Lund University

Physics Department

Bachelor Thesis

Monitoring of heat exchanges in the snowpack of Foxfonna glacier, Spitzbergen, Svalbard.

Student :
Thibault Desjonquères

Supervisors :
Andy Hodson
Elna Heimdal Nilsson
Jean-Charles Gallet

Autumn 2021

LUNDS UNIVERSITET
Naturvetenskapliga fakulteten

LRCP-241

Partnering Institutions

UNIS
The University Centre in Svalbard

"The University Centre in Svalbard (UNIS) is the world's northernmost higher education institution, located in Longyearbyen at 78°N. We provide research-based education of the next generation of Arctic experts in biology, geology, geophysics and technology." - UNIS

NORSK·POLARINSTITUTT

"The Norwegian Polar Institute (NPI) is a directorate under the Ministry of the Climate and Environment. The Institute's activities are focused on environmental management needs in the polar regions. We produce scientific knowledge, conduct mapping and provide professional and strategic advice to the Norwegian authorities. We are present in the Arctic and the Antarctic, operate research stations, carry out expeditions and are the owners of the research vessel Kronprins Haakon. The institute is the environmental authority for Bouvetøya and for Norwegian activity in Antarctica." - NPI

Acknowledgements

I would like to thank Elna Heimdal Nilsson for encouraging me to pursue my studies in Svalbard assuring me of her supervision from Lund, and for giving me her support throughout this thesis work.

I would like to thank Andy Hodson for his supervision, his advice, feedback, and for giving me the opportunity to define my own project on a glacier. Having the chance to work on a glacier at a Bachelor level is indescribable. It made me enjoy my studies more than ever before.

I would like to thank Jean-Charles Gallet for his supervision, his very useful advice, feedback, and support throughout this work.

I would like to thank Tomi Vainio for his precious time, suggestions and help.

I would like to thank Ida Köhn for her support during the writing of our bachelor thesis. Dark season presented its share of challenges, and working alongside with her made it much more manageable.

Lastly, I would like to thank the people who assisted me out in the field, Petter Ideström, Jacob Qvam Skavang, Julia Steckling, Sondre Wennberg, Daria Laurentia Paul, Daniel Heineken, Domonik Kalke, Rainette Engbers, Maiike Zwier, Mariska Hoogkamer, Miriam Bentke, Lea Lipphardt, Ymke Lathouwers, Nil Rodes and Ida Köhn. It was a pleasure to carry heavy equipment over the tricky morraine and to the top of the glacier, to dig snowpits listening to disco music, and to finish our work sessions with some incredible (and not so controlled) stunts on fielski and snowboard. This thesis work required 11 outings, representing 7 to 8 hours of work each. It involved waking up early, hiking up to the glacier for 2 hours in difficult terrain, dragging heavy equipment on sledges, performing measurements in cold conditions for 3 to 4 hours and finally hike for 2 more hours with all the tiredness accumulated throughout the day. None of it would have been possible without their precious help and good attitude, in good and in difficult times.

”*OK guys, very important information; wherever you go in life, go beyond ! So here is me, on a beautiful glacier, trying snowboard. It's been an amazing pleasure, and we're going home for taco friday ! Woohoo !*” **- Jacob Qvam Skavang, 7th of May 2021, Foxfonna glacier** (message to our roommates while on our way back home, the first day in the field).

Abstract

The processes of meltwater storage in firn and the impact of refreezing on firn-covered glaciers have been investigated on the Greenland ice-sheet. On Svalbard, climate change promotes the disappearance of firn on glaciers. This thesis project aims to monitor the heat exchanges between the snowpack and the ice surface on a non firn-covered glacier, on Foxfonna, Spitzbergen, Svalbard, and to draw the picture of the events occurring in the snowpack over a melting season, from May to July 2021.

This report has two aims. First, to highlight the evolution of snow properties; bulk density, heat flow, *Snow Water Equivalent* (SWE, amount of water that would result from the melting of $1m^2$ surface area of that snowpack), and albedo, during melting, and attempts to do so during the formation of superimposed ice, e.g. refreezing of meltwater on the glacier ice. Second, to model the snow melt using local weather data and to compare the modelled results with the observed melt, measured periodically throughout the fieldwork period. Snow pits were dug to measure the snow temperature and density, and to observe the evolution of the snowpack properties. In the snow pits, the focus was set on the snow metamorphism, albedo, water content and stakes measurements. Finally, stakes measurements were realized to assess the growth of superimposed ice.

The weather data indicates a sudden warming on the 2^{nd} of June, which rendered the snowpack isothermal (0°C over the entire column), as observed from the snow temperature. The already established melt became stronger as a result of this warm event, as observed in the snow depth data. The snow melt modelled from the 6^{th} to the 11^{th} of June indicates results diverging from the actual snow melt by 95%. Nonetheless, the model describes the main physics of the system, as it predicts that melting must occur, as expected. The divergence between the modelled and the observed melt can be attributed to the use of weather stations located away from the site. As snow melted, the bulk density increased, resulting in a stronger heat flow within the snowpack over time. The inter-comparison of the SWE and the bulk density can reveal that snow metamorphism occurred, or that snow has accumulated on the glacier. Such inter-comparison was successfully carried out at the beginning of the fieldwork, revealing the accumulation of snow by some wind, in accordance with the snow depth and albedo data. Besides this one event, the SWE data did not reveal any clear trend over time, perhaps because of the interplay between the progressive melt, which should make it decrease, and the advection of water from up-glacier snow, which would make the SWE increase. Likely the location of the grid in a bowl prevented the analyses of the SWE data. Finally, the formation of superimposed ice was detected at the very end of the fieldwork. Its formation has been prevented by the fact that snowpack became suddenly isothermal by this warm event. With the cold content of the snowpack being entirely removed, not much refreezing occurred.

Popular Introduction

Refreezing of meltwater in snow and firn is known to have a major impact on the mass and energy budget of glaciers. In the Arctic, it is sometimes the primary accumulation process of some glaciers [1]. It is a major heating process, as the refreezing of 1g of water releases enough latent heat to raise the temperature of 160g of snow by 1°C.
Superimposed Ice (SI) is a layer of ice formed by the percolation of meltwater onto the subzero temperature glacier ice. The meltwater can percolate and transfer heat deep into the snow and the firn cover, and even reach the glacier ice surface, where the refreezing removes some of the cold content of the ice by latent heat release. As a fraction of the meltwater remains captured in the snow, this process may buffer the sea level rise caused by the increased glaciers melt [2]. The processes of meltwater storage in firn and the impact of refreezing on firn-covered glaciers have been investigated on the Greenland ice-sheet [2]. On Svalbard, climate change promotes the disappearance of firn on glaciers [3]. Gaining further knowledge about superimposed ice and the refreezing processes on a firn-free glacier is crucial for understanding and predicting the future of Arctic glaciers.

This thesis aims to monitor the heat exchanges through the snowpack on the firn-free Foxfonna glacier, Spitzbergen, Svalbard, over the melting season May-July 2021. Weather data were gathered, for multiple reasons. First, to link the weather conditions to the observed snow processes. Second, to model the snow melt. The modelled results were compared with field data. The evolution of the snow processes as it melts was assessed by making snow pits, and measuring temperature and density. The snow and the weather data reveals that the weather and snow conditions were not appropriate for the SI formation in 2021.

Contents

1 Introduction 1

2 Background 2

2.1 The weather of Svalbard . . . 2

2.2 Svalbard Glaciers . . . 2

2.3 Superimposed Ice . . . 2

3 Theory 5

3.1 Snow Properties . . . 5

3.1.1 Snow Metamorphism . . . 5

3.1.2 Snow Water Equivalent (SWE) & Bulk Density . . . 6

3.1.3 Heat flow in the snow . . . 7

3.1.4 Albedo . . . 7

3.2 Surface Energy Balance (SEB) . . . 8

3.2.1 Q_N : Net Radiations . . . 8

3.2.2 Q_H, Q_L : Turbulent Fluxes . . . 9

3.2.3 Q_R : Sensible heat flux of rain . . . 9

3.2.4 Q_g : Ground heat flux . . . 10

3.2.5 $Q_{Melt\ Experimental}$: Amount of energy resulting in the observed melt. 10

4 Method 11

4.1 Snow measurements and weather data . . . 11

4.2 Melt modeling . . . 12

5 Results 15

5.1 Melt Modeling . . . 15

5.2 Comparison between the modelled and the observed melt . . . 16

5.3 Evolution of snow properties . . . 17

5.3.1 Correlation between Heat Flow & Bulk Density increase . . . 17

5.3.2 Evolution of Bulk density and SWE, with respect to the snow depth data . . . 17

5.4 Superimposed ice measurement . . . 19

6 Discussion 22

6.1 SEB - Melt model . . . 22

6.2 Evolution of snow properties . . . 23

6.2.1 Correlation between Heat Flow & Bulk Density increase . . . 23

6.2.2 Evolution of Bulk density, SWE, with respect to snow depth data . 23

6.3 Isothermal snow and Superimposed ice . . . 24

7 Conclusion 25

A Appendix 28

A.1 Introduction : Map . . . 28

A.2 Method : Field procedures and tools summary . . . 29

A.3 Code : Melt Model & Plot of Weather Data . . . 31

A.4 Code : Plot of the Snow Pits Data . . . 44

1 Introduction

Superimposed ice (SI) is a layer of ice formed at the snow/glacier interface. It is produced by the percolation of meltwater to the bottom of the snowpack, where it refreezes, removing some of the cold content of the glacier ice, by the release of latent heat. SI plays a major role in the mass and energy budget of glaciers. On some Arctic glaciers, SI formation is the main process by which accumulation occurs. Moreover, SI formation acts as a water retention mechanism on glaciers. It has been suggested that the refreezing of water on glaciers can potentially buffer the sea-level rise resulting from the glaciers melt [2]. The increasing temperatures and precipitations recorded in Svalbard over the years because of climate change modify snowpack properties. More rain-on-snow events and melting imply more refreezing of water in the snowpack. This, combined with a warmer climate, induces a faster melt of the snow. This project aims at two things. First, investigating the evolution of the snow properties; heat flow, bulk density, Snow Water Equivalent (SWE), albedo, of the glacier surface temperature as the snowmelts and superimposed ice is formed by realising in-situ snow measurements. Second, to implement a snow melt model based on the local weather conditions, and compare it to the actually observed melt. The project takes place over the melting season, from May to July 2021.

The fieldwork site is located on Foxfonna glacier, Spitzbergen, Svalbard, as shown in Figure 1 in Appendix A.1. Weather data were gathered with a weather station placed at the measurement site next to Stake 4 (78°08'06N, 16°10'15E, 650m altitude), on Breinosa (78°14'75N, 16°03'90E, 520m altitude) and Adventdalen (78°12'10N, 15°49'41E, 5m altitude), to model the melt of the snow, and link it with the amount of refreezing occurring under the snowpack. The weather stations of Breinosa and Adventdalen belong to the Kjell Henriksen Observatory and UNIS respectively and are maintained regularly. The weather station on Foxfonna was placed by Andy Hodson in 2019 and has been removed at the term of this project. Snow temperature and density data were gathered by digging snow pits in a grid enclosed by stakes, and by a thermistors string located at the weather station and powered by the same battery. The stakes were used to assess the growth of superimposed ice by measuring their depth (Figure 2 in Appendix A.2).

2 Background

2.1 The weather of Svalbard

The archipelago of Svalbard is located between the latitudes 76°25'N and 80°50'N, and the longitudes 10°30'E and 28°10'E. Its land area is 62.450 km^2, 60% of it is covered by glaciers. In winter, it is fully covered with snow [4]. The archipelago has a mountainous topography. Its coastline is complex as it is formed by *fjords* - steep cliffs with narrow, long inlets, shaped by the glacial activity. The polar-maritime climate of the archipelago is unique, due to its topography and location. It is dictated by the cold polar air and water masses coming from the Arctic, the Arctic sea ice, the mountain glaciers, and the warm North Atlantic Ocean carried by the West Spitsbergen Current, which makes Svalbard the warmest place along this latitude. The phenomenon of Arctic Amplification causes the Arctic to warm three times faster than the global average in the rest of the world. Climate change will cause the temperatures in Svalbard to increase by 10 °C in 2100, compared to today, and will also cause increased precipitations and rain on snow events [5].

2.2 Svalbard Glaciers

Glaciers form by the accumulation and compaction of snow over time. Snow that has survived one melting season is called firn, or perennial snow. When firn densifies further, it turns into ice. Glaciers react primarily to changes in temperature and precipitations, which makes them a key indicator of climate change. Glaciers and ice caps play a major role in our ecosystem. They store about 68% of the world's freshwater resources [6]. In Svalbard, they are snow-covered about 6 to 9 months a year. Snow is one of the most reflective surfaces on Earth with a very high albedo, which is the ratio between the incoming and the outgoing solar radiation reflected by a surface. The retreat of snow and ice decreases Earth's albedo, triggering the positive ice-albedo feedback, as the Earth absorbs more energy, enhancing further the melt of snow and ice, resulting in more sea-level rise.

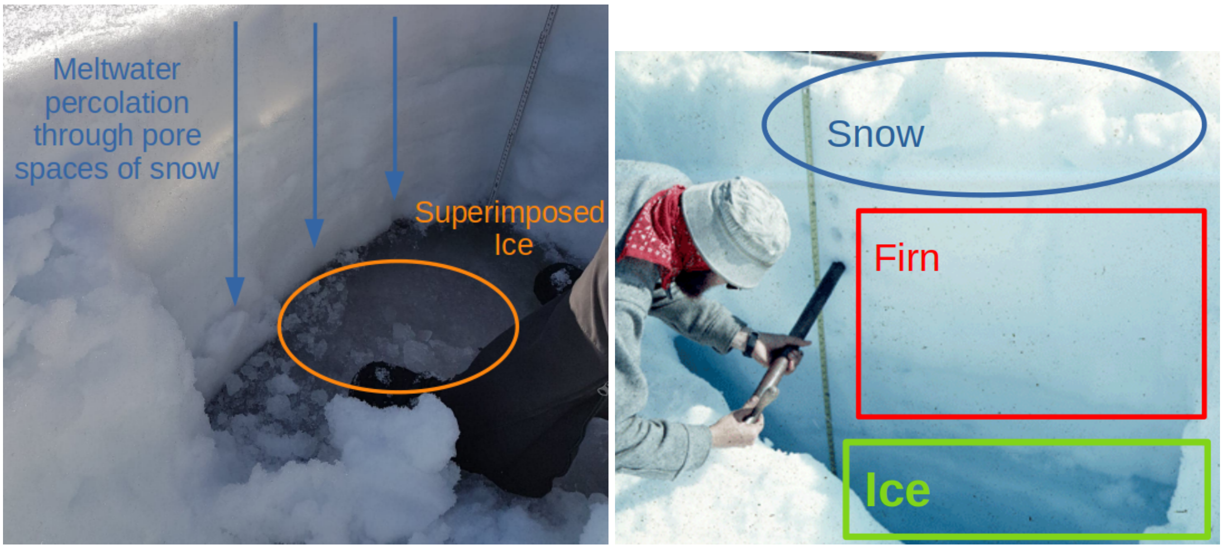
The thermal regime of a glacier defines the subglacial processes occurring there. There are three kinds of thermal regimes, warm, cold, and polythermal. Each of them exposes different characteristics. Cold glaciers have their internal structure at a temperature very much below the freezing point. Warm glaciers exist at temperatures very close to their melting point. Polythermal glaciers expose characteristics of both warm and cold glaciers [7]. Svalbard glaciers are often polythermal glaciers. On all three of these glacier types, SI formation can occur. The melt for glaciers on Svalbard has accelerated in the past decade [8], and is predicted to continue doing so due to climate change [9]. The absence of crevasses at the surface of Foxfonna glacier as well as its slow motion are clues that it is a cold glacier.

2.3 Superimposed Ice

Superimposed ice (SI) is a layer of ice forming on top of the glacier ice by the refreezing of percolating meltwater, as shown in Figure 1a. Its formation is first triggered by the warm-up of the snow cover to its melting point. The liberated meltwater can then percolate deeply into the snow, perennial snow, and eventually reach the snow/ice interface, transferring heat to it. If the bottom of the snowpack has enough cold content, it will be employed for refreezing the meltwater. The release of latent heat can have drastic consequences on the glacier thermal regime, as the heat released consumes progressively the cold content of the ice, warms it up, and will eventually modify the thermal regime of the glacier from cold to warm. The refreezing of the percolating meltwater occurs mostly in accumulation zones, where there is the most snow [10]. Refreezing acts as a mechanism of water retention on glaciers, and mass redistribution. Indeed, the porous character of snow and firn allows them to capture meltwater in the interstitial space existing between the individual grains, where it can sit until a cold wave makes it refreeze. SI formation can be a significant part of the mass and energy budget of glaciers. [11]. A study led on Midtre Lovenbreen, Svalbard, investigating the impact of SI on the mass balance of the glacier in 1999 reports that that year, SI formation accounted for 16-25% of the total accumulation. For some Arctic glaciers, SI formation is even the only accumulation process. [1].

The increasing temperatures observed in the context of climate change induce a greater glacier surface melt, and the disappearance of firn on Svalbard glaciers [3]. The role played by the snow and firn conditions and refreezing on the mass balance of a typical Svalbard glacier in a future climate has already been investigated [10]. In Winter, rain-on-snow or small melting events on a thick snowpack will induce the formation of internal ice lenses. This promotes the internal warming of the snow, reducing its refreezing potential and encourage its earlier melt as its cold content is reduced. Ultimately, it creates unfavourable conditions for the percolation of water to the surface of the ice and the formation of SI in Spring.

The processes of meltwater storage in firn have been investigated on the Greenland icesheet [2]. Firn increases the capacity of meltwater storage on glaciers, and therefore, the amount of internal refreezing occurring, until the pore spaces of firn have been exceeded, which leads to the outflow of water. Once the firn has been saturated with water, it hardly returns to its initial state. Most of the Svalbard glaciers are however not firn covered anymore, as it is the case for Foxfonna. Therefore, investigating the impact of the refreezing occurring on such glaciers is key to understand and predicting their response to climate change. The snow structure at the surface of a firn(covered glacier is represented in Figure 1b.



(a)

(b)

Figure 1: a) SI observed on the field. The ice layer lays below the snowpack and above the glacier ice. b) Structure of the snow layers at the surface of a glacier.[12]

Table 1 presents a summary of the objectives of the thesis.

Objectives	
I. Monitoring of the evolution of the snow properties during its melting and the formation of Superimposed Ice occurs.	**II. Modelling of the snow melt.**
Interests : - Temperature of snow and ice - Heat flow - Bulk density - Snow Water Equivalent (SWE) - Albedo	**Interests :** - Long-/Shortwave radiations - Relative Humidity - Wind speed & direction - Air temperature
Method : In-situ measurements of snow temperature, density, reflected shortwave radiaion.	**Method :** Weather stations in the area of Foxfonna

Table 1: Summary of the Thesis objectives.

3 Theory

3.1 Snow Properties

3.1.1 Snow Metamorphism

Snow is a porous material made of ice and air. It can contain impurities, such as black carbon and dust, which darken it and decrease its albedo. Snow is thermodynamically unstable. It is permanently subjected to metamorphism, which modifies its macrophysical properties, such as density, grain size, cohesive strength. As a consequence, other snow characteristics such as its albedo, permeability, and thermal conductivity also evolve over time. There are two metamorphism processes: constructive and destructive [13], [14]. Several factors, such as the initial crystal type, the amount of liquid water, the temperature gradient throughout the snow column dictates which metamorphism process takes place.

