guidata(hObject, handles);
end

% --- Executes during object creation, after setting all properties.
function FrameNo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to FrameNo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --------------------------------------------------------------------
function FileMenu_Callback(hObject, eventdata, handles)
% hObject    handle to FileMenu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --------------------------------------------------------------------
function OpenMenu_Callback(hObject, eventdata, handles)
% hObject    handle to OpenMenu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[file, path] = uigetfile({'*.cdf','All CDF Images (*.cdf)'; '*.*', 'All Files
(*.*)'}, 'Select an image CDF file');

if(~isequal(file,0))
    oldpath = pwd;
    cd(path);
    handles.imgcdf = netcdf.open(file, 'nowrite');
    handles.frame = str2double(get(handles.FrameNo, 'String'));
    % New I/O
    year =
netcdf.getVar(handles.imgcdf,netcdf.inqVarID(handles.imgcdf,'year'));  %
handles.imgcdf{'year'};
    handles.noFrames = length(year);
    cd(oldpath);
    [file, path] = uigetfile({'*.cdf','All CDF Images (*.cdf)'; '*.*', 'All
Files (*.*)'}, 'Select an autoanalysis CDF file');
    if(~isequal(file,0))
        cd(path);
        proccdf = netcdf.open(file, 'nowrite');
        % New NETCDF I/O by Will 02/06/2013
%         handles.reject = proccdf{'image_auto_reject'}(:);
%         handles.palength = proccdf{'image_length'}(:);
%         handles.rec_num = proccdf{'parent_rec_num'}(:);
        handles.reject =
netcdf.getVar(proccdf,netcdf.inqVarID(proccdf,'image_auto_reject'));
        handles.palength =
netcdf.getVar(proccdf,netcdf.inqVarID(proccdf,'image_length'));
        handles.rec_num =
netcdf.getVar(proccdf,netcdf.inqVarID(proccdf,'parent_rec_num'));
        handles.procdf=proccdf;
        cd(oldpath);

        imageview(handles.ImgAxes, handles.imgcdf, handles.frame,
handles.rec_num, handles.reject, handles.procdf);
        guidata(hObject, handles);

    end
end
```

```
% --------------------------------------------------------------------
function ExitMenu_Callback(hObject, eventdata, handles)
% hObject    handle to ExitMenu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
ncclose;
delete(gcf);




set(axes_handle, 'Position', new_pos)
title(axes_handle, [num2str(imgcdf{'hour'}(frame)) ':'
num2str(imgcdf{'minute'}(frame)) ':' num2str(imgcdf{'second'}(frame)) '.'
num2str(imgcdf{'millisec'}(frame))])
colormap(FlagColorMap);
imagesc(get(axes_handle, 'XLim'), get(axes_handle, 'YLim'), ni', [1 6]);
axis off;



function TimeBox_Callback(hObject, eventdata, handles)
% hObject    handle to TimeBox (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TimeBox as text
%        str2double(get(hObject,'String')) returns contents of TimeBox as a
double

% hour = handles.imgcdf{'hour'}(:);
% minute = handles.imgcdf{'minute'}(:);
% second = handles.imgcdf{'second'}(:);
hour = netcdf.getVar(handles.imgcdf,netcdf.inqVarID(handles.imgcdf,'hour'));
minute =
netcdf.getVar(handles.imgcdf,netcdf.inqVarID(handles.imgcdf,'minute'));
second =
netcdf.getVar(handles.imgcdf,netcdf.inqVarID(handles.imgcdf,'second'));

time = str2double(get(hObject, 'String'));

imgtime = (hour(:))*10000 + (minute(:))*100 + (second(:));

index = find(imgtime >= time,1);
if(~isempty(index))
    handles.frame = index;
    imageview(handles.ImgAxes, handles.imgcdf, handles.frame,
handles.rec_num, handles.reject);
    set(handles.FrameNo, 'String', num2str(handles.frame));
%     hour = handles.imgcdf{'hour'}(handles.frame);
%     minute = handles.imgcdf{'minute'}(handles.frame);
%     second = handles.imgcdf{'second'}(handles.frame);
```

```
    hour =
netcdf.getVar(handles.imgcdf,netcdf.inqVarID(handles.imgcdf,'hour'),frame-
1,1);
    minute =
netcdf.getVar(handles.imgcdf,netcdf.inqVarID(handles.imgcdf,'minute'),frame-
1,1);
    second =
netcdf.getVar(handles.imgcdf,netcdf.inqVarID(handles.imgcdf,'second'),frame-
1,1);

    set(handles.TimeBox, 'String', [num2str(hour, '%02d') num2str(minute,
'%02d') num2str(second, '%02d')]);
    guidata(hObject, handles);
end

% --- Executes during object creation, after setting all properties.
function TimeBox_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TimeBox (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



% --------------------------------------------------------------------
function SaveImage_Callback(hObject, eventdata, handles)
% hObject    handle to SaveImage (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[file, path] = uiputfile({'*.jpg','All JPG Images (*.jpg)'; '*.*', 'All Files
(*.*)'}, 'Enter name of JPEG file');

if(~isequal(file,0))
    oldpath = pwd;
    cd(path);
    %imagedata = handles.imgcdf{'data'}(1, :, :);
    imagedata =
netcdf.getVar(handles.imgcdf,netcdf.inqVarID(handles.imgcdf,'data'),[0, 0,
0], [8,1700,1]);
    dims = size(imagedata);

    figure('Position', [100 100 100+dims(1) 100+dims(2)*8]);
    no = str2double(get(handles.FrameNo, 'String'));
    axes
    set(gca, 'units', 'characters');
    imageview(gca, handles.imgcdf, no, handles.rec_num, handles.reject);
    p = get(gca, 'Position');
    new_pos(1) = p(1);
    new_pos(2) = p(2);
    new_pos(3) = (p(4)-1)*(dims(1)/(dims(2)*8));
```

```
    new_pos(4) = p(4)-1;
    set(gca, 'Position', new_pos);
    %title([num2str(handles.imgcdf{'hour'}(no)) ':'
num2str(handles.imgcdf{'minute'}(no)) ':'
num2str(handles.imgcdf{'second'}(no)) '.'
num2str(handles.imgcdf{'millisec'}(no))]);

title([num2str(netcdf.getVar(handles.imgcdf,netcdf.inqVarID(handles.imgcdf,'h
our'),no-1, 1)) ':' ...

num2str(netcdf.getVar(handles.imgcdf,netcdf.inqVarID(handles.imgcdf,'minute')
,no-1, 1)) ':' ...

num2str(netcdf.getVar(handles.imgcdf,netcdf.inqVarID(handles.imgcdf,'second')
,no-1, 1)) '.' ...

num2str(netcdf.getVar(handles.imgcdf,netcdf.inqVarID(handles.imgcdf,'millisec
'),no-1, 1))])

    % Expand figure slightly after drawing to reveal title

    set(gcf, 'PaperPositionMode', 'auto');
    print(gcf, '-djpeg', '-r600', file);
    guidata(hObject, handles);
    cd(oldpath);
end
```

MATLAB code – imageview.m

```
% imageview - Plot a 2D image record onto the given axis
%     axis_handle - handle pointing to axis you wish to draw the 2D image
%     on
%     imgcdf - netCDF file handle for uncompressed image netCDF
%     frame - # of record to draw
%     rec_num - parent_rec_num from autoanalysis netCDF
%     reject - image_auto_reject from autoanalysis netCDF
function imageview(axes_handle, imgcdf, frame, rec_num, reject, proccdf)
FlagColorMap = [1 1 1; 0 0 0; 1 0 0; 0 1 0; 0 0 1; 0 1 1];


%imagedata = imgcdf{'data'}(frame, :, :);
imagedata = netcdf.getVar(imgcdf,netcdf.inqVarID(imgcdf,'data'), ...
    [0, 0, frame-1], [8,1700,1]);
imagedata=imagedata';
dims = size(imagedata);
newimage = zeros(dims(1), dims(2)*8);
i = 1;
k = 2;
y = 1;
endslice = [170 170 170 170 170 170 170 170];
invalidslice = [-1 -1 -1 -1 -1 -1 -1 -1];

NormalColor = 2;                                                % Black
FlagStuckBitColor = 3;                                          % Blue
FlagRejectColor = 3;                                            % Red
FlagHollowColor = 4;                                            % Green