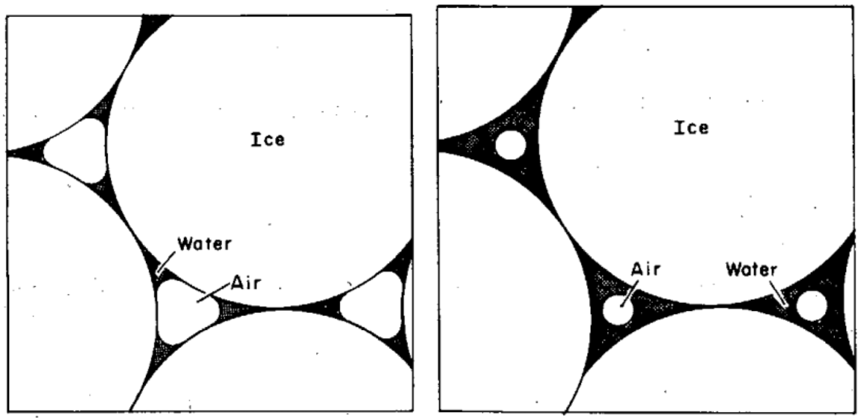
In dry snow, the metamorphism is mostly driven by the temperature gradient in the snowpack, which induces sublimation and condensation cycles, reshaping the snow grain size, and modifying snow properties. At locations where the temperature is higher, there is a higher saturation vapor pressure in the interstitial air, and sublimation can occur. The resulting water vapor then travels along a vapor pressure gradient created between the warmer and the colder snow layer, until it reaches a colder location where condensation can occur [13]. Vapor pressure is also driven by the crystals curvature radius, according to the Kelvin equation [15]. Crystals with larger curvature radius (i.e, the smallest snow grains) sublimate first and vapor condensate onto other larger crystals. The temperature gradient within the snowpack determines which one of the Kelvin effect or the vapor transport dominates the metamorphism process. Over time, the observed snow grains in the snowpack are often larger, as the smaller crystals have sublimated and only the larger ones remain [16].

In the absence of a strong temperature gradient within the snowpack (i.e. isothermal snowpack), snow metamorphism is dictated by the Kelvin effect, where the grains reduce their surface energy by growing onto the concave locations of the grain [16], resulting in the formation of small rounded grains, densifying the snowpack, reducing its permeability [17]. This very slow process occurs faster at warmer temperatures and is often defined as destructive metamorphism, as it produces rounded grains. Destructive metamorphism generally takes place if the temperature gradient is below 10 $K.m^{-1}$, [18].

When there is a strong temperature gradient within the snowpack (i.e. non-isothermal snowpack), dry snow metamorphism occurs by the advection of vapor. It results in the formation of faceted, sometimes very poorly bounded crystals such as depth hoar, making the snowpack more permeable [17]. The metamorphism occurring in such snowpack is called constructive, and it is driven by the strong temperature gradient[19]. Constructive metamorphism generally takes place if the temperature gradient is greater than 10 $K.m^{-1}$.

In wet snow, the metamorphism takes place in snow layers near 0°C. In such conditions, the water occupies the pore spaces partly, increasing the heat transfer within the snowpack [20]. The saturation regime of the snow dictates the type of metamorphism occurring in the snowpack. The thermodynamics of wet snow rules the processes of bond growth, grain growth and densification. The high and low saturation regimes differ in the way that the ice grains in the snowpack are still in contact with air at low saturation, as the water tends to locate at the grain boundary first. At high saturation, water can envelop the entire snow grains and disconnect them, so that they are slowly completely surrounded by water [19]. Wet snow metamorphism is destructive as only rounded shapes subsist.

In snowpacks with a low water content (i.e. pendular regime), the air pressure prevails over the water pressure, preventing the water to coalesce. The gaseous phase exists continuously throughout the snowpack [19]. The pendular regime is illustrated in Figure 2a. In snowpacks with a high water content (i.e. funicular regime), the water coalesces and traps the air away from the ice crystals. The funicular regime is illustrated in Figure 2b, from [19].



(a) Pendular Regime (b) Funicular Regime

Figure 2: Illustrations of the pendular and funicular regimes. [19]

3.1.2 Snow Water Equivalent (SWE) & Bulk Density

The SWE of a snowpack corresponds to the amount of water that would result from the melting of $1m^2$ surface area of that snowpack, expressed in mm water equivalent. SWE can augment by precipitation (freshly fallen snow), or by adding of material by percolation/runoff of water from other locations on the glacier due to the topography. The SWE [mm w.e.] of a single snow layer is calculated according to Equation 1, [21].

$$SWE = z_{layer} \times \frac{\rho_{snow}}{\rho_{water}} \tag{1}$$

Where z$_{layer}$ [mm] is the thickness of the considered snow layer, $\rho_{snow/water}$ [kg·m^{-3}] is the density of snow and water respectively.
The SWE of an entire snow column is calculated by summing up the SWE of each discrete layer according to Equation 2 (z$_i$ [mm] : thickness of the snow layer, numbered by i) :

$$SWE_{tot} = \int_{n=1}^{n=i} \rho_{snow}(z)[kg \cdot m^{-3}] \cdot z_i[mm]\, dz \cdot \frac{1}{1000\ [kg \cdot m^{-3}]} \tag{2}$$

The bulk density of the snow relates the total thickness of a snow column to its mass. [22]. It is calculated by dividing the SWE_{tot} by the thickness of the snowpack h_s, as shown in Equation 3, [23].

$$Bulk\,Density = \frac{SWE_{tot}\ [mm\ w.e.]}{h_s\ [mm]} \tag{3}$$

Snow depth and bulk density are inversely correlated over a short period of time, mainly because of three reasons. First, freshly fallen snow has a lower density than the bulk one. Snow density increases over time due to metamorphism and densification by gravity. The new snow crystal gets gradually destroyed and metamorphose [24]. Therefore, freshly fallen snow increases the snow depth, without considerably increasing the bulk density. Second, snow thickness decreases as the crystals undergo compaction, leading to an increase in the bulk density. Third, snowmelt reduced the snow thickness and increases bulk density by water percolation [22].

Changes in bulk density can reflect snow metamorphism and densification. In the case where no snowfall has occurred, if the SWE does not change and the bulk density increases, it means the snow became denser, by metamorphism or melting and refreezing.

3.1.3 Heat flow in the snow

The heat transfers in the snow are dictated by three manners. First, the conductivity through the interconnected ice crystals, second, the latent heat fluxes by the motion of water vapor and by condensation and sublimation cycles, and third, by convection through the air space. The heat flow across a snowpack can be calculated by Equation 4, [13].

$$q = -k_{effective}(\rho_{snow}) \cdot \frac{dT}{dz} \tag{4}$$

Where q [W.m^{-1}] is the flow across the snow, $k_{effective}$ [W.m^{-1}.K^{-1}] is the thermal conductivity of snow and is a function of the snow density (i.e. how tightly connected each individual snow grains are) and accounts for those three processes occurring in the snow, T [K] is the snow temperature and z [m] is the depth of the considered snow layer.
The effective thermal conductivity k of the snow accounts for the vapor diffusion and is empirically calculated with the snow density by Equation 5 [25].

$$k_{effective}(\rho_{snow}) = 2.22362 \cdot \rho_{snow}^{1.885} \tag{5}$$

3.1.4 Albedo

The albedo (α) of a surface characterises the amount of shortwave radiation this medium reflects. It is crucial to assess how much of the shortwave radiation is effectively absorbed and heats up and finally melts the snowpack. The albedo of snow is dictated by the snow grain size, its ageing, density, crystals shapes, and the number of impurities, typically black carbon, dust, small debris [26]. It is calculated by taking the ratio of the outgoing ($S\uparrow$) and incoming ($S\downarrow$) shortwave radiation at the snow surface, as shown in Equation 6, [7]. The range of the considered shortwave radiation is comprised between 0.28-2.5 µm.

$$\alpha = \frac{S\uparrow}{S\downarrow} \tag{6}$$

Fresh snow has the highest albedo. Over time, snow metamorphism increases the snow grain size, and thus decreases the snow albedo, warming up the snowpack. This can induce a change in the temperature gradient of the snowpack and modify the metamorphism type.

3.2 Surface Energy Balance (SEB)

The melt of the snow at the glacier surface is controlled by the SEB, which is the sum of all the energy fluxes, as shown in Equation 7, [20].

$$Q_N + Q_H + Q_L + Q_R + Q_G = Q_{Melt\ Model} \tag{7}$$

Where Q_N is the net radiation, Q_H is the sensible heat flux, Q_L is the latent heat flux, Q_R is the sensible heat flux of rain, Q_G is the ground heat flux, $Q_{Melt\ Model}$ is the estimated energy provided to the snow, to heat it up and ultimately melt it. All terms are expressed in [W.m^{-2}]

Most of the melt is attributed to the radiation factor Q_N, followed by the sensible heat factor Q_H. The latent heat factor contributes relatively less to the melt compared to these other terms ([20] Table 2). The SEB calculations are based on a review of glaciers melt processes and modelling [20] and glacier physics literature [7].

3.2.1 Q_N : Net Radiations

The sum of the shortwave and longwave radiation absorbed by a surface is the net radiation budget. Shortwave radiations are emitted by the sun. The amount of incoming shortwave radiation depends on the top of the atmosphere radiation (solar zenith angle), the atmosphere's transmissivity (cloudiness), and ground and surrounding topography. The longwave radiations budget is part of the thermal radiations emitted by any Earth's surface above 0 °K, with wavelengths between 3-100μm, which are reflected back to the surface by the atmosphere [27]. Therefore, clear skies offer the best conditions for Earth's surface cooling by longwave emission, as not much radiation is reflected back. Covered skies reflect parts of the longwave radiation, limiting this cooling mechanism, (water vapor is the most effective molecule in terms of natural greenhouse effect). The longwave radiation reflected by the atmosphere result from the water vapor, carbon dioxide and ozone molecules [20]. Therefore, longwave radiation varies accordingly with the cloudiness and the temperature.

Q_N can be expressed as according to Equation 8 ([20]) :

$$Q_N = S\downarrow \cdot (1-\alpha) + L\downarrow - L\uparrow \tag{8}$$

where α is the albedo, S $\downarrow$ [W.m^{-2}] is the incoming shortwave radiation, L$\downarrow$ [W.m^{-2}] is the incoming longwave radiation, and L$\uparrow$ [W.m^{-2}] is the outgoing longwave radiation. The outgoing longwave radiation L $\uparrow$ can also be calculated with the Planck law, as snow or ice surfaces radiate as a black body in the infrared [7].

$$L\uparrow = \sigma \cdot T_s^4 \tag{9}$$

where $\sigma = 5.670367 \cdot 10^{-8}$ W.m^{-2}.K^{-4} is the Stefan-Boltzmann constant, T_s [K] is the surface temperature.

3.2.2 Q_H, Q_L : Turbulent Fluxes

Latent heat fluxes and sensible heat fluxes are referred to as turbulent fluxes. Sensible heat is the energy associated with temperature changes, without phase changes, and is driven by the temperature gradient between the air and the surface. It is more efficient on a rough surface, where more turbulence occurs, facilitating the exchange of energy. The latent heat flux is the heat flux associated with the evaporation, condensation, sublimation and deposition of water vapor between the atmosphere and the snow. It is driven by the moisture gradient between the atmosphere and the surface. (Paterson, 1994 (see master thesis))
The sensible Q_H and latent Q_L heat flux are calculated according to the flux-gradient method by Equation 10 and Equation 11 ([7]):

$$Q_H = \rho_{air} \cdot c_p \cdot C_H \cdot u \cdot (T_z - T_0) \tag{10}$$

$$Q_L = \frac{0.622}{P_0} \cdot \overset{\circ}{\rho}_{air} \cdot L_{fusion} \cdot C_L \cdot u \cdot (e_z - e_0) = 22.2 \cdot C_L \cdot u \cdot (e_z - e_0) \tag{11}$$

Where 0.622 is the molecular weight of water vapor to dry air, $\overset{\circ}{\rho}_{air}$ [kg.m^{-3}] is the air density at standard pressure P$_0$ [Pa] , L$_f$ [J.kg$^-$1] is the latent heat of fusion, ρ [kg.m^{-3}] is the density, c_p = 1003 J.kg^{-1}.K^{-1} is the specific heat capacity of air at T=0°C , C is the exchange coefficient (C_H = 0.0019, C_L = 0.0015, empirical transfer coefficient including surface roughness and atmospheric stability, [7]) , u [m.s^{-1}] is the wind speed at the surface, T [K] is the temperature at the surface and 2m above it.

The vapor pressure e_z is calculated according to Tenten's equation for temperatures below 0°C [28] in Equation 12 :

$$e_z = \frac{RH}{100} \cdot 610.78\ [Pa] \cdot exp\frac{21.875 \cdot T}{265.5 + T} \tag{12}$$

e_z and e_0 [Pa] are the vapor pressure at some altitude z and at the surface respectively. e_0 is calculated assuming saturation (RH = 100%). T [°C] is the temperature at the surface and at the considered height.

3.2.3 Q_R : Sensible heat flux of rain

The sensible heat flux of rain Q_R [W.m^{-2}] is very often neglected in SEB calculations, as its contribution to melting is minor compared to the other fluxes. It is expressed as follow ([20]) :

$$Q_R = \rho_{water} \cdot c_w \cdot R \cdot (T_r - T_s)$$

(13)

Where ρ_{water} [kg.m^{-3}] is the density of water, c_w = 4200 J.kg^{-1}.K^{-1} is the specific heat capacity of water, R is the rainfall rate [mm.h^{-1}] , T_r and T_s [K] are the temperatures of the rain and the surface respectively.

3.2.4 Q_g : Ground heat flux

The ground heat flux Q_g [W.m^{-2}] in snow and ice is calculated using Equation 14 ([20]).

$$Q_g = \int_0^Z \rho(z) \cdot c_p \cdot \frac{\partial T}{\partial t} dz \tag{14}$$

Where z is the ice thickness [m], from the surface to the point where seasonal temperature variations are negligible. $\frac{\partial T\ [K]}{\partial t\ [s]}$ is the rate of change of the ice temperature. Q_g is positive when the flow is directed from the ice surface to the lowermost boundary [20]. That is, when heat is conveyed to the glacier ice.

The density of ice as a function of temperature is well documented [29]. The heat transferred to the ice usually comes from the snow above, so the snow density component requires to be measured periodically, as snow density changes over time by compaction and refreezing.

3.2.5 $Q_{Melt\ Experimental}$: Amount of energy resulting in the observed melt.

The amount of melt energy resulting in the observed melt $Q_{Melt\ Experimental}$ [W.m^{-2}] can be calculated according to Equation 15 :

$$Q_{Melt\ Experimental} = \frac{\Delta h_s \cdot \rho_{snow} \cdot L_{fusion}}{\Delta t} \tag{15}$$

where Δ h_s [m] is the change of snow thickness, Δ t[s] the duration of the melting period considered.

In the field, the snow density is measured for two reasons. First, to calculate the SWE and estimate whether precipitations occurred or not, and to estimate the amount of water available for refreezing the bulk density. Second, to calculate the snow heat conductivity, and ultimately, determine the evolution of the heat flow across the melting snow once combined with the snow temperature measurements. The albedo is measured to assess its evolution as snow is melting. The melt model consists of the calculation of each term described in 3.2, to finally estimate $Q_{Melt\ Model}$, the amount of energy provided by the atmosphere, heating up and melting the snowpack. The quality of the model is assessed by comparing $Q_{Melt\ Model}$ with $Q_{Melt\ Experimental}$, the amount of energy necessary to produce the melt observed in the field.

4 Method

4.1 Snow measurements and weather data

The fieldwork site is located on the glacier of Foxfonna, in Svalbard, as indicated in Figure 1. Measurements were taken from the 7^{th} of May 2021 until the 27^{th} of June 2021, about once every 7-10 days when it was possible. The fieldwork site was reached on skis, and all the equipment was dragged on sledges (Figure 2[a] in Appendix A.2). On the first day in the field, 5 stakes were installed, in a grid, as shown on the map. During each fieldwork session, the following measurements were carried on; the depth of the stakes was measured to assess the growth of SI (Figure 2[e] in Appendix A.2). As SI forms at the bottom of the stakes, the depth from the bottom of the ice to the top of the stakes must decrease. The snow depth was measured to evaluate the snowmelt and accumulation. The albedo of the snow was measured using a pyranometer (Figure 2[f] in Appendix A.2). This measurement is accurate only when the sky is cloud-free or fully covered, as clouds scatter the short wave radiations. This causes the instability of the radiation field and makes the measurement inaccurate. The pyranometer was held by hand, and the albedo was calculated by measuring the averaged incoming and outgoing shortwave radiation over five measurements. The greatest amount of divergence amongst five consecutive measurement of solar irradiance was 16%. The weather was favorable for albedo measurements five times. Three pits were dug each time, at random locations enclosed by the grid formed by the stakes, always at spaced from each other, as repeating snow processes measurements at the same place would disturb the system (Figure 2[b] in Appendix A.2). In each pit, the temperature was measured first, and the density was measured second, every 10 cm from the top to the bottom of the pit (Figure 2[c] in Appendix A.2). The tools used for the snow measurements are shown in (Figure 2[d] in Appendix A.2). A weather station placed close to the grid (by S4) recorded values of incoming shortwave radiation, air temperature (2m above the ground), and relative humidity. A thermistors string located by the weather station recorded continuously snow and glacier ice temperature, down to a depth of 12cm below the ice surface (at the start of the season).

The weather station on Foxfonna was placed in 2019 at Stake 4 (78°08'06N, 16°10'15E, 650 altitude). The wind sensor was broken. It does not include a longwave radiation sensor, it only measures the incoming shortwave radiation. Therefore, the wind data was collected from the weather station in Breinosa (78°14'75N, 16°03'90E, 520m altitude, 4km away from the fieldwork site) and the longwave radiation data was gathered from the weather station in Adventhalen (78°12'10N, 15°49'41E, 5m altitude, 12km away from the fieldwork site). Both weather stations belong to the Kjell Henriksen Observatory and UNIS respectively, and they are regularly maintained. The use of weather stations located away from the field site exposes some drawbacks as the topography is different. Indeed, both the wind and the radiation measurements are strongly dependant on the topography. The weather data is meant to be used as inputs in the melt model. Therefore, the use of weather stations located away from the site is expected to make the melt model less accurate.

The timeline of the fieldwork is represented in Figure 3.

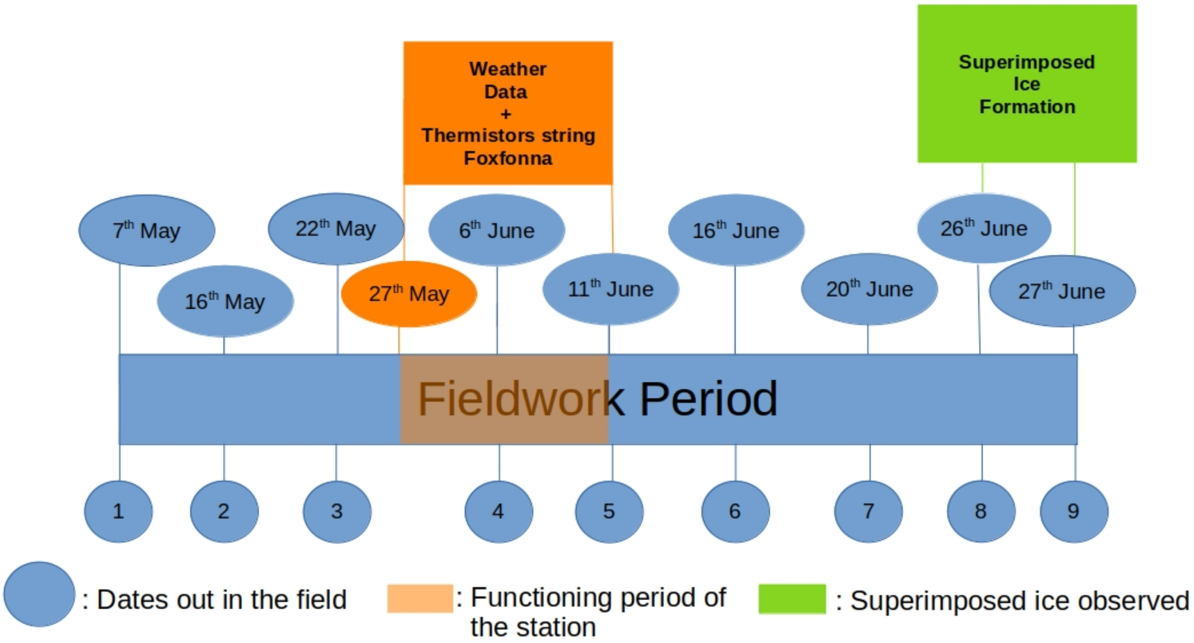
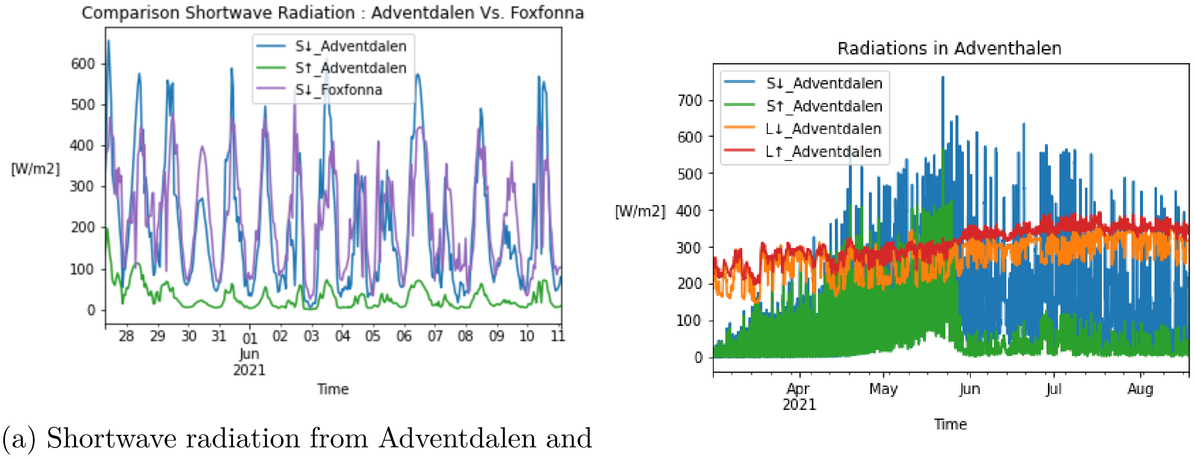


Figure 3: Fieldwork Timeline. Each outing required 7-8 hours of work.

4.2 Melt modeling

The melt model consists of two steps. First, the calculation of the each term described in 3.2; namely Q_N, Q_H, Q_L, Q_R, Q_G. Second, the addition of each result as described in Equation 7, to estimates $Q_{Melt\ Model}$, the amount of energy heating up and melting the snow. The calculation of each term in Equation 7 requires the acquisition of weather data. Both the weather station on Foxfonna and the thermistors string were powered by the same battery. The station was recording from the 27^{th} of May, until the 11^{th} of June. The melt has therefore been modelled within this time range. The melt model was coded using *Python* by myself based on the SEB theory, which is well documented [7], [20]. The two main codes are shown in subsection A.3.

Figure 4a shows the incoming and outgoing shortwave radiation recorded in Adventdalen, and the incoming shortwave radiation recorded in the field site. The incoming shortwave radiation at both sites is very similar. The differences between the two measurements can be attributed to the reflection of solar radiation by the surrounding topography. The outgoing shortwave measured in Adventdalen is much lower than the one emitted by a snow-covered surface. Figure 4b shows the radiation data recorded in Adventdalen over Spring and Summer. A sudden drop in the outgoing shortwave radiation measurement can be noted at the beginning of June, which must be attributed to the melt of the snow, (see Figure 4b, S↑), caused by the lower albedo of the ground. At this time of the year, the glacier was still snow-covered. The earlier snow melt in Adventdalen must be considered when using radiation data from this location, as the recorded outgoing shortwave and longwave radiation can no longer be considered as representative of the glacier's conditions once the snow has disappeared. The outgoing longwave radiation recordings do increase slightly from the 27^{th} of May as a result of the disappearance of the snow. When the snow has completely disappeared, the temperature of the ground increases progressively, emitting more longwave radiation.

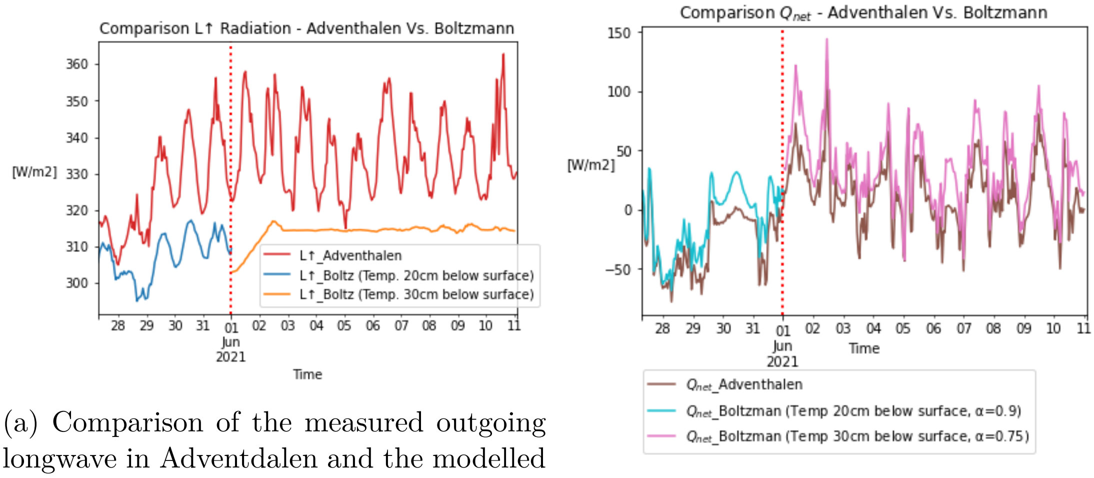


(a) Shortwave radiation from Adventdalen and Foxfonna (during fieldwork).

(b) Radiation data from Adventdalen.

Figure 4: a) Comparison of the shortwave radiation in Adventdalen Vs. Foxfonna. b) Radiations in Adventdalen.

In order to overcome the issue imposed by the location of the radiation sensors away from the site, at a lower elevation, the term $L\uparrow$ originally attributed to the outgoing longwave radiations data from Adventdalen could be replaced in the Q_{net} calculations of Equation 8 by the black body emission formula Equation 9, using the snow temperature recorded by a sensor from the thermistors string, located as close as possible to the surface. Figure 5a and Figure 5b illustrate the potential outcome of this solution, by comparing the resulting outgoing longwave radiation $L\uparrow$ and net radiation Q_{net} when this solution is applied. Over this time period, the snow melts, decreasing its thickness and albedo. Therefore, $L\uparrow$ is modelled with different surface temperatures, as shown in Figure 5a, and Q_{net} is modelled with an albedo evolving from $\alpha = 0.9$ to $\alpha = 0.77$ (according to the field measurements). From the 27^{th} of May, until the 1^{st} of June ($L\uparrow$ orange), the surface temperature is considered to be the one measured by the temperature sensor of the thermistors string located 122cm above the ice, about 20cm below the snow surface. From the 1^{st} until the 11^{th} of June, the surface temperature is considered to be the one measured by the temperature sensor situated 97cm above the ice, about 30cm below the snow surface. The choice of using a sensor located deeper in the snow from the 1^{st} of June is crucial to model the physics of the system. Indeed, the sensor located 122cm above the ice recorded positive temperatures from this date, while the snow temperature cannot exceed 0°C. Therefore, the longwave radiation emitted by a melting snow surface must not exceed a maximum of 316 $W.m^{-2}$, as modelled. The modelled outgoing longwave radiation is slightly lower than the measured one. This offset is because the now snow-free ground in Adventdalen had warmed up, and as a consequence, emits more longwave radiation. The daily variation of the snow temperature placed 122cm above the ice can be attributed to its proximity to the surface, rendering the measurement more sensitive. The recording of temperatures exceeding 0°C from this sensor, in spite of being below the snow surface can be attributed to the fact that the thermistors string might be heated by radiations and conducted heat along its cable to the sensors. Q_{net} is therefore calculated using the modelled outgoing longwave radiation, and the albedo evolving as observed in the field. The modelled net radiation balance is higher than the one obtained using the Adventdalen outgoing longwave radiation sensor. Therefore, the glacier surface is modelled to absorb more net radiations when it is calculated with the modelled longwave radiation than with the one calculated using the Adventdalen data.



(a) Comparison of the measured outgoing longwave in Adventdalen and the modelled one using the black body calculation. (Orange : sensor 122cm above ice, Green : sensor 97cm above ice.)

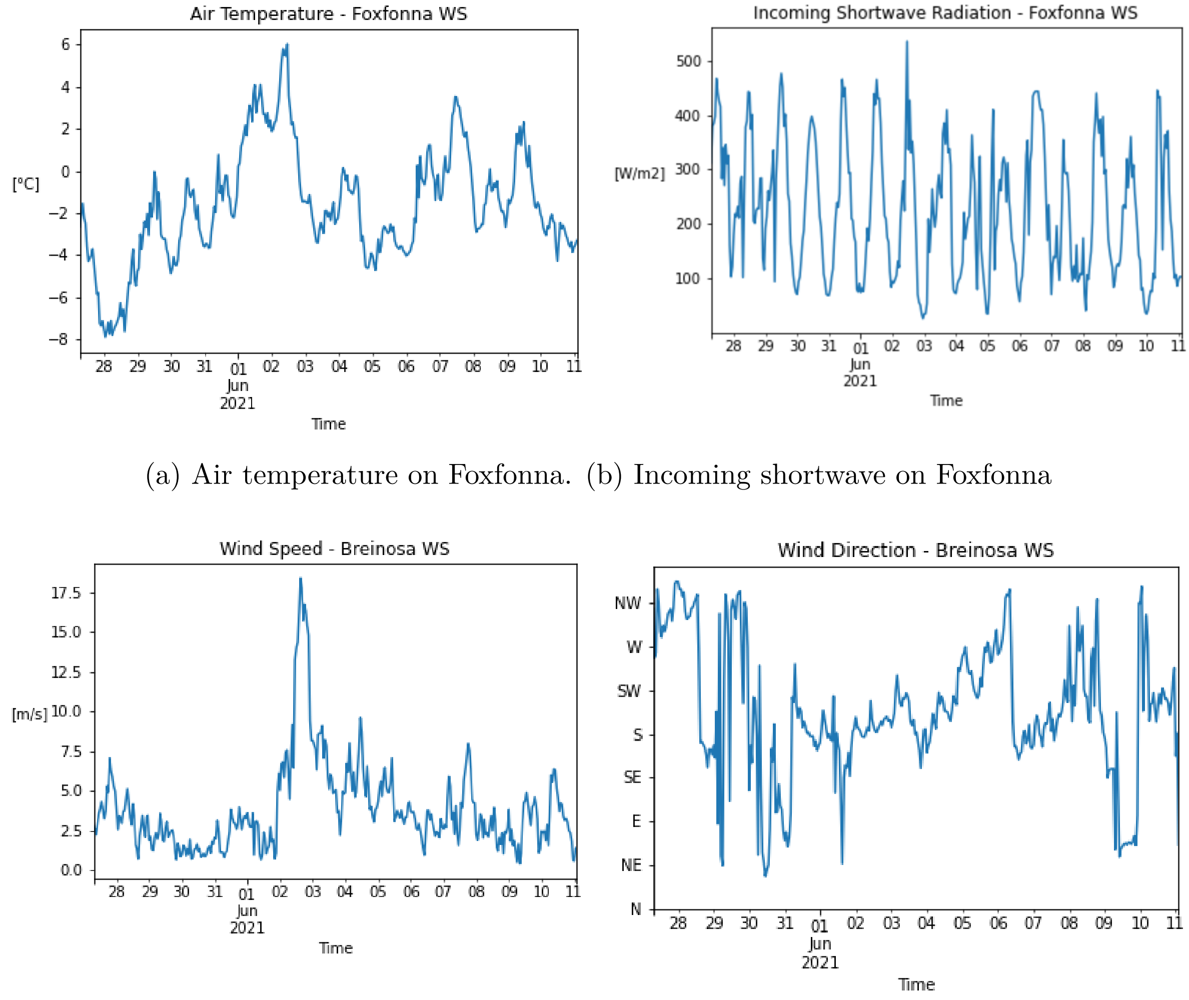
(b) Comparison of the net radiation calculated with the outgoing longwave measured in Adventdalen and the modelled one.

Figure 5: a) Comparison of the radiation data in Adventdalen and the modelled one with Equation 9. b) Comparison of Q_{net} when calculated with the modelled outgoing longwave and the outgoing longwave measured in Adventdalen.

5 Results

5.1 Melt Modeling

The weather data gathered by the Weather Stations (WS) on Foxfonna and on Breinosa are displayed in Figure 6. The wind and the temperature both peaked positively on the 2^{nd} of June, with the wind coming from the south. The incoming shortwave radiation data oscillates as a response of the diurnal cycle and of the shadowing from the surrounding topography.



(a) Air temperature on Foxfonna. (b) Incoming shortwave on Foxfonna

(c) Wind speed and (d) wind direction on Breinosa.

Figure 6: Weather data from Foxfonna and Breinosa.

Figure 7 is the modelled SEB. Each term used in the modelling of the melt is calculated as described in subsection 3.2, and summed to finally calculate Q_{melt} as shown in Equation 7. The data shows a sudden and strong melt peak on the 2^{th} of June. The sensible heat flux peaks on this day, as well as the ground heat flux. The latent heat flux is negative most of the time, which indicates that water evaporates from the surface to the atmosphere.

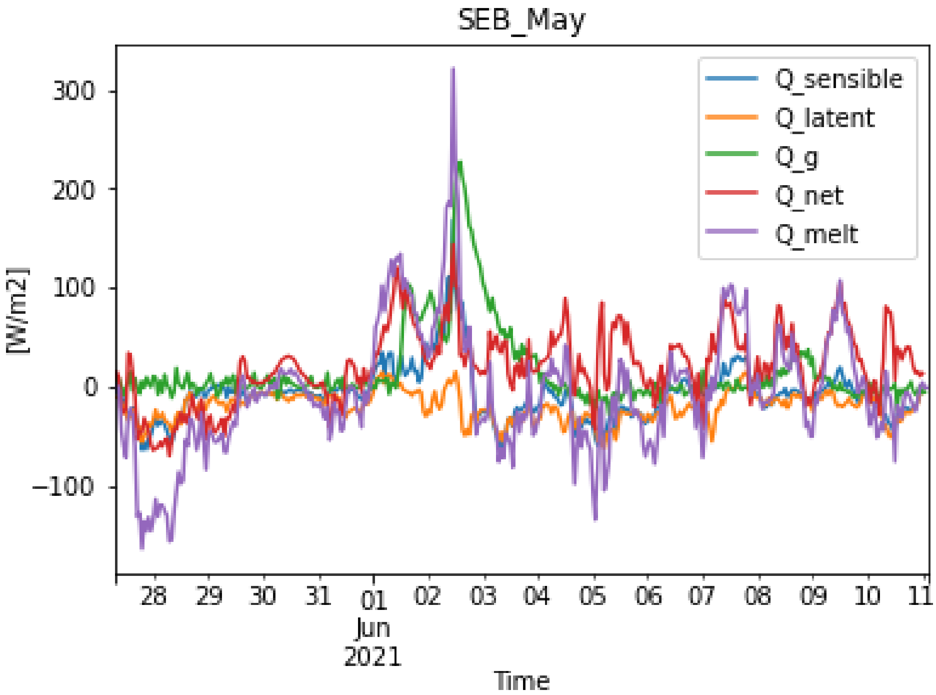


Figure 7: Surface Energy Balance. Q_{melt} is the net amount of energy received by the snowpack. Negative Q_{melt} means refreezing is occurring. Positive Q_{melt} means melting is occurring. Q_{melt} is obtained by summing the four other terms calculated from the weather data.

5.2 Comparison between the modelled and the observed melt

In order to estimate the quality of the melt model, one must compare the modelled results with the in-situ observation. On the 6^{th} and on the 11^{th} of June, both the snow depth data and the density measurements have been carried on. Therefore, the modelled results are compared with the observed melt between these dates. The amount of energy which implied the observed melt $Q_{Melt\ Experimental}$ [W.m^{-2}] is calculated using Equation 15, divided by the number of seconds between the two dates. On average, at each stake, 15cm of snow has melted between the 6^{th} and the 11^{th} of June, as it can be seen on Figure 9a. The value of the snow density used in this equation is the averaged density of two upper layers of snow (each layer being 10cm thick) of each of the three pits dug on the 6^{th} of June. The obtained result is then :

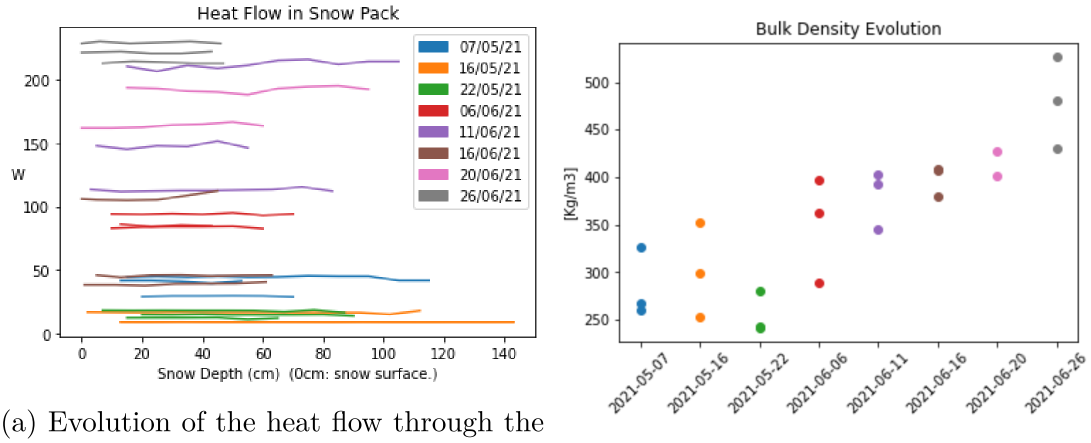
$$Q_{Melt\ Experimental} = \frac{391[kg.m^{-3}] \cdot L_{fusion} \cdot 0.15\ [m]}{432000[s]} = 45.4\ [W.m^{-2}] \tag{16}$$

The cumulative modelled melt between the two dates is $Q_{Melt\ Model} = 1114$ W.m^{-2}, or 9.3 W.m^{-2} on average per hour, meaning that the snow is melting. The difference between the experimental and the modelled results is of 95%. As the melt seems driven by the net radiation Q_{net}, a sensitivity test is carried out on the albedo parameter α. Increasing the albedo by 10% ($\alpha = 0.87$) results in a melt energy of $Q_{Melt\ Model}$ = -1702.6 W.m^{-2}, or -14.2 W.m^{-2} on average per hour, meaning that the snow would refreeze with such albedo. Increasing the albedo by 10% produces a result diverging from the initial one by 253%.

5.3 Evolution of snow properties

5.3.1 Correlation between Heat Flow & Bulk Density increase

The evolution of the vertical heat flow q within the snowpack is represented in Figure 8a. It is calculated with Equation 4 using the density measurements of each discrete layer in each snow pit. In some cases, ice lenses were present at the bottom of the snowpack, which made the density measurement impossible. Therefore, some heat flows have not been calculated until the bottom of the snowpack. In each pit, q is uniform. The heat flow increases over time. The evolution of the snow bulk density is shown in Figure 8b. It is calculated with Equation 3, using the snow density measurements. Each dot represents the value at one pit. The snow bulk density increases over time.