%timeentry = imgcdf{'hour'}(frame)*10000 + imgcdf{'minute'}(frame)*100 +
imgcdf{'second'}(frame);
timeentry = netcdf.getVar(imgcdf,netcdf.inqVarID(imgcdf,'hour'),frame-1,
1)*10000 + netcdf.getVar(imgcdf,netcdf.inqVarID(imgcdf,'minute'),frame-1,
1)*100 + netcdf.getVar(imgcdf,netcdf.inqVarID(imgcdf,'second'),frame-1, 1);

particlepos = find(rec_num == frame,1);


if(~isempty(particlepos))
  CurColor = FlagRejectColor;
  if(reject(particlepos) == '0')
       CurColor = NormalColor;
  end
  if((reject(particlepos) == 'h') || (reject(particlepos) == 'H') ||
(reject(particlepos) == 'u'))
        CurColor = FlagHollowColor;
  end
else
    CurColor = NormalColor;
end

% Find the first autoanalysis entry associated with this image
p = 0;
```

```
while(k<dims(1))


    % Get particle header first
    %particleno = uint8(imagedata(k, 1:2));
    %date = uint8(imagedata(k, 3:7));
    %slicecount = bitshift(uint8(imagedata(k,8)), -1);
    %slicecount = mod(uint8(imagedata(k,8)), 128)
    if(isequal(imagedata(k,1:3),[170 170 170]) == 0 &&
isequal(imagedata(k,1), 85) == 0)
        if(isequal(imagedata(k, :),invalidslice)==0)

           for(i=1:dims(2))
            for(j=1:8)
               if(imagedata(k, i) == -1)
                   newimage(y, (i-1)*8+j) = FlagRejectColor;
               else
                   newimage(y, (i-1)*8+j) = CurColor-
CurColor*bitget(uint8(imagedata(k, i)), 9-j);
               end
            end
           end
           try
                particlenum(y) = particlepos;
           end
           y = y + 1;
           p = p + 1;
        else
           k = dims(1);
        end
         k = k + 1;
    else
        newimage(y,:) = 6;
        y = y + 1;
        if(imagedata(k,1) == 85)
             k = k + 1;
        end
        k = k + 2;
        p = 0;
        if(~isempty(particlepos))
            CurColor = FlagRejectColor;
            if(particlepos > length(reject))
                particlepos = length(reject)
            end
            if(reject(particlepos) == '0')
                CurColor = NormalColor;
            end
            if(reject(particlepos) == 'h' || reject(particlepos) == 'H' ||
reject(particlepos) == 'u')
                CurColor = FlagHollowColor;
            end
            particlepos = particlepos + 1;
        end
    end
```

```
end

ni = newimage(1:y,:);
axes(axes_handle);

% Set aspect ratio to scale - change width to adjust to height of ImgAxes
p = get(axes_handle, 'Position');
new_pos(1) = p(1);
new_pos(2) = p(2);
new_pos(3) = (p(4))*(dims(1)/(dims(2)*8));
new_pos(4) = p(4);

title(axes_handle,
[num2str(netcdf.getVar(imgcdf,netcdf.inqVarID(imgcdf,'hour'),frame-1, 1)) ':'
...
    num2str(netcdf.getVar(imgcdf,netcdf.inqVarID(imgcdf,'minute'),frame-1,
1)) ':' ...
    num2str(netcdf.getVar(imgcdf,netcdf.inqVarID(imgcdf,'second'),frame-1,
1)) '.' ...
    num2str(netcdf.getVar(imgcdf,netcdf.inqVarID(imgcdf,'millisec'),frame-1,
1))])
%title(axes_handle, [num2str(imgcdf{'hour'}(frame)) ':'
num2str(imgcdf{'minute'}(frame)) ':' num2str(imgcdf{'second'}(frame)) '.'
num2str(imgcdf{'millisec'}(frame))])
colormap(FlagColorMap);
h=imagesc(get(axes_handle, 'XLim'), get(axes_handle, 'YLim'), ni', [1 6]);
set(h, 'ButtonDownFcn',@particle_info)
axis off;
title(frame);

    function particle_info(o,e)
        try
            close(m)
        end

        q = get(gca,'CurrentPoint');
        xlims = get(gca, 'XLim');
        mod = xlims(1);
        xlims = xlims -mod;
        xdex = q(1,1) - mod;
        index = floor((xdex)/xlims(2)*length(ni));
        pdex = particlenum(index)-1;
        details=analyzer_prep(proccdf,pdex-1,1);

        Message = sprintf('Particle number is %g \n Particle length is %g \n
Particle width is %g \n Particle area is %g \n Particle diameter is %g \n
Reject status is %c \n Inter-arrival time is %g \n Percent Shadowed Area is
%g \n Habit is %c \n Hollow status is %g \n Max Hole diameter is %g \n', ...
                pdex, details.length, details.width, details.area,
details.diameter, details.reject, details.interarrival, details.shadowed,
details.habit, details.hollow, details.holediam);
        m = msgbox(Message);
```