(a) Evolution of the heat flow through the snowpack.

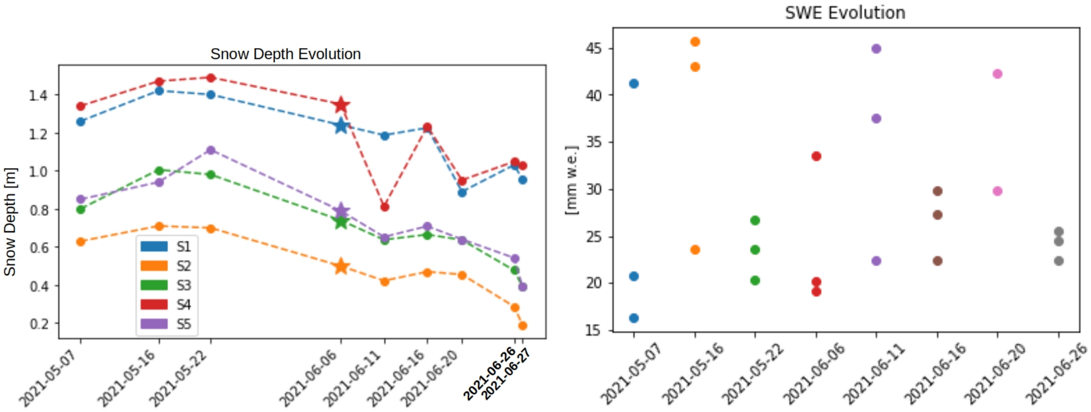
(b) Evolution of the snow bulk density.

Figure 8: Evolution of the heat flow and the bulk density of the snowpack. Each color represents each day. Three pits were dug each time.

5.3.2 Evolution of Bulk density and SWE, with respect to the snow depth data

Figure 9a shows the evolution of the snow depth from the stakes measurements, measured with the probe, as stated in Table 2 in Appendix A.2. The lack of data on the 6^{th} of June is due to an oversight. The symbols "$\star$" represent the interpolated data. The data point at stake 4 on the 11^{th} of June, and at Stake 1 and Stake 4 on the 20^{th} of June are unreasonably low. The observer must have hit an ice lens when measuring with the probe or has carried out the measurement in a "well" over the glacier surface. Glaciers topography can be highly irregular, especially on Foxfonna. Even at 650m elevation, glaciers can undergo strong melting and severe negative mass balance. 2020 was for example record-breaking in terms of glaciers negative mass balance in Ny-Ålesund (J.C.Gallet, personal communication). This induces some uncertainty in the measurement of about 5cm. The snow depth evolution comports two stages. From the 7^{th} to the 16^{th} of May, there was an accumulation of snow in the grid. The snow then started to melt until the end of the fieldwork. It must be noted that on the first time out in the field, the pits were dug close to S1, S2 and S3 to have a look at the snow stratigraphy. Ice lenses of 5cm were found at the bottom of each pit, most likely due to the rain event which must have occurred at the beginning of the winter season.

Figure 9b shows the SWE measurements of each day, from each snow pit. It is calculated using Equation 2, and the snow density data.



(a) Evolution of the snow depth at each stake. "⋆" : Interpolated data.

(b) Evolution of the SWE. It does not appear to decrease as the snow melts.

Figure 9: Comparison of the snow depth and the SWE. SWE is expected increase when the snow depth increases and vice-versa.

The weather data from Svalbard Airport (figure not shown) indicates that the last snowfall occurred in Longyearbyen on the 5th of May. Svalbard Airport (77°87'50, 20°97'52E, 5m elevation) is located about 11km away from the field site, almost at sea level. Therefore, the use of weather data from the airport must be considered with care, as the weather conditions vary greatly between the two sites. A snowfall event between the 7th the 16th of May at the airport could have been strengthened by the relief of a snowfall on Foxfonna at these dates.

Figure 10 shows the evolution of the measured broadband albedo of the snow, calculated according to Equation 6, measured with the pyranometer. Only measurements for when the sky was fully covered or cloud-free are considered. The albedo increased between the 7th and the 16th of May and was found to be quite high on the 6th of June as well. It decreased to a value of $\alpha \approx 0.77$ towards the end of the fieldwork.

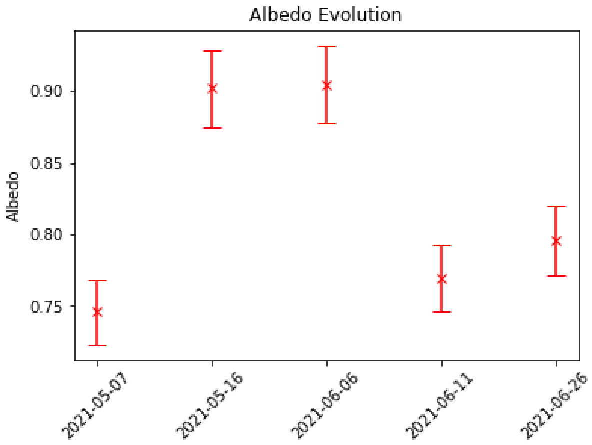


Figure 10: Albedo of the snow measured with a pyranometer including the 3% of linear error.

5.4 Superimposed ice measurement

Figure 11 shows the evolution of the ice depth measured at each stake with a probe, from the bottom of the ice to the top of the stake. The measurement had not been carried out on the 16^{th} of May because of an oversight. Superimposed ice was observed in the field from the 26^{th} of June. Therefore, the depth measurements should be constant, and drop from this day only. The variations in ice depth measurements of the order of less than 5cm can be attributed to the reading on skewed stakes. The sudden successive decrease and increase of the order of more than 10cm can be attributed to the fact that the observer must have hit an ice lens when performing the measurement, such as on the 11^{th} of June at Stake 4 and on the 20^{th} of June at Stake 1 and Stake 4. The snowpits dug on the 7^{th} and 16^{th} of May were located close to the stakes, to identify the possible presence of ice lenses. Ice lenses were indeed found close to each stake at the lowermost 5-10cm, at the very beginning of the fieldwork.

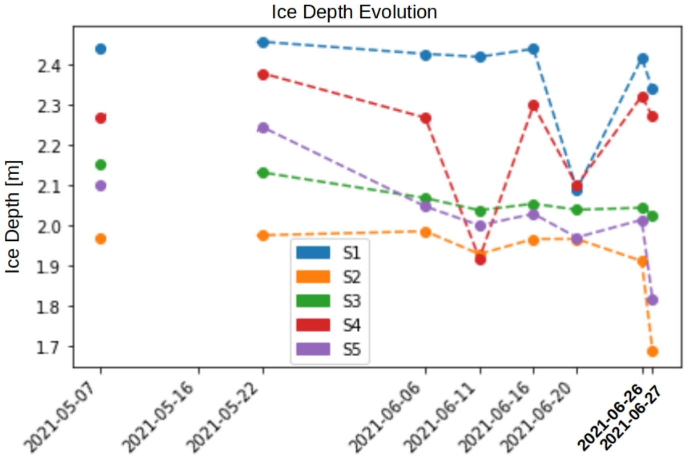
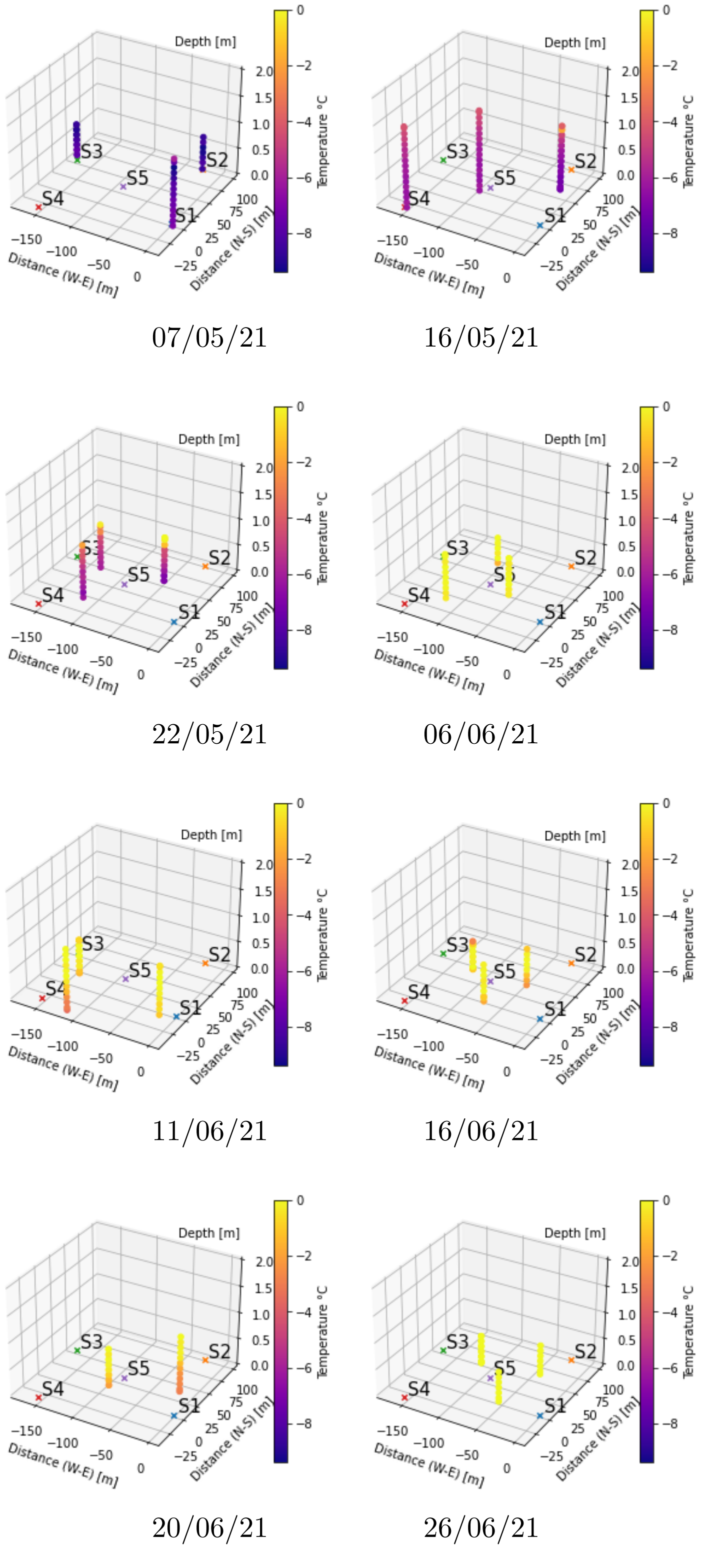


Figure 11: Evolution of the ice depth at each stake.

Figure 12 shows the temporal evolution of the snow pits temperature profiles, gathered from the snow pits data as shown in Figure 2[c] in Appendix A.2.



07/05/21

16/05/21

22/05/21

06/06/21

11/06/21

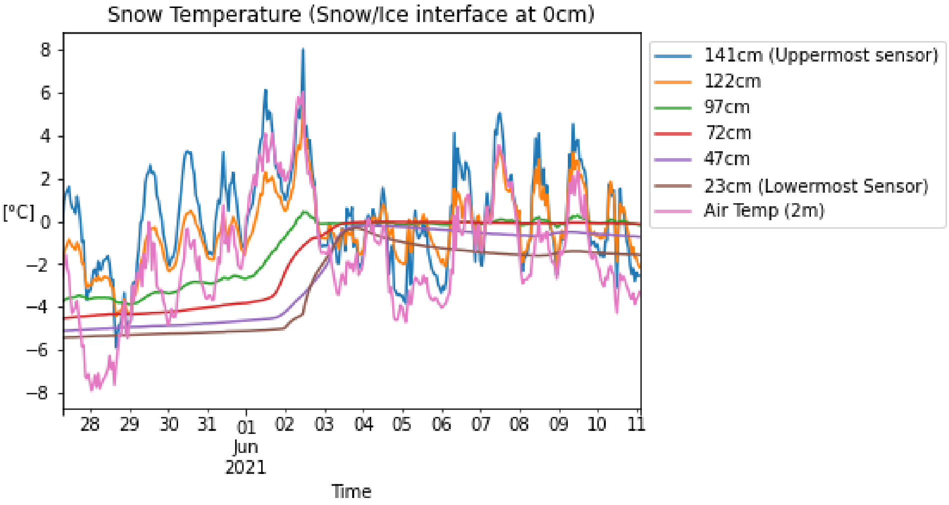
16/06/21

20/06/21

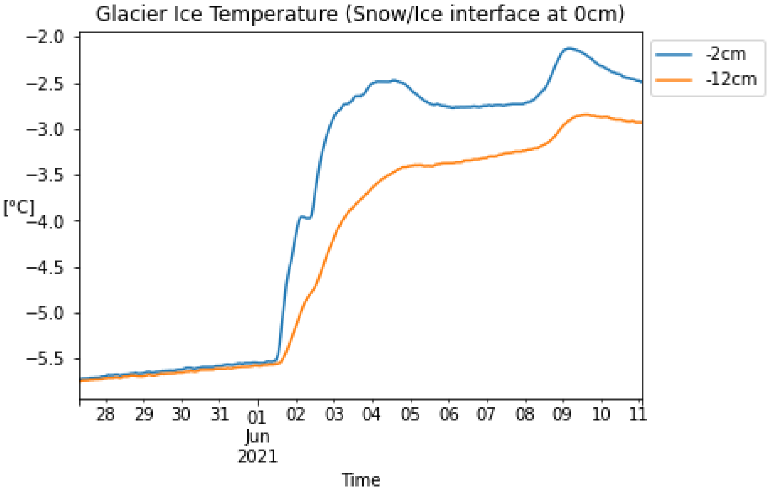
26/06/21

Figure 12: Temperature profiles of the snowpits. The crosses indicates the location of the respective stakes.

Figure 13 shows the snow and ice temperature data recorded by the thermistors string at the weather station. In Figure 13a, the sensors located more than 97cm above the ice were in contact with the air, as they indicate positive temperatures and fluctuate in a similar trend as the air temperature. The ones located below 97cm above the ice are measuring the snow temperature. The strong warming event on the 2^{nd} of June can be observed in the snow and ice data.



(a) Snow temperature



(b) Ice temperature

Figure 13: Thermistors string data. Ice surface is the depth reference.

6 Discussion

6.1 SEB - Melt model

The combination of the weather data Figure 6 with the SEB results presented in Figure 7 and the snow data exposes several interesting features.

Temperatures started to increase from the 29th of May, and peaked on the 2nd of June, as shown in Figure 6[a]. Figure 6[c] indicates a sudden increase of the wind speed on this day, which coincides with peaks of $Q_{sensible}$, Q_g, and Q_{melt} in Figure 7. The wind was coming from the south according to the wather station (see Figure 6[b]). In such condition, it typically advects warm air to Svalbard and increases the air humidity and the wind speeds. These results are coherent, as the sensible heat flux is a function of wind speed and temperature, and the greater air temperature induced the warming of the snow and the ice. Those factors, combined with the shortwave radiation-induced an increased melt. A negative latent heat flux indicates that water is evaporating from the snow surface to the atmosphere.

To compare, on the 30th of May, the air temperature was comprised between 0 °C and -4 °C, and the wind speed was only of about 2.5 m.s^{-1}. This results in a sensible heat flux and a latent heat flux close to zero. Therefore, the melt seems to be driven by the net heat radiation on this day.

On the 28th of May and on the 5th of June, the air temperature was negative, at -8°C and -4 °C respectively. Relatively strong wind speed were recorded, between 5 m.s^{-1} and 7.5 m.s^{-1}. On these days, the turbulent heat fluxes are modelled to be negative, which means it played in favor of the refreezing of the snow.

On the 7th of June, a strong positive peak of temperature has been recorded, with relatively strong wind speeds of about 7.8 m.s^{-1}. This results in positive turbulent heat fluxes, generating more melt on the glacier.

Figure 13a and Figure 13b shows that the warm event on the 2nd of June has had the effect of bringing the snowpack to its melting point, and bringing the glacier ice relatively close to it. This illustrates how crucial the role of snow is on glaciers, as it insulates the ice surface, and in a context where no accumulation occurs on a glacier, snow can at best delay the melting of the ice.

About the modelling of the snow melt between the 6th and the 11th of June, the modelled amount of melt energy (45.4 [W.m^2]) and the one calculated experimentally (1114 [W.m^2]) using Equation 15 differ by 95%. This great divergence indicates that the model does not describe the melt which occurred on the glacier successfully within this period. In spite of differing strongly from the observed melt, the model still predicts that melting should occur on the glacier, and not refreezing. Therefore, the model can reproduce the main physical processes on the system. The use of local shortwave radiation data and wind data would certainly have made the results closer to the observed melt. According to Figure 7, the melt seems driven by the net radiation Q_{net}. Therefore, a sensitivity test was carried out for the albedo parameter. It revealed that increasing it by 10% ($\alpha = 0.87$, the higher the albedo, the greater the surface reflectivity) decreases the melt energy by

253%. This shows the importance of the albedo parameter in modelling snow melt.

6.2 Evolution of snow properties

6.2.1 Correlation between Heat Flow & Bulk Density increase

There is a correlation between the increase of the vertical heat flow shown in Figure 8a, and the increase of the bulk density over time observed in Figure 8b. The bulk density increases as the snow melts, liberating the water in the snowpack and densifying it. As the snow is denser, it conducts heat more efficiently. It must be noted that density measurements in the snow can be complicated by the formation of ice lenses in the snowpack. It sometimes prevents the collection of snow density data [7].

6.2.2 Evolution of Bulk density, SWE, with respect to snow depth data

The snow depth data in Figure 9a suggests an accumulation of snow between the 7^{th} and the 16^{th} of May, most likely by a strong wind event that remobilized snow in the grid. This statement is comforted by the increase of SWE, and the bulk density remaining similarly high compared to the preceding measurement, as well as Figure 10. On the 16^{th} of May, the albedo is close to 1, and fresh snow, even deposited by the wind has the highest albedo. The hypothesis of a snowfall event is not supported by the precipitation data at Svalbard Airport (figure not shown), which indicates that the last snowfall occurred on the 5^{th} of May in Longyearbyen. To conclude, the increase of SWE, the fact that the bulk density remained unchanged, and the high albedo comfort the hypothesis that snow was remobilized in the grid by strong winds.

Combining Figure 8b and Figure 9a, allows us to highlight the correlation between the snow melt and the bulk density increase. The more the melt, the more water is released in the snowpack, the denser the snowpack.

The SWE measurements do not expose any net increase or decrease in the amount of water contained in the snowpack. As the snow melts, as shown in Figure 9a, the SWE is expected to decrease over time. The grid is located in a flat area, surrounded by a bowl, as seen in Figure 1. It is therefore subjected to a lot of advection of meltwater from the above surroundings. This, combined with the continuous melting might explain why the SWE varies so much. The grid's location seems to prevent any further analysis of the SWE data.

About the albedo evolution, the unexpectedly high value recorded on the 6^{th} of June does not seem to be attributed to the presence of freshly fallen snow grains, as the snow depth data in Figure 9a indicates a net melt over this period of time. It could have been attributable to windblown snow grains, which also have a higher albedo, but this can only be an assumption as the state of the snow surface of this day has not been investigated with care. At the end of the fieldwork, the albedo reached $\alpha = 0.77$ which is a typical value for ageing snow.