```
    end
end
```

MATLAB code – read_binary_image_plows.m

```
% Converts a binary images in the PMS-2D format to the format used by the
% code designed by Matt Freer/Robert Jackson for ISDAC/RACORO in the
processing
% of size distributions in terms of maximum dimension and area ratio
% Right now, we read Fast-2DC and 2DP. This will probably need to be
% modified to match the PLOWS data later on, which I can do easily.

function read_binary_image_plows(infile, cdffile)

%/* Possible values for the 'id' field. */
%#define PMS2D_C1         0x4331        // First 2DC
%#define PMS2D_C2         0x4332        // Second 2DC
%#define PMS2D_C4         0x4334         // 64 bit 25 um Fast 2DC
%#define PMS2D_C6         0x4336         // 64 bit 10 um Fast 2DC
%#define PMS2D_G1         0x4731        // First 2D Greyscale
%#define PMS2D_G2         0x4732        // Second 2D Greyscale
%#define PMS2D_H1         0x4831        // First HVPS
%#define PMS2D_H2         0x4832        // Second HVPS
%#define PMS2D_P1         0x5031        // First 2DP
%#define PMS2D_P2         0x5032        // Second 2DP

%struct P2d_rec {
%  short id;                             /* 'P1','C1','P2','C2', H1, etc */
%  short hour;
%  short minute;
%  short second;
%  short year;                           /* starting in 2007 w/ PACDEX */
%  short month;                          /* starting in 2007 w/ PACDEX */
%  short day;                            /* starting in 2007 w/ PACDEX */
%  short tas;                            /* true air speed  */
%  short msec;                           /* msec of this record */
%  short overld;                         /* overload time, msec */
%  unsigned char data[4096];        /* image buffer     */
%  };
%typedef struct P2d_rec P2d_rec;

f = fopen(infile, 'r');

outcdf_2dc = netcdf.create([cdffile '.2dc.cdf'], 'clobber');
outcdf_2dc2 = netcdf.create([cdffile '.2dp.cdf'], 'clobber');
outcdf_2dc3 = netcdf.create([cdffile '.f2dc.cdf'], 'clobber');

% Fast-2DC and 2DP CDF creation

dimid0 = netcdf.defDim(outcdf_2dc,'time',netcdf.getConstant('NC_UNLIMITED'));
dimid1 = netcdf.defDim(outcdf_2dc,'ImgRowlen',4);
dimid2 = netcdf.defDim(outcdf_2dc,'ImgBlocklen',1024);

varid0 = netcdf.defVar(outcdf_2dc,'year','double',dimid0);
varid1 = netcdf.defVar(outcdf_2dc,'month','double',dimid0);
varid2 = netcdf.defVar(outcdf_2dc,'day','double',dimid0);
varid3 = netcdf.defVar(outcdf_2dc,'hour','double',dimid0);
varid4 = netcdf.defVar(outcdf_2dc,'minute','double',dimid0);
varid5 = netcdf.defVar(outcdf_2dc,'second','double',dimid0);
```

```
varid6 = netcdf.defVar(outcdf_2dc,'millisec','double',dimid0);
varid7 = netcdf.defVar(outcdf_2dc,'tas','double',dimid0);
varid8 = netcdf.defVar(outcdf_2dc,'data','double',[dimid1 dimid2 dimid0]);
netcdf.endDef(outcdf_2dc)

% outcdf_2dc('time') = 0;
% outcdf_2dc('ImgBlocklen') = 512;
% outcdf_2dc('ImgRowlen') = 8;
% outcdf_2dc{'year'} = 'time';
% outcdf_2dc{'month'} = 'time';
% outcdf_2dc{'day'} = 'time';
% outcdf_2dc{'hour'} = 'time';
% outcdf_2dc{'minute'} = 'time';
% outcdf_2dc{'second'} = 'time';
% outcdf_2dc{'millisec'} = 'time';
% outcdf_2dc{'tas'} = 'time';
% outcdf_2dc{'data'} = {'time', 'ImgBlocklen', 'ImgRowlen'};
dimid20 =
netcdf.defDim(outcdf_2dc2,'time',netcdf.getConstant('NC_UNLIMITED'));
dimid21 = netcdf.defDim(outcdf_2dc2,'ImgRowlen',4);
dimid22 = netcdf.defDim(outcdf_2dc2,'ImgBlocklen',1024);