6.3 Isothermal snow and Superimposed ice

Figure 11 and in-situ observation of the ice surface indicate that the formation of superimposed ice took place from the 26th of June, and was more important at S2 ($\Delta h_{ice} \approx 22.2cm$) and S5 ($\Delta h_{ice} \approx 19.7cm$). The warm event that occurred on the 2nd of June made the snowpack isothermal, as observed on the temperature profiles of the snow pits from the 6th of June in Figure 12, and on Figure 13, which shows the snow temperature recorded by the thermistors strings. On the 22nd of May, the surface had not reached 0°C yet, so no melting had occurred yet. According to Figure 9a the snow melt started after that date. The sudden warm event which rendered the entire snowpack isothermal had drastic consequences on the formation of superimposed ice, as its formation requires first a strong melt at the surface to allow the water to percolate at the bottom of the snowpack, and refreeze if snow allows that. When the snow is isothermal, the water flows along the glacier surface. SI formation is therefore prevented. The use of a thermistors string to measure the snow and ice temperature exposes some drawbacks. The string can be a pathway for percolating meltwater. Moreover, the thermistors located close to the surface are more susceptible to be heated by radiations.

The use of weather stations located away from the field site exposes some drawbacks as the topography is different. Indeed, both the wind and the radiation measurements are strongly topography dependant. The wind measurements collected in Breinosa are not truly representative of the wind conditions experienced in the grid. Also, the amount of radiation reaching the radiation sensors in Adventdalen is dependant on the topography. Moreover, the snow disappears earlier in the valley than on the glacier, making the longwave radiation measurements not representative of the field conditions.

7 Conclusion

The weather data indicates that a strong warm event occurred on the 2^{nd} of June, which induced a positive peak in the sensible and ground heat flux, which largely contributed to the resulting peak in melt energy supplied to the snow. The modelled amount of melt energy supplied to the snow between the 6^{th} and the 11^{th} of June is 45.4 W.m^{-2}, and the amount of melt energy calculated from the observed melt during this period is 9.3 W.m^{-2}. The two results differ by 95%. In spite of not being able to reproduce the observed melt accurately, the model at least predicts that melt should occur on the glacier. The divergence between the observed and the modelled melt can be attributed to the use of weather data from weather stations located far away from the field site for the longwave radiation and the wind data. A sensitivity test carried out on the albedo parameter highlights how crucial it is for modelling snowmelt accurately, as increasing the albedo by 10% leads to a result 283% smaller than the initial one, meaning that a lot of refreezing would occur.

The frequent in-situ snow measurements permitted the depiction of the evolution of several properties of the snow as it melts. As the snowmelts, the water coalesces within it, the snowpack bulk density increases, which results in a greater heat flow through the snowpack. The comparison of the evolution of the SWE and the bulk density, as well as the albedo increase seem to indicate that wind deposited snow on the site between the 7^{th} and the 16^{th} of May. Unfortunately, the SWE data does not expose any clear trend after the 16^{th} of May. Perhaps the interplay between the advection of melt water from the upper part of the glacier and the progressive melt of the snowpack prevents a clear understanding of the evolution of the SWE of the snowpack. The grid's location in a bowl of the glacier seems to prevent further analysis of the SWE data.

The albedo evolution, and the correlation between snow bulk density and the SWE, are in accordance with the snow depth measurements, which indicate a snow accumulation between the 7^{th} and the 16^{th} of May, most likely due to remobilized snow by wind. The unexpectedly high albedo value measured on the 16^{th} of June is supposely attributed to the formation of a windblown snow surface. Towards the end of the fieldwork, the snow albedo decreased to α=0.77, which is a characteristic value for ageing snow.

The warm event which occurred on the 2^{nd} of June rendered the snowpack isothermal and brought the ice surface very close to its melting point abruptly. This prevented the formation of SI as it requires a strong surface melting feeding the bottom of the snowpack with meltwater. The amount of SI formed, and the event of its formation is not systematic every year. It depends on the weather conditions during the melt. The measurement of the sudden ice temperature increase illustrates the importance of the snow cover on glaciers, and how its absence can modify the thermal regime of the glacier.

The work carried out in this report could be followed by further studies on the impact of refreezing on Svalbard glaciers that are still firn-covered. Ultimately, an inter-comparison of the refreezing impacts on Svalbard glaciers and Greenland glaciers could be carried out in order to estimate the amount of meltwater contributing to sea level rise from Greenland and Svalbard, nowadays and in a few decades.

References

[1] J. L. Wadham and A.-M. Nuttall, "Multiphase formation of superimposed ice during a mass-balance year at a maritime high-arctic glacier," *Journal of Glaciology*, vol. 48, no. 163, pp. 545–551, 2002. DOI: 10.3189/172756502781831025.

[2] J. Harper, N. Humphrey, W. T. Pfeffer, J. Brown, and X. Fettweis, "Greenland ice-sheet contribution to sea-level rise buffered by meltwater storage in firn," *Nature*, vol. 491, no. 7423, pp. 240–243, 2012.

[3] B. Noël *et al.*, "Low elevation of svalbard glaciers drives high mass loss variability," *Nature communications*, vol. 11, no. 1, pp. 1–8, 2020.

[4] I. Esau, S. Argentini, R. Przybylak, I. Repina, and A. Sjöblom, *Svalbard meteorology*, 2012.

[5] E. J. Førland, R. Benestad, I. Hanssen-Bauer, J. E. Haugen, and T. E. Skaugen, "Temperature and precipitation development at svalbard 1900–2100," *Advances in Meteorology*, vol. 2011, 2011.

[6] N. Snow and I. D. Center, *Facts about glaciers*, https://nsidc.org/cryosphere/glaciers/quickfacts.html.

[7] K. M. Cuffey and W. S. B. Paterson, *The physics of glaciers*. Academic Press, 2010.

[8] J. Malecki, "Elevation and volume changes of seven dickson land glaciers, svalbard, 1960–1990–2009," *Polar Research*, vol. 32, no. 1, p. 18 400, 2013.

[9] W. J. van Pelt, T. V. Schuler, V. A. Pohjola, and R. Pettersson, "Accelerating future mass loss of svalbard glaciers from a multi-model ensemble," *Journal of Glaciology*, vol. 67, no. 263, pp. 485–499, 2021.

[10] W. J. van Pelt, V. A. Pohjola, and C. H. Reijmer, "The changing impact of snow conditions and refreezing on the mass balance of an idealized svalbard glacier," *Frontiers in Earth Science*, vol. 4, p. 102, 2016.

[11] A. Wright, J. Wadham, M. Siegert, A. Luckman, J. Kohler, and A. Nuttall, "Modeling the refreezing of meltwater as superimposed ice on a high arctic glacier: A comparison of approaches," *Journal of Geophysical Research: Earth Surface*, vol. 112, no. F4, 2007.

[12] Wikipedia. "File:taku glacier firn ice sampling.png." (2005), [Online]. Available: https://commons.wikimedia.org/wiki/File:Taku_glacier_firn_ice_sampling.png.

[13] K. Keegan, "Snow metamorphism and firn compaction," in Jan. 2021, ISBN: 9780124095489. DOI: 10.1016/B978-0-12-818234-5.00041-9.

[14] R. Sommerfeld and E. LaChapelle, "The classification of snow metamorphism," *Journal of Glaciology*, vol. 9, no. 55, pp. 3–18, 1970.

[15] S. Colbeck, "Thermodynamics of snow metamorphism due to variations in curvature," *Journal of Glaciology*, vol. 26, no. 94, pp. 291–301, 1980.

[16] Z. Yosida *et al.*, "Physical studies on deposited snow. .; thermal properties.," *Contributions from the Institute of Low Temperature Science*, vol. 7, pp. 19–74, 1955.

[17] F. Domine, S. Morin, E. Brun, M. Lafaysse, and C. M. Carmagnola, “Seasonal evolution of snow permeability under equi-temperature and temperature-gradient conditions,” *The Cryosphere*, vol. 7, no. 6, pp. 1915–1929, 2013. DOI: `10.5194/tc-7-1915-2013`. [Online]. Available: `https://tc.copernicus.org/articles/7/1915/2013/`.

[18] S. Colbeck, “Theory of metamorphism of dry snow,” *Journal of Geophysical Research: Oceans*, vol. 88, no. C9, pp. 5475–5482, 1983.

[19] ——, *Theory of Metamorphism of Wet Snow*, ser. CRREL report. U.S. Army Cold Regions Research and Engineering Laboratory, 1973. [Online]. Available: `https://books.google.com/books?id=JFgoOgAACAAJ`.

[20] R. Hock, “Glacier melt: A review of processes and their modelling,” *Progress in Physical Geography: Earth and Environment*, vol. 29, no. 3, pp. 362–391, 2005. DOI: `10.1191/0309133305pp453ra`. eprint: `https://doi.org/10.1191/0309133305pp453ra`. [Online]. Available: `https://doi.org/10.1191/0309133305pp453ra`.

[21] M. Sturm, B. Taras, G. E. Liston, C. Derksen, T. Jonas, and J. Lea, “Estimating snow water equivalent using snow depth data and climate classes,” *Journal of Hydrometeorology*, vol. 11, no. 6, pp. 1380–1394, 2010.

[22] J. L. Mccreight and E. Small, “Modeling bulk density and snow water equivalent using daily snow depth observations,” *The Cryosphere*, vol. 8, no. 2, pp. 521–536, 2014.

[23] J. Kirkham *et al.*, “Near real-time measurement of snow water equivalent in the nepal himalayas,” Jul. 2019.

[24] A. Judson and N. Doesken, “Density of freshly fallen snow in the central rocky mountains,” *Bulletin of the American Meteorological Society*, vol. 81, no. 7, pp. 1577–1588, 2000. DOI: `10.1175/1520-0477(2000)081<1577:DOFFSI>2.3.CO;2`. [Online]. Available: `https://journals.ametsoc.org/view/journals/bams/81/7/1520-0477_2000_081_1577_doffsi_2_3_co_2.xml`.

[25] Y.-C. Yen, *Review of thermal properties of snow, ice, and sea ice*, 10. US Army, Corps of Engineers, Cold Regions Research and Engineering Laboratory, 1981, vol. 81.

[26] D. Perovich, T. Grenfell, B. Light, and P. Hobbs, “Seasonal evolution of the albedo of multiyear arctic sea ice,” *Journal of Geophysical Research: Oceans*, vol. 107, no. C10, SHE–20, 2002.

[27] D. Hartmann, *Global physical climatology: Second Edition.* Jan. 2015, pp. 1–485.

[28] F. W. Murray, “On the computation of saturation vapor pressure.,” Rand Corp Santa Monica Calif, Tech. Rep., 1966.

[29] A. Melinder, “Thermophysical properties of aqueous solutions used as secondary working fluids,” 2007.

A Appendix

A.1 Introduction : Map

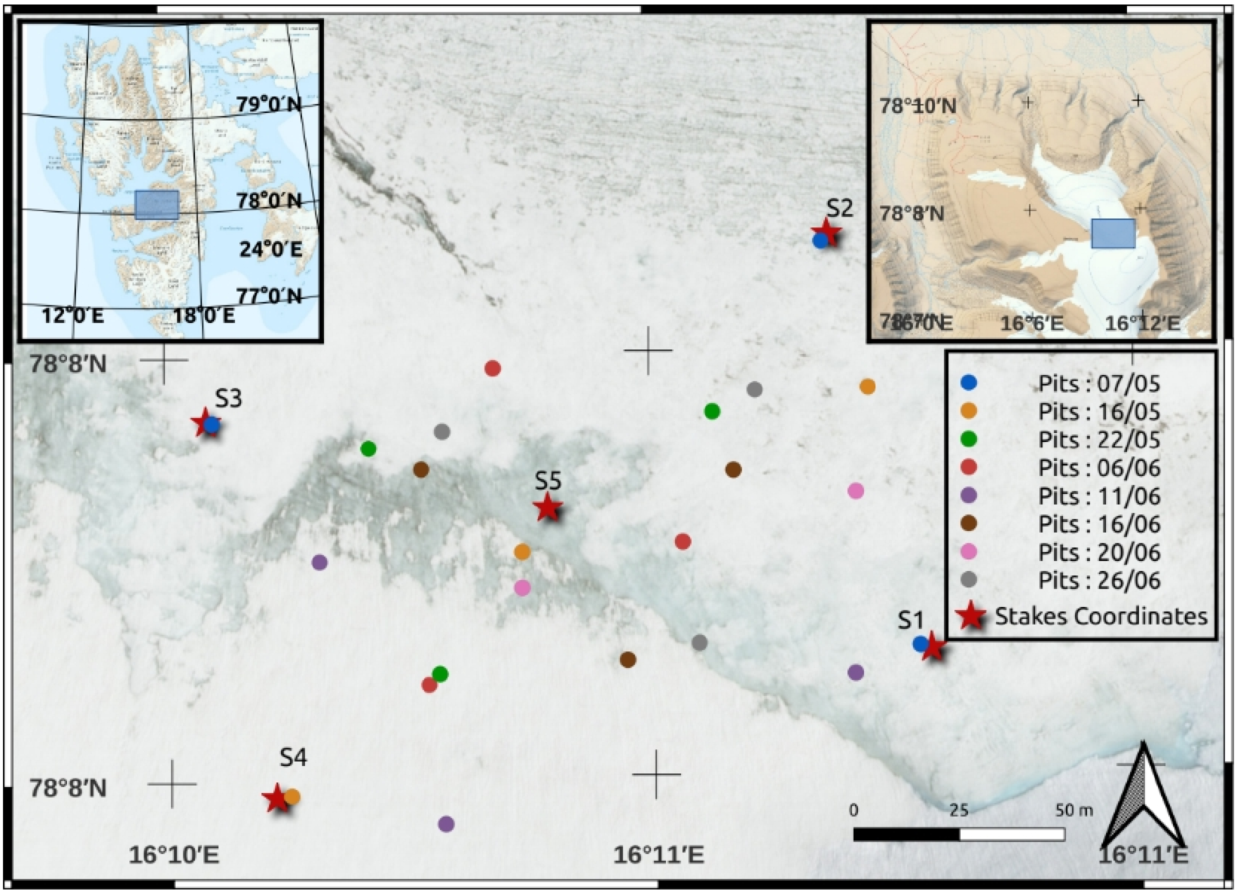


Figure 1: Location of the fieldwork site. Foxfonna glacier is located 10km East of Longyearbyen. The site is located in a flat bassin of the glacier, under the 650m altitude line. The grid has the dimensions [160m E-W ×100m N-S]. The local weather station is located by S4.

A.2 Method : Field procedures and tools summary

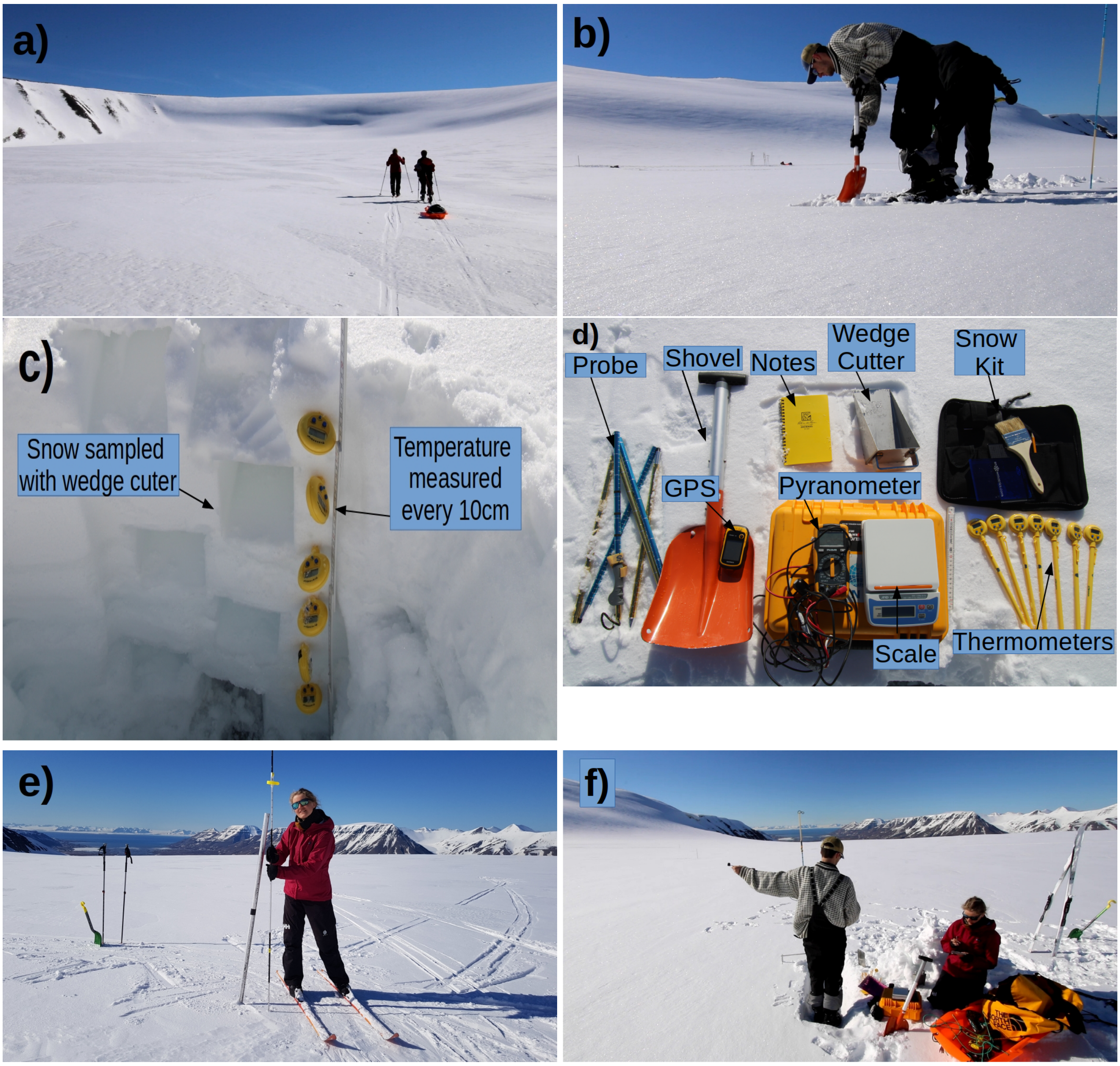


Figure 2: Fieldwork routine. a) Going to the site. b) Digging a snow pit. c) Temperature and density profile. d) Snow pits tools. e) Stake depth measurement. f) Albedo measurement.

Table 2 presents a summary of all the measurements carried out in the field, as well as the tools used.

Snow Measurements	**Tools and Uncertainty**	**Details**
Ice Thickeness	Probe (± 5cm, Stakes not Straight)	From ice to marker on the stake
Snow Thickeness		At each stake and each pit
Albedo	Pyranometer SP-230 APOGEE INSTRUMENTS, Range : 360 to 1120 nm Linear error : 3%	Average of 5 measurements
Snow Temperature	Lollipop Traceable Thermometer ($\pm 1C$) + Thermistors String RST (±0.1°C)	Temperature (Lollipop) and density measurement every 10 cm from the top to the bottom of the pit Six sensors on the string from 23cm to 141cm above ice surface.
Snow Density	Scale HT-3000 AND (±1g) + Wedge Cutter 1000cc Snowmetrics	
Pits coordinates	GPS	
Glacier Ice Temperature	Thermistors String RST (±0.1°C)	Two sensors below the glacier ice at -2cm and -12 cm.
Weather Measurements	**Tools and Uncertainty**	**Details**
Shortwave Radiation	Pyranometer SKS 1110 Skye Instrument 0.2 % error	Weather Station – Foxfonna
Air Temperature (2m)	HMP60-L Campbell Scientific ± 0.6°C	
Relative Humidity	HMP60-L Campbell Scientific (± 5%)	
Longwave Radiation	Campbell Scientific CNR1(± 10%)	Weather Station - Adventhalen
Wind	Campbell Scientific 05103-L ± 0.3 m/s	Weather Station - Breinosa

Table 2: Summary of the measurements made on the field.