varid20 = netcdf.defVar(outcdf_2dc2,'year','double',dimid20);
varid21 = netcdf.defVar(outcdf_2dc2,'month','double',dimid20);
varid22 = netcdf.defVar(outcdf_2dc2,'day','double',dimid20);
varid23 = netcdf.defVar(outcdf_2dc2,'hour','double',dimid20);
varid24 = netcdf.defVar(outcdf_2dc2,'minute','double',dimid20);
varid25 = netcdf.defVar(outcdf_2dc2,'second','double',dimid20);
varid26 = netcdf.defVar(outcdf_2dc2,'millisec','double',dimid20);
varid27 = netcdf.defVar(outcdf_2dc2,'tas','double',dimid20);
varid28 = netcdf.defVar(outcdf_2dc2,'data','double',[dimid21 dimid22
dimid20]);
netcdf.endDef(outcdf_2dc2)
% outcdf_2dc2('time') = 0;
% outcdf_2dc2('ImgBlocklen') = 512;
% outcdf_2dc2('ImgRowlen') = 8;
% outcdf_2dc2{'year'} = 'time';
% outcdf_2dc2{'month'} = 'time';
% outcdf_2dc2{'day'} = 'time';
% outcdf_2dc2{'hour'} = 'time';
% outcdf_2dc2{'minute'} = 'time';
% outcdf_2dc2{'second'} = 'time';
% outcdf_2dc2{'millisec'} = 'time';
% outcdf_2dc2{'tas'} = 'time';
% outcdf_2dc2{'data'} = {'time', 'ImgBlocklen', 'ImgRowlen'};

dimid30 =
netcdf.defDim(outcdf_2dc3,'time',netcdf.getConstant('NC_UNLIMITED'));
dimid31 = netcdf.defDim(outcdf_2dc3,'ImgRowlen',8);
dimid32 = netcdf.defDim(outcdf_2dc3,'ImgBlocklen',512);

varid30 = netcdf.defVar(outcdf_2dc3,'year','double',dimid30);
varid31 = netcdf.defVar(outcdf_2dc3,'month','double',dimid30);
varid32 = netcdf.defVar(outcdf_2dc3,'day','double',dimid30);
varid33 = netcdf.defVar(outcdf_2dc3,'hour','double',dimid30);
varid34 = netcdf.defVar(outcdf_2dc3,'minute','double',dimid30);
```

```
varid35 = netcdf.defVar(outcdf_2dc3,'second','double',dimid30);
varid36 = netcdf.defVar(outcdf_2dc3,'millisec','double',dimid30);
varid37 = netcdf.defVar(outcdf_2dc3,'tas','double',dimid30);
varid38 = netcdf.defVar(outcdf_2dc3,'data','double',[dimid31 dimid32
dimid30]);
netcdf.endDef(outcdf_2dc3)

ii = 1;
jj = 1;
kk = 1;
% First, we will read the XML header that is in the file

xml_header = fgetl(f)
pms2d = fgetl(f)
source = fgetl(f)
project = fgetl(f)
platform = fgetl(f)
line = [];
while(~isequal(line, '</OAP>'))
    line = fgetl(f);
end
[str,maxsize,endian] = computer;
while feof(f) == 0
    if(mod(ii+jj+kk,100) == 0)
        disp([num2str(ii+jj+kk) 'frames written']);
    end
    if(endian == 'L')
        id = swapbytes(uint16(fread(f,1,'uint16')));
        hour0 = swapbytes(uint16(fread(f,1,'uint16')));
        minute0 = swapbytes(uint16(fread(f,1,'uint16')));
        second0 = swapbytes(uint16(fread(f,1,'uint16')));
        year0 = swapbytes(uint16(fread(f,1,'uint16')));
        month0 = swapbytes(uint16(fread(f,1,'uint16')));
        day0 = swapbytes(uint16(fread(f,1,'uint16')));
        tas = double(swapbytes(uint16(fread(f,1,'uint16'))));
        msec0 = swapbytes(uint16(fread(f,1,'uint16')));
        overld0 = swapbytes(uint16(fread(f,1,'uint16')));
        data0 = uint8(fread(f,4096,'uint8'));
    else
        id = (fread(f,1,'uint16'));
        hour0 = (fread(f,1,'uint16'));
        minute0 = (fread(f,1,'uint16'));
        second0 = (fread(f,1,'uint16'));
        year0 = (fread(f,1,'uint16'));
        month0 = (fread(f,1,'uint16'));
        day0 = (fread(f,1,'uint16'));
        tas = double((fread(f,1,'uint16')));
        msec0 = (fread(f,1,'uint16'));
        overld0 = (fread(f,1,'uint16'));
        data0 = (fread(f,4096,'uint8'));
    end

    hour   = double(hour0);
    minute = double(minute0);
    second = double(second0);
    year   = double(year0);
```

```
    month  = double(month0);
    day    = double(day0);
    msec   = double(msec0);
    data   = double(data0);


    if(id == hex2dec('4331'))
%         outcdf_2dc{'year'}(i) = year;
%         outcdf_2dc{'month'}(i) = month;
%         outcdf_2dc{'day'}(i) = day;
%         outcdf_2dc{'hour'}(i) = hour;
%         outcdf_2dc{'minute'}(i) = minute;
%         outcdf_2dc{'second'}(i) = second;
%         outcdf_2dc{'millisec'}(i) = msec;
%         outcdf_2dc{'tas'}(i) = tas;
%         outcdf_2dc{'data'}(i,:,:) = reshape(data,8,512)';
        netcdf.putVar ( outcdf_2dc, varid0, ii-1, 1, year );
        netcdf.putVar ( outcdf_2dc, varid1, ii-1, 1, month );
        netcdf.putVar ( outcdf_2dc, varid2, ii-1, 1, day );
        netcdf.putVar ( outcdf_2dc, varid3, ii-1, 1, hour );
        netcdf.putVar ( outcdf_2dc, varid4, ii-1, 1, minute );
        netcdf.putVar ( outcdf_2dc, varid5, ii-1, 1, second );
        netcdf.putVar ( outcdf_2dc, varid6, ii-1, 1, msec );
        netcdf.putVar ( outcdf_2dc, varid7, ii-1, 1, tas );
        netcdf.putVar ( outcdf_2dc, varid8, [0, 0, ii-1], [4,1024,1],
reshape(data,4,1024));
        ii = ii + 1;
    elseif(id == hex2dec('5031'))
%         outcdf_2dc2{'year'}(j) = year;
%         outcdf_2dc2{'month'}(j) = month;
%         outcdf_2dc2{'day'}(j) = day;
%         outcdf_2dc2{'hour'}(j) = hour;
%         outcdf_2dc2{'minute'}(j) = minute;
%         outcdf_2dc2{'second'}(j) = second;
%         outcdf_2dc2{'millisec'}(j) = msec;
%         outcdf_2dc2{'tas'}(j) = tas;
%         outcdf_2dc2{'data'}(j,:,:) = reshape(data,8,512)';
        netcdf.putVar ( outcdf_2dc2, varid20, jj-1, 1, year );
        netcdf.putVar ( outcdf_2dc2, varid21, jj-1, 1, month );
        netcdf.putVar ( outcdf_2dc2, varid22, jj-1, 1, day );
        netcdf.putVar ( outcdf_2dc2, varid23, jj-1, 1, hour );
        netcdf.putVar ( outcdf_2dc2, varid24, jj-1, 1, minute );
        netcdf.putVar ( outcdf_2dc2, varid25, jj-1, 1, second );
        netcdf.putVar ( outcdf_2dc2, varid26, jj-1, 1, msec );
        netcdf.putVar ( outcdf_2dc2, varid27, jj-1, 1, tas );
        netcdf.putVar ( outcdf_2dc2, varid28, [0, 0, jj-1], [4,1024,1],
reshape(data,4,1024));
        jj = jj + 1;

     elseif(id == hex2dec('4335'))

        netcdf.putVar ( outcdf_2dc3, varid30, kk-1, 1, year );
        netcdf.putVar ( outcdf_2dc3, varid31, kk-1, 1, month );
        netcdf.putVar ( outcdf_2dc3, varid32, kk-1, 1, day );
        netcdf.putVar ( outcdf_2dc3, varid33, kk-1, 1, hour );
        netcdf.putVar ( outcdf_2dc3, varid34, kk-1, 1, minute );
```

```
        netcdf.putVar ( outcdf_2dc3, varid35, kk-1, 1, second );
        netcdf.putVar ( outcdf_2dc3, varid36, kk-1, 1, msec );
        netcdf.putVar ( outcdf_2dc3, varid37, kk-1, 1, tas );
        netcdf.putVar ( outcdf_2dc3, varid38, [0, 0, kk-1], [8,512,1],
reshape(data,8,512));
        kk = kk + 1;
    end
end
fclose(f);
%close(outcdf_2dc);
netcdf.close(outcdf_2dc);
netcdf.close(outcdf_2dc2);
netcdf.close(outcdf_2dc3);
```

MATLAB code – read_binary_image_ec.m