A.3 Code : Melt Model & Plot of Weather Data

```
######### This code reads and plots the weather data ###########"

import numpy as np
from matplotlib import pyplot as plt
import pandas as pd
import sys
# sys.path is a list of absolute path strings

sys.path.append(’/home/thibault/Documents/UNIVERSITY/UNIS/Thesis/Python/
   ↪ Thermistor_String’)
import Thermistors



#%% DATES

# ==========================================
# Winter_Fox 2020-10-21 / 2021-05-06
# May_Fox 2021-05-27 / 2021-06-11
# Ad5 2020-02-03 / 2021-08-17
# Advh 2020-01-01 / 2021-08-17
#
# ================================================
#%% INITIAL DataFrame NAMES + READ DataFrames

colnamesJans_init = [’TIMESTAMP’, ’RECORD’, ’ID’, ’LT10cm_minutt_Max’, ’
   ↪ LT10cm_minutt_Min’,
      ’LT10cm_minutt_Avg’, ’LT1m_minutt_Max’, ’LT1m_minutt_Min’,
      ’LT1m_minutt_Avg’, ’LT3m_minutt_Max’, ’LT3m_minutt_Min’,
      ’LT3m_minutt_Avg’, ’LF_minutt_Max’, ’LF_minutt_Avg’, ’AT_mbar’,
      ’VH_mps_Max’, ’VH_10_minutt’, ’VH_mps_mid’, ’VR_gr_framh’, ’NB_time
         ↪ ’,
      ’SD_m’, ’SD_kval’, ’Batt_V_Min’]

colnamesAd5_init = [’TIMESTAMP’, ’RECORD’, ’CM3_opp_Wpm2_Avg’, ’
   ↪ CG3_opp_Wpm2_Avg’,
      ’CM3_ned_Wpm2_Avg’, ’CG3_ned_Wpm2_Avg’, ’CNR1_temp_gr_C_Avg’]

colnamesAdvH_init = [’TIMESTAMP’, ’RECORD’, ’ID’, ’LT1_gr_C_Max’, ’
   ↪ LT1_gr_C_Min’,
      ’LT1_gr_C_Avg’, ’LT2_gr_C_Max’, ’LT2_gr_C_Min’, ’LT2_gr_C_Avg’,
      ’LT3_gr_C_Max’, ’LT3_gr_C_Min’, ’LT3_gr_C_Avg’, ’LF1_prst_Max’,
      ’LF1_prst_Avg’, ’LT4_gr_C_Max’, ’LT4_gr_C_Min’, ’LT4_gr_C_Avg’,
      ’LF2_prst_Max’, ’LF2_prst_Avg’, ’AT_mbar’, ’VH1_mps_Max’, ’
         ↪ VH1_10_min’,
      ’VH1_mps_mid’, ’VR1_gr_framh’, ’VH2_sek_Max’, ’VH2_10_min’,
      ’VH2_mps_mid’, ’VR2_gr_framh’, ’Batt_V_Min’]
```

```
Jans = pd.read_csv(’Jans.csv’, parse_dates=["TIMESTAMP"])
Ad5 = pd.read_csv(’Adv_5min.csv’, parse_dates=["TIMESTAMP"])
Bull = pd.read_csv(’Adv_5min_BULLSHIT.csv’)
Ad5_AprilMay = pd.read_csv(’Adv_5minAprilMay.csv’, parse_dates=["
   ↪ TIMESTAMP"])
AdvH = pd.read_csv(’Adv_Hour.csv’, parse_dates=["TIMESTAMP"])
Brei = pd.read_csv(’Breinosa_Res_Data.csv’, parse_dates=["TIMESTAMP"])

#print(Jans[’LF_minutt_Avg’])
MayFox = Thermistors.May
YearFox = Thermistors.Winter
#print(Jans[’LF_minutt_Avg’].head())



Jans = Jans.drop(columns=[’RECORD’,’ID’, ’LT10cm_minutt_Max’, ’
   ↪ LT10cm_minutt_Min’,
      ’LT1m_minutt_Max’, ’LT1m_minutt_Min’,
      ’LT3m_minutt_Max’, ’LT3m_minutt_Min’,
      ’LF_minutt_Max’, ’AT_mbar’,
      ’VH_mps_Max’, ’VH_10_minutt’, ’NB_time’,
      ’SD_m’, ’SD_kval’, ’Batt_V_Min’])

Advh = AdvH.drop(columns=[’RECORD’, ’ID’, ’LT1_gr_C_Max’, ’LT1_gr_C_Min’,
    ’LT2_gr_C_Max’, ’LT2_gr_C_Min’,
      ’LT3_gr_C_Max’, ’LT3_gr_C_Min’, ’LF1_prst_Max’,
      ’LT4_gr_C_Max’, ’LT4_gr_C_Min’,
      ’LF2_prst_Max’, ’VH1_mps_Max’, ’VH1_10_min’,
      ’VH2_sek_Max’, ’VH2_10_min’,
      ’Batt_V_Min’])

Ad5 = Ad5.drop(columns=["CNR1_temp_gr_C_Avg"])

Brei = Brei.drop(columns=[’RECORD’, ’ID’,’Batt_Volt_Min’])


J_Time = Jans["TIMESTAMP"]
J_T10 = Jans[’LT10cm_minutt_Avg’]
J_T1 = Jans[’LT1m_minutt_Avg’]
J_T3 = Jans[’LT3m_minutt_Avg’]
J_Hum = Jans[’LF_minutt_Avg’]
J_ws = Jans[’VH_mps_mid’]
J_wd = Jans[’VR_gr_framh’]

Advh_Time = Advh["TIMESTAMP"]
```

```
Advh_T1 = Advh[’LT1_gr_C_Avg’]
Advh_T2 = Advh[’LT2_gr_C_Avg’]
Advh_T3 = Advh[’LT3_gr_C_Avg’]
Advh_T4 = Advh[’LT4_gr_C_Avg’]
Advh_Hum1 = Advh[’LF1_prst_Avg’]
Advh_Hum2 = Advh[’LF2_prst_Avg’]
Advh_wd = Advh[’VH1_mps_mid’]
Advh_wd = Advh[’VR1_gr_framh’]
Advh_wd = Advh[’VH2_mps_mid’]
Advh_wd = Advh[’VR2_gr_framh’]
BreiRHmax = Brei[’RH_Max’]



#%% Set Index + Hour Data Avrage
Jans = Jans.set_index(’TIMESTAMP’)
Brei = Brei.set_index(’TIMESTAMP’)
Ad5 = Ad5.set_index(’TIMESTAMP’)
Advh = Advh.set_index(’TIMESTAMP’)
Ad5_AprilMay = Ad5_AprilMay.set_index(’TIMESTAMP’)

Jans = Jans.resample("H").mean()
Brei = Brei.resample("H").mean()
Advh = Advh.resample("H").mean()
Ad5 = Ad5.resample("H").mean()
Ad5_AprilMay = Ad5_AprilMay.resample("H").mean()


#%% Plot AD5 Data
# ===========================================
# plt.plot(Ad5[’S_Up’])
# plt.plot(Ad5[’L_Up’])
# plt.plot(Ad5[’S_Down’]/Ad5[’S_Up’])
# plt.plot(Ad5[’S_Up’])
# plt.plot(Ad5[’S_Down’])
#
# Ad5.plot()
# ================================
#%% Constants

rho_w = 997 #Kg/m3
rho_air = 1.2 # Kg/m3
Cw = 4200 #J/Kg.K Heat Cap. Water
C_air = 1012 #cP Jg1K1 specific heat capacity of air
L_vaporisation = 2838 #J/kg
C_Sensible = 0.0019 #W/(m2 C)
C_Latent = 0.0015 #W/(m2 C) #p.157 Book Glaciers
L_fusion = 334000 #J/Kg M3
```

```
A= 0.0019 # (Cuffey and Paterson, 2010) an empirical transfer
   ↪ coefficient including surface roughness and atmospheric stability,
B = 0.0015
RH_sfc = 100
T = 20
T_sfc_26Jul = 0
T_sfc_06May = -8.7
Boltz_cst = 5.67*(10**-8)
T_sfc_06May = -8.7



#%% 21/10/20 to 06/05/21 YEAR Thermistors Data

start_date_YearFox = YearFox.index[0]

end_date_YearFox = YearFox.index[-1]


after_start_date_Ad5 = Ad5.index >= start_date_YearFox
before_end_date_Ad5 = Ad5.index <= end_date_YearFox
between_two_dates__Ad5 = after_start_date_Ad5 & before_end_date_Ad5
Ad5_YearFox = Ad5.loc[between_two_dates__Ad5]


after_start_date_Brei = Brei.index >= start_date_YearFox
before_end_date_Brei = Brei.index <= end_date_YearFox
between_two_dates_Brei = after_start_date_Brei & before_end_date_Brei
Brei_YearFox = Brei.loc[between_two_dates_Brei]


T_sfc = Thermistors.Winter_40_48["T(42)"]

q_air_YearFox = (YearFox["RH"]/100)*611.12*np.exp((22.2*YearFox["Air␣Temp
   ↪ "])/(272.62 + YearFox["Air␣Temp"]))
q_sfc_YearFox = (RH_sfc/100)*611.12*np.exp((22.2*T_sfc)/272.62 + T_sfc)

#Albedo should be changed as well as T_surface
ALBEDO = 0.8

Q_net_YearFox = YearFox["Inc␣Radn"] * (1 - ALBEDO) + Ad5_YearFox["L_Down"
   ↪ ] - Ad5_YearFox["L_Up"] ### ALBEDO TO IMPROVE ###
#Q_net_YearFox_Boltz = YearFox["Inc Radn"] * (1 - ALBEDO) + Ad5_YearFox
   ↪ ["L_Down"] - Boltz_cst*((T_sfc+273)**4)
Q_sens_YearFox = rho_air * C_air * C_Sensible * Brei_YearFox["WS_ms_Avg"]
   ↪  * ( YearFox["Air␣Temp"] - T_sfc ) #Cp = sp. heat. cap . air #
   ↪ ADJUST BY A STABILITY FACTOR (C_sensible) ### T_sfc CAN BE
   ↪ IMPROVED ###
```

```
Q_lat_YearFox = 22.2 * C_Latent * Brei_YearFox["WS_ms_Avg"] * (
   ↪ q_air_YearFox - q_sfc_YearFox) # Or Humidity difference # 22.2
   ↪ from p.157 Book Glaciers ### T_sfc CAN BE IMPROVED ###


Q_net_YearFox = pd.DataFrame(Q_net_YearFox)
Q_net_YearFox.columns = ["Q_net_YearFox"]

# ===============================================================
# Q_net_YearFox_Boltz = pd.DataFrame(Q_net_YearFox_Boltz)
# Q_net_YearFox_Boltz.columns = ["Q_net_YearFox_Boltz"]
# ============================================================


Q_sens_YearFox = pd.DataFrame(Q_sens_YearFox)
Q_sens_YearFox.columns = ["Q_sens_YearFox"]

Q_lat_YearFox = pd.DataFrame(Q_lat_YearFox)
Q_lat_YearFox.columns = ["Q_lat_YearFox"]

Q_g_YearFox = pd.DataFrame(Thermistors.df_B_year)
Q_g_YearFox.columns = ["Q_g_YearFox"]


plt.figure()
ax = Q_net_YearFox.plot(legend=True)
# ==============================================
# Q_net_YearFox_Boltz.plot(legend=True)
# =============================================
Q_sens_YearFox.plot(ax=ax, legend=True)
Q_lat_YearFox.plot(ax=ax, legend=True)
Q_g_YearFox.plot(ax=ax, legend=True)
plt.title("SEB_Year")
plt.ylabel("W/m2")
plt.xlabel("Time")
plt.ylim(-300,300)

#%% 2021-05-27 to 2021-06-11 MAY Thermistors Data

start_date_MayFox = MayFox.index[0]

end_date_MayFox = MayFox.index[-1]


after_start_date_Ad5_May = Ad5.index >= start_date_MayFox
before_end_date_Ad5_May = Ad5.index <= end_date_MayFox
between_two_dates_Ad5_May = after_start_date_Ad5_May &
   ↪ before_end_date_Ad5_May
```

```
Ad5_MayFox = Ad5.loc[between_two_dates_Ad5_May]


after_start_date_Brei_May = Brei.index >= start_date_MayFox
before_end_date_Brei_May = Brei.index <= end_date_MayFox
between_two_dates_Brei_May = after_start_date_Brei_May &
   ↪ before_end_date_Brei_May
Brei_MayFox = Brei.loc[between_two_dates_Brei_May]




# ===================================================
# for row in Thermistors.May_40_48.iterrows() :
# Tsfc = pd.DataFrame(Thermistors.May_40_48[Thermistors.May_40_48[row] <
   ↪  Thermistors.May_40_48["Air Temp"]])
# =====================================================

#
   ↪ ================================================================================
   ↪
# Tsfc40 = pd.DataFrame(Thermistors.May_40_48[Thermistors.May_40_48["T
   ↪ (40)"] < Thermistors.May_40_48["Air Temp"] ])
# Tsfc41 = pd.DataFrame(Thermistors.May_40_48[Thermistors.May_40_48["T
   ↪ (41)"] < Thermistors.May_40_48["Air Temp"] ])
# Tsfc42 = pd.DataFrame(Thermistors.May_40_48[Thermistors.May_40_48["T
   ↪ (42)"] < Thermistors.May_40_48["Air Temp"] ])
# Tsfc43 = pd.DataFrame(Thermistors.May_40_48[Thermistors.May_40_48["T
   ↪ (43)"] < Thermistors.May_40_48["Air Temp"] ])
# Tsfc44 = pd.DataFrame(Thermistors.May_40_48[Thermistors.May_40_48["T
   ↪ (44)"] < Thermistors.May_40_48["Air Temp"] ])
# Tsfc45 = pd.DataFrame(Thermistors.May_40_48[Thermistors.May_40_48["T
   ↪ (45)"] < Thermistors.May_40_48["Air Temp"] ])
# Tsfc46= pd.DataFrame(Thermistors.May_40_48[Thermistors.May_40_48["T
   ↪ (46)"] < Thermistors.May_40_48["Air Temp"] ])
# Tsfc47= pd.DataFrame(Thermistors.May_40_48[Thermistors.May_40_48["T
   ↪ (47)"] < Thermistors.May_40_48["Air Temp"] ])
# Tsfc48= pd.DataFrame(Thermistors.May_40_48[Thermistors.May_40_48["T
   ↪ (48)"] < Thermistors.May_40_48["Air Temp"] ])
#
   ↪ ================================================================================
   ↪
Tsfc_beg = Thermistors.May_40_48["T(41)"][(Thermistors.May_40_48["T(41)"
   ↪ ].index > '2021-05-27') & (Thermistors.May_40_48["T(41)"].index <=
   ↪ '2021-06-01')]
Tsfc_end = Thermistors.May_40_48["T(42)"][(Thermistors.May_40_48["T(42)"
   ↪ ].index > '2021-06-01') & (Thermistors.May_40_48["T(42)"].index <=
   ↪ '2021-06-11')]
T_sfc_MayFox = Tsfc_beg
T_sfc_MayFox = T_sfc_MayFox.append(Tsfc_end)