```
%% Reads Environment Canada format 2DC/2DP files

% Header:   There are 7 tag numbers and each contains 16 bytes of
% information. (int = 2 bytes, char = 1 byte)  See below structure...
%
%             struct RecordHeader{
%
% unsigned int tagNumber;
% unsigned int dataOffset;
% unsigned int number_of_Bytes;
% unsigned int number_of_samples;
% unsigned int bytesPerSample;
% unsigned char DataType;
% unsigned char paramete1;
% unsigned char parameter2;
% unsigned char parameter3;
% unsigned int address;
%
% } typedef RecordHeader;
%
% 2DC Tag numbers:
%
%      Tag  0    = date and time
%
%           7000 = 2DC image data
%
%           7006 = 2DC TAS
%
%           7007 = 2DC elapsed time
%
%           7008 = 2DC elapsed TAS
%
%           7009 = 2DC elapsed shadow
%
% 2DP Tag numbers
%
%      Tag  0    = date and time
%
%           8000 = 2DP image data
%
%           8006 = 2DP TAS
%
%           8007 = 2DP elapsed time
%
%           8008 = 2DP elapsed TAS
%
%           8009 = 2DP elapsed Shadow-OR
%
%
%
% Sample of a record header:
%
% 0   112    18     1    18   1   0   0   0 43605
%
%   7000   144  4096     1  4096   5   0 103   8  5888
```

```
%
%   7006   130     4     1     4   6   0   0   0  5888
%
%   7007   134     4     1     4   7   0   0   0  5888
%
%   7008   138     4     1     4   8   0   0   0  5888
%
%   7009   142     2     1     2   9   0   0   0  5888
%
%    999  4240     0     0     0   0   0   0   0 43605
%
%
%
% Followed by date and time starting byte # 113 ( 9 values and 2 bytes
% each )
%
%   2007    11    28    19    57    53       0   64             64
%
%   year   mon   day    hr   min    sec      millisec   max-freq
% (millisec = 1st value / 2nd value = 0 / 64)
%
%
%
% Followed by other information starting byte # 131 (2 values & 2 bytes
% each), 135 (1 value; 4 bytes), 139 (1 value; 4 bytes), 143 (1 value; 2
% bytes):
%
%      49   3    2376087   474740         0
%
%       TAS     elp-time     elp-tas   elp-sh   ( TAS has 2 values; 1st
% value = multiply factor=49 and 2nd value = divide factor=3 )
%
%
%
% Followed by image data starting byte 145.  The size of the image is 4096
% bytes.
%
%
%
%
% Here is the format of the 2D Mono data:
%
%
%
%             Header record:   112 bytes
%
%             Date & Time:     18 bytes
%
%             Other data:        14 bytes   ( Elp. Time, TAS, etc. )
%
%             2D Image:         4096 bytes
%
%
%
%             Total:    4240 bytes (fixed)          To read the image
% only, skip 144 bytes.
%
```

```
%

function read_binary_image_ec(infile)

in = fopen(infile, 'r');

cdffile = netcdf.create([infile '.cdf'], 'clobber');

cdffile('time') = 0;
cdffile('ImgBlocklen') = 1024;
cdffile('ImgRowlen') = 4;
cdffile{'year'} = 'time';
cdffile{'month'} = 'time';
cdffile{'day'} = 'time';
cdffile{'hour'} = 'time';
cdffile{'minute'} = 'time';
cdffile{'second'} = 'time';
cdffile{'millisec'} = 'time';
cdffile{'tas'} = 'time';
cdffile{'elapsed_time'} = 'time';
cdffile{'elapsed_tas'} = 'time';
cdffile{'shadow_or'} = 'time';
cdffile{'data'} = {'time', 'ImgBlocklen', 'ImgRowlen'};

dimid0 = netcdf.defDim(cdffile,'time',netcdf.getConstant('NC_UNLIMITED'));
dimid1 = netcdf.defDim(cdffile,'ImgRowlen',4);
dimid2 = netcdf.defDim(cdffile,'ImgBlocklen',1024);