ALbedo = pd.DataFrame(T_sfc_MayFox)
ALbedo.drop(columns=ALbedo.columns[0],
       axis=1,
       inplace=True)
ALbedo["0.9"] = 0.9
ALbedo["0.75"] = 0.75

ALbedo_beg = ALbedo["0.9"][(ALbedo["0.9"].index > '2021-05-27') & (ALbedo
    ↪ ["0.9"].index <= '2021-06-01')]
ALbedo_end = ALbedo["0.75"][(ALbedo["0.75"].index > '2021-06-01') & (
    ↪ ALbedo["0.75"].index <= '2021-06-11')]

albedo = ALbedo_beg
albedo = albedo.append(ALbedo_end)


q_air_MayFox = (MayFox["RH"]/100)*611.12*np.exp((21.875*MayFox["Air␣Temp"
    ↪ ])/(265.5 + MayFox["Air␣Temp"]))
q_sfc_MayFox = (RH_sfc/100)*611.12*np.exp((22.2*T_sfc_MayFox)/(272.62 +
    ↪ T_sfc_MayFox))




Q_net_MayFox = MayFox["Inc␣Radn"] * (1 - albedo) + Ad5_MayFox["L_Down"] -
    ↪  Ad5_MayFox["L_Up"]
Q_net_MayFox_Boltz = MayFox["Inc␣Radn"] * (1 - albedo) + Ad5_MayFox["
    ↪ L_Down"] - Boltz_cst*(( T_sfc_MayFox+273)**4) # Albedo 16 May




Lup_Boltz = pd.DataFrame(Boltz_cst*(( T_sfc_MayFox+273)**4))

Lup_Boltz_LEFT = pd.DataFrame(Boltz_cst*(( Tsfc_beg+273)**4))
Lup_Boltz_RIGHT = pd.DataFrame(Boltz_cst*(( Tsfc_end+273)**4))

AA = pd.DataFrame(MayFox["Inc␣Radn"]*0.8 + Ad5_MayFox["L_Down"] -
    ↪ Ad5_MayFox["L_Up"])
BB = pd.DataFrame( - Ad5_MayFox["L_Down"] + Boltz_cst*(( T_sfc_MayFox
    ↪ +273)**4))

AA.plot()
#BB.plot()

plt.figure(19191)
plt.plot(Ad5_MayFox["L_Down"], "green")
plt.plot(Ad5_MayFox["L_Up"], "blue")
plt.plot(Lup_Boltz, ’red’)
plt.title("ADV␣RAD")
plt.show()

plt.figure(18181)
plt.plot(Ad5["L_Down"], "green")
plt.plot(Ad5["L_Up"], "blue")
plt.plot(Lup_Boltz, ’rx’)
plt.legend()
plt.show()

plt.figure(181)
plt.plot(Ad5_MayFox)
plt.plot(MayFox["Inc␣Radn"])
plt.show()

Ad5_May_PlotSW = pd.DataFrame(Ad5_MayFox["S_Up"])
Ad5_May_PlotSW = Ad5_May_PlotSW.rename({’S_Up’ : ’S\u2191_Adventdalen’},
   ↪ axis=1)

Ad5_May_PlotSW["S_Down␣Adventdalen"] = Ad5_MayFox["S_Down"]
Ad5_May_PlotSW = Ad5_May_PlotSW.rename({’S_Down␣Adventdalen’ : ’S\
   ↪ u2193_Adventdalen’},axis=1)

Ad5_May_PlotSW["S_Down␣Foxfonna"] = MayFox["Inc␣Radn"]
Ad5_May_PlotSW = Ad5_May_PlotSW.rename({’S_Down␣Foxfonna’ : ’S\
   ↪ u2193_Foxfonna’},axis=1)


          # u’\u2193’ \u2191
plt.figure(102)
ax =Ad5_May_PlotSW["S\u2193_Adventdalen"].plot(legend=True, color="tab:
   ↪ blue")
Ad5_May_PlotSW["S\u2191_Adventdalen"].plot(legend=True, color="tab:green"
   ↪ )
Ad5_May_PlotSW["S\u2193_Foxfonna"] .plot(legend=True, color="tab:purple")
plt.legend()
plt.title("Comparison␣Shortwave␣Radiation␣:␣Adventdalen␣Vs.␣Foxfonna")
plt.ylabel("[W/m2]", rotation=0,labelpad=25)
plt.xlabel("Time")
#plt.ylim(-100,100)
plt.show()


Q_net_MayFox_Boltz = pd.DataFrame(Q_net_MayFox_Boltz)
Q_net_MayFox_Boltz.columns= ["Q_net_Boltzman"]
```

```
Q_sens_MayFox = rho_air * C_air * C_Sensible * Brei_MayFox["WS_ms_Avg"] *
   ↪  (MayFox["Air␣Temp"] - T_sfc_MayFox ) #Cp = sp. heat. cap . air #
   ↪ ADJUST BY A STABILITY FACTOR (C_sensible)
Q_lat_MayFox = 22.2 * C_Latent * Brei_MayFox["WS_ms_Avg"] * (q_air_MayFox
   ↪  - q_sfc_MayFox)



Q_net_MayFox = pd.DataFrame(Q_net_MayFox)
Q_net_MayFox.columns = ["Q_net_MayFox"]

Q_sens_MayFox = pd.DataFrame(Q_sens_MayFox)
Q_sens_MayFox.columns = ["Q_sens_MayFox"]

Q_lat_MayFox = pd.DataFrame(Q_lat_MayFox)
Q_lat_MayFox.columns = ["Q_lat_MayFox"]

Q_g_MayFox = pd.DataFrame(Thermistors.df_B_may)
Q_g_MayFox.columns = ["Q_g_MayFox"]



# ==========================================
# Q_melt = Q_net_MayFox.add(Q_sens_MayFox, fill_value=0)
# Q_melt = Q_melt.add(Q_lat_MayFox, fill_value=0)
# Q_melt = Q_melt.sum(axis=1)
# Q_melt = pd.DataFrame(Q_melt)
# Q_melt.index = Q_lat_MayFox.index
# Q_melt.columns = ["Q_melt"]
#
# ====================================
Q_melt_Bltz = Q_net_MayFox_Boltz.add(Q_sens_MayFox, fill_value=0)
Q_melt_Bltz = Q_melt_Bltz.add(Q_lat_MayFox, fill_value=0)
Q_melt_Bltz = Q_melt_Bltz.sum(axis=1)
Q_melt_Bltz = pd.DataFrame(Q_melt_Bltz)
Q_melt_Bltz.index = Q_lat_MayFox.index
Q_melt_Bltz.columns = ["Q_melt_Bltz"]

#Met_Rate = Q_melt/(141*L_fusion) #M = Q[m/s] / (density*L_fusion)


plt.figure(101)
ax = Q_sens_MayFox.plot(legend=True)
Q_lat_MayFox.plot(ax=ax, legend=True)
Q_g_MayFox.plot(ax=ax, legend=True) #Using Thermistors 23 and 24
Q_net_MayFox_Boltz.plot(ax=ax, legend=True)
Q_melt_Bltz.plot(ax=ax, legend=True)
```

```
#Thermistors.May_40_48["Air Temp"].plot(ax=ax, legend=True)
plt.title("SEB_May")
plt.ylabel("[W/m2]",rotation=0,labelpad=25)
plt.xlabel("Time")
#plt.ylim(-100,300)
plt.legend(bbox_to_anchor=(1.0,1.0))
plt.show()


# =========================================
# Q_net_MayFox.plot(ax=ax, legend=True)
# Q_melt.plot(ax=ax, legend=True)
# =========================================




#%% ComparisonNET Boltzman and Adventhalen Radiation

Q_net_Boltz_LEFT = Q_net_MayFox_Boltz[(Q_net_MayFox_Boltz.index > ’
   ↪ 2021-05-27’) & (Q_net_MayFox_Boltz.index <= ’2021-06-01’)]
Q_net_Boltz_RIGHT = Q_net_MayFox_Boltz[(Q_net_MayFox_Boltz.index > ’
   ↪ 2021-06-01’) & (Q_net_MayFox_Boltz.index <= ’2021-06-11’)]

Q_net_Boltz_LEFT = Q_net_Boltz_LEFT.rename({’Q_net_Boltzman’ : ’$Q_{net}
   ↪ $_Boltzman␣(Temp␣20cm␣below␣surface,␣\u03B1=0.9)’},axis=1)
Q_net_Boltz_RIGHT = Q_net_Boltz_RIGHT.rename({’Q_net_Boltzman’ : ’$Q_{net
   ↪ }$_Boltzman␣(Temp␣30cm␣below␣surface,␣\u03B1=0.75)’},axis=1)

plt.figure(102)
Q_net_MayFox_renamed = Q_net_MayFox.rename({’Q_net_MayFox’ : ’$Q_{net}
   ↪ $_Adventhalen’},axis=1)
ax = Q_net_MayFox_renamed.plot(legend=True, color="tab:brown")
Q_net_Boltz_LEFT.plot(ax=ax, legend=True, color="tab:cyan")
Q_net_Boltz_RIGHT.plot(ax=ax, legend=True, color="tab:pink")
ax.axvline(pd.to_datetime(’2021-06-01’), color=’r’, linestyle=’:’, lw=2)
plt.legend(loc="lower␣right",bbox_to_anchor=(0.9,-0.5))
plt.title("Comparison␣$Q_{net}$␣-␣Adventhalen␣Vs.␣Boltzmann")
plt.ylabel("[W/m2]", rotation=0,labelpad=10)
plt.xlabel("Time",labelpad=-20)
#plt.ylim(-100,100)
plt.show()

#%% WEATHER PLOT

# ===========================
# plt.figure(10001)
# Brei_MayFox["WS_ms_Avg"].plot()
# plt.title("Wind Speed - Breinosa WS")
# plt.ylabel("[m/s]", rotation=0, labelpad=15)
# plt.xlabel("Time")
# plt.show()
#
# plt.figure(10002)
# MayFox["Air Temp"].plot()
# plt.title("Air Temperature - Foxfonna WS")
# plt.ylabel("[C]", rotation=0, labelpad=15)
# plt.xlabel("Time")
# plt.show()
#
#
#
#
# plt.figure(10003)
# MayFox["RH"].plot()
# plt.xlabel("Time")
# plt.title("Relative Humidity (%) - Foxfonna WS")
# plt.show()
#
# ================================

plt.figure(10004)
MayFox["Inc␣Radn"].plot()
plt.title("Incoming␣Shortwave␣Radiation␣-␣Foxfonna␣WS")
plt.ylabel("[W/m2]", rotation=0, labelpad=25)
plt.xlabel("Time")
plt.show()

POS = np.arange(0,360,45)
LAB = np.array([’N’, ’NE’, ’E’, ’SE’,"S","SW","W","NW"])

# ==========================
# plt.figure(10005)
# Brei_MayFox["WindDir"].plot()
# plt.yticks(POS,LAB)
# plt.title("Wind Direction - Breinosa WS")
# plt.xlabel("Time")
# plt.show()
# ============================

def uwind(x,y):
    u = np.cos(x*np.pi/180)*y
    return u

def vwind(x,y):
    v = np.sin(x*np.pi/180)*y
    return v


# ============================================
# plt.figure(10006)
# Brei_MayFox["BP_mBar"].plot()
# plt.title("Atmospheric Pressure - Breinosa WS")
# plt.xlabel("Time")
# plt.ylabel("[mBar]", rotation=0, labelpad=25)
# plt.show()
# ==============================================

#Ad5_AprilMay_renamed = Ad_AprilMay.rename({’S_Up’ : ’S_Up’,’S_Down’ : ’
   ↪ S_Down’, ’L_Up’ : ’L_Up’, ’L_Down’ : ’L_Down’},axis=1)

plt.figure(10007)
Ad5_AprilMay.plot()
plt.title("Radiations␣-␣Adventhalen␣WS")
plt.xlabel("Time")
plt.ylabel("[W/m2]", rotation=0, labelpad=25)
plt.show()

Ad5_AprilMay_SD = pd.DataFrame(Ad5_AprilMay["S_Down"])
Ad5_AprilMay_SU = pd.DataFrame(Ad5_AprilMay["S_Up"])
Ad5_AprilMay_LD = pd.DataFrame(Ad5_AprilMay["L_Down"])
Ad5_AprilMay_LU = pd.DataFrame(Ad5_AprilMay["L_Up"])

Ad5_AprilMay_SD = Ad5_AprilMay_SD.rename({’S_Down’ : ’S\u2193_Adventdalen
   ↪ ’},axis=1)
Ad5_AprilMay_SU = Ad5_AprilMay_SU.rename({’S_Up’ : ’S\u2191_Adventdalen’
   ↪ },axis=1)
Ad5_AprilMay_LD = Ad5_AprilMay_LD.rename({’L_Down’ : ’L\u2193_Adventdalen
   ↪ ’},axis=1)
Ad5_AprilMay_LU = Ad5_AprilMay_LU.rename({’L_Up’ : ’L\u2191_Adventdalen’
   ↪ },axis=1)

plt.figure(102)
ax = Ad5_AprilMay_SD.plot(legend=True, color="tab:blue")
Ad5_AprilMay_SU.plot(ax=ax, legend=True, color="tab:green")
Ad5_AprilMay_LD.plot(ax=ax, legend=True, color="tab:orange")
Ad5_AprilMay_LU.plot(ax=ax, legend=True, color="tab:red")
plt.legend()
plt.title("Comparison␣Net␣Radiation␣:␣Adventhalen␣Vs.␣Boltzmann")
plt.ylabel("[W/m2]", rotation=0,labelpad=25)
plt.xlabel("Time")
#plt.ylim(-100,100)
plt.show()
```

```
#%% Boltz_Adv Comparison



Lup_Boltz_renamed = Lup_Boltz.rename({’T(42)’ : ’L\u2191_Boltz’},axis=1)
Ad5_MayFox_renamed = Ad5_MayFox.rename({’L_Up’ : ’L\u2191_Adventhalen’},
   ↪ axis=1)
Lup_Boltz_LEFT = Lup_Boltz_LEFT.rename({’T(41)’ : ’L\u2191_Boltz␣(Temp.␣
   ↪ 20cm␣below␣surface)’},axis=1)
Lup_Boltz_RIGHT = Lup_Boltz_RIGHT.rename({’T(42)’ : ’L\u2191_Boltz␣(Temp.
   ↪ ␣30cm␣below␣surface)’},axis=1)
plt.figure(1099)
ax = Ad5_MayFox_renamed["L\u2191_Adventhalen"].plot(legend=True, color="
   ↪ tab:red")
Lup_Boltz_LEFT.plot(ax=ax, legend=True)
Lup_Boltz_RIGHT.plot(ax=ax, legend=True)
#Lup_Boltz_renamed.plot(ax=ax, legend=True)
ax.axvline(pd.to_datetime(’2021-06-01’), color=’r’, linestyle=’:’, lw=2)
plt.legend(loc="best",bbox_to_anchor=(0.37,0.28))
plt.title("Comparison␣L\u2191␣Radiation␣-␣Adventhalen␣Vs.␣Boltzmann")
plt.ylabel("[W/m2]", rotation=0,labelpad=25)
plt.xlabel("Time")
#plt.ylim(-100,100)
plt.show()

#%% Calculate Melt

Q_melt_exp = (391.1666666666667*L_fusion*0.15)/432000

Q_melt_Cumulative1 = Q_melt_Bltz[(Q_melt_Bltz.index > ’2021-06-06’) & (
   ↪ Q_melt_Bltz.index <= ’2021-06-11’)].sum()
Q_melt_Cumulative = Q_melt_Bltz[(Q_melt_Bltz.index > ’2021-06-06’) & (
   ↪ Q_melt_Bltz.index <= ’2021-06-11’)].sum()/len(Q_melt_Bltz[(
   ↪ Q_melt_Bltz.index > ’2021-06-06’) & (Q_melt_Bltz.index <= ’
   ↪ 2021-06-11’)])

#%%
```