varid0 = netcdf.defVar(cdffile,'year','double',dimid0);
varid1 = netcdf.defVar(cdffile,'month','double',dimid0);
varid2 = netcdf.defVar(cdffile,'day','double',dimid0);
varid3 = netcdf.defVar(cdffile,'hour','double',dimid0);
varid4 = netcdf.defVar(cdffile,'minute','double',dimid0);
varid5 = netcdf.defVar(cdffile,'second','double',dimid0);
varid6 = netcdf.defVar(cdffile,'millisec','double',dimid0);
varid7 = netcdf.defVar(cdffile,'tas','double',dimid0);
varid8 = netcdf.defVar(cdffile,'data','double',[dimid1 dimid2 dimid0]);
varid9 = netcdf.defVar(cdffile,'elapsed_time', 'double', dimid0);
varid10 = netcdf.defVar(cdffile,'elapsed_tas', 'double', dimid0);
varid11 = netcdf.defVar(cdffile,'shadow_or', 'double', dimid0);
netcdf.endDef(cdffile)

%% Read in the header first (144 bytes)
[str,maxsize,endian] = computer;
i = 1;
while (feof(in) == 0)
    % Skip the first 112 bytes
    %total_bytes = 0;
    %while(total_bytes < 112);
    %flag = fread(in, 1, 'uint16');
    %total_bytes = total_bytes + 2;
    %    if(flag == 7009)
        discard = fread(in, 112, 'uint8');
    %   total_bytes = total_bytes + 9;
    %    discard = fread(in, 1, 'uint8');5
```

```
%       shadow_or = fread(in, 1, 'uint16');
%        cdffile{'shadow_or'}(i) = shadow_or;
        %discard = fread(in, 1, 'uint8');
%       total_bytes = total_bytes + 3;
%   end
 %   end
    if(mod(i,100) == 0)
        disp([num2str(i) 'frames written']);
    end
year = 0;
if(feof(in) == 0)




        ii = ii + 1;
        while(year ~= 2008)
            year = fread(in, 1, 'uint16');
        end
        %cdffile{'year'}(i) = year;
         netcdf.putVar ( cdffile, varid0, ii-1, 1, year );
        month = fread(in, 1, 'uint16');
        netcdf.putVar ( cdffile, varid1, ii-1, 1, month );
        day = fread(in, 1, 'uint16');
        netcdf.putVar ( cdffile, varid2, ii-1, 1, day );
        hour = fread(in, 1, 'uint16');
        netcdf.putVar ( cdffile, varid3, ii-1, 1, hour );
        minute = fread(in, 1, 'uint16');
        netcdf.putVar ( cdffile, varid4, ii-1, 1, minute );
        second = fread(in, 1, 'uint16');
        netcdf.putVar ( cdffile, varid5, ii-1, 1, second );
        millisec1 = fread(in, 1, 'uint16');
        millisec2 = fread(in, 1, 'uint16');
        netcdf.putVar ( cdffile, varid6, ii-1, 1, millisec1/millisec2 );

        discard = fread(in, 2, 'uint8');
        tas1 = fread(in,1,'uint16');
        tas2 = fread(in,1,'uint16');
        netcdf.putVar ( cdffile, varid7, ii-1, 1, tas );

        elapsed_time = fread(in, 1, 'uint32');
        %cdffile{'elapsed_time'}(i) = ;                  % Elapsed time is in
25 us increments
        netcdf.putVar ( cdffile, varid9, ii-1, 1, elapsed_time*25e-6 );
        elapsed_time*25e-6
        elapsed_tas = fread(in, 1, 'uint32');
        %cdffile{'elapsed_tas'}(i) = elapsed_tas;
        netcdf.putVar ( cdffile, varid10, ii-1, 1, elapsed_tas );
        shadow_or = fread(in, 1, 'uint16');
        %cdffile{'shadow_or'}(i) = shadow_or;
        netcdf.putVar ( cdffile, varid11, ii-1, 1, shadow_or );
        raw = uint8(fread(in, 4096, 'uint8'));
        if(length(raw) == 4096)
            data = reshape(raw, 4, 1024)';
            data = fliplr(data);
```

```
            netcdf.putVar ( cdffile, varid8, [0, 0, ii-1], [4,1024,1],
reshape(data,4,1024));

        else
            netcdf.putVar ( cdffile, varid8, [0, 0, ii-1], [4,1024,1],
zeros(4,1024));
        end


end
    i = i + 1;
end
fclose(in);
netcdf.close(cdffile);
```