[language=Python]

A.4 Code : Plot of the Snow Pits Data

```
################# This code visualizes the snow pits data #################

import numpy as np
import matplotlib as plt
import pandas as pd


colnames = ['Depth', 'T', 'depth', 'mass', "Longitude", "Latitude"]
colna = ['Depth', 'T', 'depth', 'mass']
colours = ['red', 'orange', 'yellow', 'green', 'blue', 'purple', 'violet' , 'cyan']


k12k = pd.read_csv('Rod1_2.csv', names = colna, header=0)["T"]
k13k = pd.read_csv('Rod1_3.csv', names = colna, header=0)["T"]


df1_1 = pd.read_csv('Rod1_1.csv', names = colnames, header=0) # Read each of my snow pit file.
df1_2 = pd.read_csv('Rod1_2.csv', names = colnames, header=0)
df1_3 = pd.read_csv('Rod1_3.csv', names = colnames, header=0)
df2_1 = pd.read_csv('Rod2_1.csv', names = colnames, header=0)
df2_2 = pd.read_csv('Rod2_2.csv', names = colnames, header=0)
df2_3 = pd.read_csv('Rod2_3.csv', names = colnames, header=0)
df3_1 = pd.read_csv('Rod3_1.csv', names = colnames, header=0)
df3_2 = pd.read_csv('Rod3_2.csv', names = colnames, header=0)
df3_3 = pd.read_csv('Rod3_3.csv', names = colnames, header=0)
df4_1 = pd.read_csv('Rod4_1.csv', names = colnames, header=0)
df4_2 = pd.read_csv('Rod4_2.csv', names = colnames, header=0)
df4_3 = pd.read_csv('Rod4_3.csv', names = colnames, header=0)
df5_1 = pd.read_csv('Rod5_1.csv', names = colnames, header=0)
df5_2 = pd.read_csv('Rod5_2.csv', names = colnames, header=0)
df5_3 = pd.read_csv('Rod5_3.csv', names = colnames, header=0)
df6_1 = pd.read_csv('Rod6_1.csv', names = colnames, header=0)
df6_2 = pd.read_csv('Rod6_2.csv', names = colnames, header=0)
df6_3 = pd.read_csv('Rod6_3.csv', names = colnames, header=0)
df7_1 = pd.read_csv('Rod7_1.csv', names = colnames, header=0)
df7_2 = pd.read_csv('Rod7_2.csv', names = colnames, header=0)
df8_1 = pd.read_csv('Rod8_1.csv', names = colnames, header=0)
df8_2 = pd.read_csv('Rod8_2.csv', names = colnames, header=0)
df8_3 = pd.read_csv('Rod8_3.csv', names = colnames, header=0)

A = [
     df1_1,df1_2,df1_3, #012
     df2_1,df2_2,df2_3, #345
     df3_1,df3_2,df3_3, #678
     df4_1,df4_2,df4_3, #9_10_11
     df5_1,df5_2,df5_3, #12_13_14
     df6_1,df6_2,df6_3, #15_16_17
     df7_1,df7_2, #18_19
     df8_1,df8_2,df8_3] #20_21_22

S1 = np.array([1527032, 8673770,0,0]) # Coordinates of the Stakes.
S2 = np.array([1527007, 8673868,0,0])
S3 = np.array([1526860, 8673823,0,0])
S4 = np.array([1526877, 8673734,0,0])
S5 = np.array([1526941, 8673803,0,0])

Diff_S2_S1 = S2-S1
Diff_S3_S1 = S3-S1
Diff_S4_S1 = S4-S1
Diff_S5_S1 = S5-S1

P1_1 = np.array([df1_1["Longitude"][0],df1_1["Latitude"][0],df1_1["Depth"
    ↪ ], df1_1["T"].astype(float)])
P1_2 = np.array([df1_2["Longitude"][0],df1_2["Latitude"][0],df1_2["Depth"
    ↪ ], df1_2["T"].astype(float)])
P1_3 = np.array([df1_3["Longitude"][0],df1_3["Latitude"][0],df1_3["Depth"
    ↪ ], df1_3["T"].astype(float)])
P2_1 = np.array([df2_1["Longitude"][0],df2_1["Latitude"][0],df2_1["Depth"
    ↪ ], df2_1["T"].astype(float)])
P2_2 = np.array([df2_2["Longitude"][0],df2_2["Latitude"][0],df2_2["Depth"
    ↪ ], df2_2["T"].astype(float)])
P2_3 = np.array([df2_3["Longitude"][0],df2_3["Latitude"][0],df2_3["Depth"
    ↪ ], df2_3["T"].astype(float)])
P3_1 = np.array([df3_1["Longitude"][0],df3_1["Latitude"][0],df3_1["Depth"
    ↪ ], df3_1["T"].astype(float)])
P3_2 = np.array([df3_2["Longitude"][0],df3_2["Latitude"][0],df3_2["Depth"
    ↪ ], df3_2["T"].astype(float)])
P3_3 = np.array([df3_3["Longitude"][0],df3_3["Latitude"][0],df3_3["Depth"
    ↪ ], df3_3["T"].astype(float)])
P4_1 = np.array([df4_1["Longitude"][0],df4_1["Latitude"][0],df4_1["Depth"
    ↪ ], df4_1["T"].astype(float)])
P4_2 = np.array([df4_2["Longitude"][0],df4_2["Latitude"][0],df4_2["Depth"
    ↪ ], df4_2["T"].astype(float)])
P4_3 = np.array([df4_3["Longitude"][0],df4_3["Latitude"][0],df4_3["Depth"
    ↪ ], df4_3["T"].astype(float)])
P5_1 = np.array([df5_1["Longitude"][0],df5_1["Latitude"][0],df5_1["Depth"
    ↪ ], df5_1["T"].astype(float)])
P5_2 = np.array([df5_2["Longitude"][0],df5_2["Latitude"][0],df5_2["Depth"
    ↪ ], df5_2["T"].astype(float)])
P5_3 = np.array([df5_3["Longitude"][0],df5_3["Latitude"][0],df5_3["Depth"
    ↪ ], df5_3["T"].astype(float)])
```

```
P6_1 = np.array([df6_1["Longitude"][0],df6_1["Latitude"][0],df6_1["Depth"], df6_1["T"].astype(float)])
P6_2 = np.array([df6_2["Longitude"][0],df6_2["Latitude"][0],df6_2["Depth"], df6_2["T"].astype(float)])
P6_3 = np.array([df6_3["Longitude"][0],df6_3["Latitude"][0],df6_3["Depth"], df6_3["T"].astype(float)])
P7_1 = np.array([df7_1["Longitude"][0],df7_1["Latitude"][0],df7_1["Depth"], df7_1["T"].astype(float)])
P7_2 = np.array([df7_2["Longitude"][0],df7_2["Latitude"][0],df7_2["Depth"], df7_2["T"].astype(float)])
P8_1 = np.array([df8_1["Longitude"][0],df8_1["Latitude"][0],df8_1["Depth"], df8_1["T"].astype(float)])
P8_2 = np.array([df8_2["Longitude"][0],df8_2["Latitude"][0],df8_2["Depth"], df8_2["T"].astype(float)])
P8_3 = np.array([df8_3["Longitude"][0],df8_3["Latitude"][0],df8_3["Depth"], df8_3["T"].astype(float)])

P1_1_S1 = P1_1-S1
P1_2_S1 = P1_2-S1
P1_3_S1 = P1_3-S1
P2_1_S1 = P2_1-S1
P2_2_S1 = P2_2-S1
P2_3_S1 = P2_3-S1
P3_1_S1 = P3_1-S1
P3_2_S1 = P3_2-S1
P3_3_S1 = P3_3-S1
P4_1_S1 = P4_1-S1
P4_2_S1 = P4_2-S1
P4_3_S1 = P4_3-S1
P5_1_S1 = P5_1-S1
P5_2_S1 = P5_2-S1
P5_3_S1 = P5_3-S1
P6_1_S1 = P6_1-S1
P6_2_S1 = P6_2-S1
P6_3_S1 = P6_3-S1
P7_1_S1 = P7_1-S1
P7_2_S1 = P7_2-S1
P8_1_S1 = P8_1-S1
P8_2_S1 = P8_2-S1
P8_3_S1 = P8_3-S1

tupple = (P1_1_S1,P1_2_S1,P1_3_S1, #012
         P2_1_S1,P2_2_S1,P2_3_S1, #345
         P3_1_S1,P3_2_S1,P3_3_S1, #678
         P4_1_S1,P4_2_S1,P4_3_S1, #9_10_11
         P5_1_S1,P5_2_S1,P5_3_S1, #12_13_14
         P6_1_S1,P6_2_S1,P6_3_S1, #15_16_17
         P7_1_S1,P7_2_S1, #18_19
```

```
          P8_1_S1,P8_2_S1,P8_3_S1) #20_21_22


#%% Plot the temperature data of each snow pit.


DoStuff = [0,3,6,9,12,15,18,20]
DoNothing = [1,2,4,5,7,8,10,11,13,14,16,17,19,21,22]


for i in np.arange(0,21,1) :
    print(i)
    if i == 18 :
        fig = plt.figure(i)
        ax = plt.axes(projection='3d')
        ax.set_zlim(0,2)
        ax.scatter(0,0,0, marker="x")
        ax.scatter(Diff_S2_S1[0],Diff_S2_S1[1],Diff_S2_S1[2], marker="x")
        ax.scatter(Diff_S3_S1[0],Diff_S3_S1[1],Diff_S3_S1[2], marker="x")
        ax.scatter(Diff_S4_S1[0],Diff_S4_S1[1],Diff_S4_S1[2], marker="x")
        ax.scatter(Diff_S5_S1[0],Diff_S5_S1[1],Diff_S5_S1[2], marker="x")
        ax.scatter(tupple[i][0],tupple[i][1],tupple[i][2], c=tupple[i][3],
            ↪ cmap='plasma', vmin = -9.4, vmax=0,depthshade=False)
        ax.scatter(tupple[i+1][0],tupple[i+1][1],tupple[i+1][2], c=tupple[
            ↪ i+1][3],cmap='plasma', vmin = -9.4, vmax=0,depthshade=False)

        ax.text(3, 3, .03, "S1", fontsize=15)
        ax.text(Diff_S2_S1[0]+3, Diff_S2_S1[1]+3, Diff_S2_S1[2]+.03, "S2",
            ↪  fontsize=15)
        ax.text(Diff_S3_S1[0]+3, Diff_S3_S1[1]+3, Diff_S3_S1[2]+.03, "S3",
            ↪  fontsize=15)
        ax.text(Diff_S4_S1[0]+3, Diff_S4_S1[1]+3, Diff_S4_S1[2]+.03, "S4",
            ↪  fontsize=15)
        ax.text(Diff_S5_S1[0]+3, Diff_S5_S1[1]+3, Diff_S5_S1[2]+.03, "S5",
            ↪  fontsize=15)

        ax.text(0.6, -0.3, 3.2, "Depth␣[m]", fontsize=10)


        p = ax.scatter(tupple[i][0],tupple[i][1],tupple[i][2], c=tupple[i
            ↪ ][3],cmap='plasma', vmin = -9.4, vmax=0,depthshade=False)
        plt.xlabel("Distance␣(W-E)␣[m]")
        plt.ylabel("Distance␣(N-S)␣[m]")
        fig.colorbar(p, label = "Temperature␣C")
        plt.show()


        step = 2
        continue

    if i in DoStuff :
        fig = plt.figure(i)
        ax = plt.axes(projection='3d')
        ax.set_zlim(0,2)
        ax.scatter(0,0,0, marker="x")

        ax.scatter(Diff_S2_S1[0],Diff_S2_S1[1],Diff_S2_S1[2], marker="x")
        ax.scatter(Diff_S3_S1[0],Diff_S3_S1[1],Diff_S3_S1[2], marker="x")
        ax.scatter(Diff_S4_S1[0],Diff_S4_S1[1],Diff_S4_S1[2], marker="x")
        ax.scatter(Diff_S5_S1[0],Diff_S5_S1[1],Diff_S5_S1[2], marker="x")


        ax.scatter(tupple[i][0],tupple[i][1],tupple[i][2], c=tupple[i][3],
            ↪ cmap='plasma', vmin = -9.4, vmax=0,depthshade=False)
        ax.scatter(tupple[i+1][0],tupple[i+1][1],tupple[i+1][2], c=tupple[
            ↪ i+1][3],cmap='plasma', vmin = -9.4, vmax=0,depthshade=False)
        ax.scatter(tupple[i+2][0],tupple[i+2][1],tupple[i+2][2], c=tupple[
            ↪ i+2][3],cmap='plasma', vmin = -9.4, vmax=0,depthshade=False)

        ax.text(3, 3, .03, "S1", fontsize=15)
        ax.text(Diff_S2_S1[0]+3, Diff_S2_S1[1]+3, Diff_S2_S1[2]+.03, "S2",
            ↪  fontsize=15)
        ax.text(Diff_S3_S1[0]+3, Diff_S3_S1[1]+3, Diff_S3_S1[2]+.03, "S3",
            ↪  fontsize=15)
        ax.text(Diff_S4_S1[0]+3, Diff_S4_S1[1]+3, Diff_S4_S1[2]+.03, "S4",
            ↪  fontsize=15)
        ax.text(Diff_S5_S1[0]+3, Diff_S5_S1[1]+3, Diff_S5_S1[2]+.03, "S5",
            ↪  fontsize=15)

        ax.text(0.6, -0.3, 3.2, "Depth␣[m]", fontsize=10)

        p = ax.scatter(tupple[i+2][0],tupple[i+2][1],tupple[i+2][2], c=
            ↪ tupple[i+2][3],cmap='plasma', vmin = -9.4, vmax=0,depthshade
            ↪ =False)
        plt.xlabel("Distance␣(W-E)␣[m]")
        plt.ylabel("Distance␣(N-S)␣[m]")
        fig.colorbar(p, label = "Temperature␣C")
        plt.show()
        continue

    if i in DoNothing :
        continue

#%% Calculation of the heat flow for each pit (Same command repeated
    ↪ each time).
```

```
#%% q1_1

E1_1 = []
q1_1 = []
for i in np.arange(1, len(df1_1)) :

    dTdz_vals1_1 = (df1_1["T"][i] - df1_1["T"][i-1])/0.10
    E1_1.append(dTdz_vals1_1+273)
E1_1 = pd.DataFrame(E1_1)

for i in np.arange(0, len(df1_1)) :

    DensityMg1_1 = df1_1["mass"][0:11]*10**(-3)
    k1_1 = 2.22362*DensityMg1_1**(1.885)
    q1_1 = k1_1[0:11]*E1_1[0:11]
    q1_1 = q1_1[0]

q1_1 = pd.DataFrame(q1_1)
q1_1 = q1_1.set_index(df1_1["depth"][0:11])
q1_1.columns = ["07/05"]
q1_1.plot(legend=True)

print(q1_1)

#%% q1_2

E1_2 = []
q1_2 = []
for i in np.arange(1, len(df1_2)) :

    dTdz_vals1_2 = (df1_2["T"][i] - df1_2["T"][i-1])/0.10
    E1_2.append(dTdz_vals1_2+273)
E1_2 = pd.DataFrame(E1_2)

for i in np.arange(0, len(df1_2)) :

    DensityMg1_2 = df1_2["mass"][0:5]*10**(-3)
    k1_2 = 2.22362*DensityMg1_2**(1.885)
    q1_2 = k1_2[0:5]*E1_2[0:5]
    q1_2 = q1_2[0]

q1_2 = pd.DataFrame(q1_2)
q1_2 = q1_2.set_index(df1_2["depth"][0:5])
q1_2.columns = ["07/05"]
q1_2.plot(legend=True)

plt.plot(q1_2)
print(q1_2)
```

```
#%% q1_3

E1_3 = []
q1_3 = []
for i in np.arange(1, len(df1_3)) :

    dTdz_vals1_3 = (df1_3["T"][i] - df1_3["T"][i-1])/0.10
    E1_3.append(dTdz_vals1_3+273)
E1_3 = pd.DataFrame(E1_3)

for i in np.arange(0, len(df1_3)) :

    DensityMg1_3 = df1_3["mass"][0:8]*10**(-3)
    k1_3 = 2.22362*DensityMg1_3**(1.885)
    q1_3 = k1_3[0:8]*E1_3[0:8]
    q1_3 = q1_3[0]

q1_3 = pd.DataFrame(q1_3)
q1_3 = q1_3.set_index(df1_3["depth"][0:7])
q1_3.columns = ["07/05"]
q1_3.plot(legend=True)

plt.plot(q1_3)
print(q1_3)


#%% q2_1

E2_1 = []
q2_1 = []
for i in np.arange(1, len(df2_1)) :

    dTdz_vals2_1 = (df2_1["T"][i] - df2_1["T"][i-1])/0.10
    E2_1.append(dTdz_vals2_1+273)
E2_1 = pd.DataFrame(E2_1)

for i in np.arange(0, len(df2_1)) :

    DensityMg2_1 = df2_1["mass"][0:14]*10**(-3)
    k2_1 = 2.22362*DensityMg2_1**(1.885)
    q2_1 = k2_1[0:14]*E2_1[0:14]
    q2_1 = q2_1[0]

q2_1 = pd.DataFrame(q2_1)
q2_1 = q2_1.set_index(df2_1["depth"][0:14])
q2_1.columns = ["16/05"]
q2_1.plot(legend=True)
```

```
plt.plot(q2_1)
print(q2_1)

#%% q2_2

E2_2 = []
q2_2 = []
for i in np.arange(1, len(df2_2)) :

    dTdz_vals2_2 = (df2_2["T"][i] - df2_2["T"][i-1])/0.10
    E2_2.append(dTdz_vals2_2+273)
E2_2 = pd.DataFrame(E2_2)

for i in np.arange(0, len(df2_2)) :

    DensityMg2_2 = df2_2["mass"][0:14]*10**(-3)
    k2_2 = 2.22362*DensityMg2_2**(1.885)
    q2_2 = k2_2[0:14]*E2_2[0:14]
    q2_2 = q2_2[0]

q2_2 = pd.DataFrame(q2_2)
q2_2 = q2_2.set_index(df2_2["depth"][0:14])
q2_2.columns = ["16/05"]
q2_2.plot(legend=True)

plt.plot(q2_2)
print(q2_2)


#%% q2_3

E2_3 = []
q2_3 = []
for i in np.arange(1, len(df2_3)) :

    dTdz_vals2_3 = (df2_3["T"][i] - df2_3["T"][i-1])/0.10
    E2_3.append(dTdz_vals2_3+273)
E2_3 = pd.DataFrame(E2_3)

for i in np.arange(0, len(df2_3)) :

    DensityMg2_3 = df2_3["mass"][0:12]*10**(-3)
    k2_3 = 2.22362*DensityMg2_3**(1.885)
    q2_3 = k2_3[0:12]*E2_3[0:12]
    q2_3 = q2_3[0]

q2_3 = pd.DataFrame(q2_3)
```

```
q2_3 = q2_3.set_index(df2_3["depth"][0:12])
q2_3.columns = ["16/05"]
q2_3.plot(legend=True)

plt.plot(q2_3)
print(q2_3)


#%% q3_1

E3_1 = []
q3_1 = []
for i in np.arange(1, len(df3_1)) :

    dTdz_vals3_1 = (df3_1["T"][i] - df3_1["T"][i-1])/0.10
    E3_1.append(dTdz_vals3_1+273)
E3_1 = pd.DataFrame(E3_1)

for i in np.arange(0, len(df3_1)) :

    DensityMg3_1 = df3_1["mass"][0:9]*10**(-3)
    k3_1 = 2.22362*DensityMg3_1**(1.885)
    q3_1 = k3_1[0:9]*E3_1[0:9]
    q3_1 = q3_1[0]

q3_1 = pd.DataFrame(q3_1)
q3_1 = q3_1.set_index(df3_1["depth"][0:9])
q3_1.columns = ["22/05"]
q3_1.plot(legend=True)

print(q3_1)


#%% q3_2

E3_2 = []
q3_2 = []
for i in np.arange(1, len(df3_2)) :

    dTdz_vals3_2 = (df3_2["T"][i] - df3_2["T"][i-1])/0.10
    E3_2.append(dTdz_vals3_2+273)
E3_2 = pd.DataFrame(E3_2)

for i in np.arange(0, len(df3_2)) :

    DensityMg3_2 = df3_2["mass"][1:9]*10**(-3)
    k3_2 = 2.22362*DensityMg3_2**(1.885)
    k3_2.index = E3_2[0:8].index
    q3_2 = k3_2*E3_2
    q3_2 = q3_2[0]

q3_2 = pd.DataFrame(q3_2)
q3_2 = q3_2.set_index(df3_2["depth"][0:9])
q3_2.columns = ["22/05"]
q3_2.plot(legend=True)
print(q3_2)

#%% q3_3

E3_3 = []
q3_3 = []
for i in np.arange(1, len(df3_3)) :

    dTdz_vals3_3 = (df3_3["T"][i] - df3_3["T"][i-1])/0.10
    E3_3.append(dTdz_vals3_3+273)
E3_3 = pd.DataFrame(E3_3)

for i in np.arange(0, len(df3_3)) :

    DensityMg3_3 = df3_3["mass"][0:7]*10**(-3)
    k3_3 = 2.22362*DensityMg3_3**(1.885)
    q3_3 = k3_3[0:7]*E3_3[0:7]
    q3_3 = q3_3[0]

q3_3 = pd.DataFrame(q3_3)
q3_3 = q3_3.set_index(df3_3["depth"][0:7])
q3_3.columns = ["22/05"]
q3_3.plot(legend=True)

plt.plot(q3_3)
print(q3_3)

#%% q4_1

E4_1 = []
q4_1 = []
for i in np.arange(1, len(df4_1)) :

    dTdz_vals4_1 = (df4_1["T"][i] - df4_1["T"][i-1])/0.10
    print(dTdz_vals4_1)

    E4_1.append(dTdz_vals4_1+273)
E4_1 = pd.DataFrame(E4_1)

for i in np.arange(0, len(df4_1)) :
    DensityMg4_1 = df4_1["mass"][1:8]*10**(-3)
    k4_1 = 2.22362*DensityMg4_1**(1.885)
    k4_1.index = E4_1[0:7].index
    q4_1 = k4_1[0:7]*E4_1[0:7]
    q4_1 = q4_1[0]

q4_1 = pd.DataFrame(q4_1)
q4_1 = q4_1.set_index(df4_1["depth"][1:8])
q4_1.columns = ["06/06"]
q4_1.plot(legend=True)

plt.plot(q4_1)
print(q4_1)

#%% q4_2

E4_2 = []
q4_2 = []
for i in np.arange(1, len(df4_2)) :

    dTdz_vals4_2 = (df4_2["T"][i] - df4_2["T"][i-1])/0.10
    E4_2.append(dTdz_vals4_2+273)
E4_2 = pd.DataFrame(E4_2)

for i in np.arange(0, len(df4_2)) :

    DensityMg4_2 = df4_2["mass"][0:4]*10**(-3)
    k4_2 = 2.22362*DensityMg4_2**(1.885)
    q4_2 = k4_2[0:4]*E4_2[0:4]
    q4_2 = q4_2[0]

q4_2 = pd.DataFrame(q4_2)
q4_2 = q4_2.set_index(df4_2["depth"][0:4])
q4_2.columns = ["06/06"]
q4_2.plot(legend=True)

plt.plot(q4_2)
print(q4_2)


#%% q4_3

E4_3 = []
q4_3 = []
for i in np.arange(1, len(df4_3)) :

    dTdz_vals4_3 = (df4_3["T"][i] - df4_3["T"][i-1])/0.10
    E4_3.append(dTdz_vals4_3+273)
```

```
E4_3 = pd.DataFrame(E4_3)

for i in np.arange(0, len(df4_3)) :

    DensityMg4_3 = df4_3["mass"][0:6]*10**(-3)
    k4_3 = 2.22362*DensityMg4_3**(1.885)
    q4_3 = k4_3[0:6]*E4_3[0:6]
    q4_3 = q4_3[0]

q4_3 = pd.DataFrame(q4_3)
q4_3 = q4_3.set_index(df4_3["depth"][0:6])
q4_3.columns = ["06/06"]
q4_3.plot(legend=True)

plt.plot(q4_3)
print(q4_3)



#%% q5_1

E5_1 = []
q5_1 = []
for i in np.arange(1, len(df5_1)) :

    dTdz_vals5_1 = (df5_1["T"][i] - df5_1["T"][i-1])/0.10
    E5_1.append(dTdz_vals5_1+273)
E5_1 = pd.DataFrame(E5_1)

for i in np.arange(0, len(df5_1)) :

    DensityMg5_1 = df5_1["mass"][0:10]*10**(-3)
    k5_1 = 2.22362*DensityMg5_1**(1.885)
    q5_1 = k5_1[0:10]*E5_1[0:10]
    q5_1 = q5_1[0]

q5_1 = pd.DataFrame(q5_1)
q5_1 = q5_1.set_index(df5_1["depth"][0:10])
q5_1.columns = ["11/06"]
q5_1.plot(legend=True)

plt.plot(q5_1)
print(q5_1)



#%% q5_2
```

```
E5_2 = []
q5_2 = []
for i in np.arange(1, len(df5_2)) :

    dTdz_vals5_2 = (df5_2["T"][i] - df5_2["T"][i-1])/0.10
    E5_2.append(dTdz_vals5_2+273)
E5_2 = pd.DataFrame(E5_2)

for i in np.arange(0, len(df5_2)) :

    DensityMg5_2 = df5_2["mass"][0:9]*10**(-3)
    k5_2 = 2.22362*DensityMg5_2**(1.885)
    q5_2 = k5_2[0:9]*E5_2[0:9]
    q5_2 = q5_2[0]


q5_2 = pd.DataFrame(q5_2)
q5_2 = q5_2.set_index(df5_2["depth"][0:9])
q5_2.columns = ["11/06"]
q5_2.plot(legend=True)


print(q5_2)



#%% q5_3

E5_3 = []
q5_3 = []
for i in np.arange(1, len(df5_3)) :

    dTdz_vals5_3 = (df5_3["T"][i] - df5_3["T"][i-1])/0.10
    E5_3.append(dTdz_vals5_3+273)
E5_3 = pd.DataFrame(E5_3)

for i in np.arange(0, len(df5_3)) :

    DensityMg5_3 = df5_3["mass"][0:6]*10**(-3)
    k5_3 = 2.22362*DensityMg5_3**(1.885)
    q5_3 = k5_3[0:6]*E5_3[0:6]
    q5_3 = q5_3[0]

q5_3 = pd.DataFrame(q5_3)
q5_3 = q5_3.set_index(df5_3["depth"][0:6])
q5_3.columns = ["11/06"]
q5_3.plot(legend=True)
```

```
print(q5_3)

#%% q6_1

E6_1 = []
q6_1 = []
for i in np.arange(1, len(df6_1)) :

    dTdz_vals6_1 = (df6_1["T"][i] - df6_1["T"][i-1])/0.10
    E6_1.append(dTdz_vals6_1+273)
E6_1 = pd.DataFrame(E6_1)

for i in np.arange(0, len(df6_1)) :

    DensityMg6_1 = df6_1["mass"][0:7]*10**(-3)
    k6_1 = 2.22362*DensityMg6_1**(1.885)
    q6_1 = k6_1[0:7]*E6_1[0:7]
    q6_1 = q6_1[0]


q6_1 = pd.DataFrame(q6_1)
q6_1 = q6_1.set_index(df6_1["depth"][0:7])
q6_1.columns = ["16/06"]
q6_1.plot(legend=True)


print(q6_1)

#%% q6_2

E6_2 = []
q6_2 = []
for i in np.arange(1, len(df6_2)) :

    dTdz_vals6_2 = (df6_2["T"][i] - df6_2["T"][i-1])/0.10
    E6_2.append(dTdz_vals6_2+273)
E6_2 = pd.DataFrame(E6_2)

for i in np.arange(0, len(df6_2)) :

    DensityMg6_2 = df6_2["mass"][0:7]*10**(-3)
    k6_2 = 2.22362*DensityMg6_2**(1.885)
    q6_2 = k6_2[0:7]*E6_2[0:7]
    q6_2 = q6_2[0]

q6_2 = pd.DataFrame(q6_2)
q6_2 = q6_2.set_index(df6_2["depth"][0:7])
```

```
q6_2.columns = ["16/06"]
q6_2.plot(legend=True)


print(q6_2)


#%% q6_3

E6_3 = []
q6_3 = []
for i in np.arange(1, len(df6_3)) :

    dTdz_vals6_3 = (df6_3["T"][i] - df6_3["T"][i-1])/0.10
    E6_3.append(dTdz_vals6_3+273)
E6_3 = pd.DataFrame(E6_3)

for i in np.arange(0, len(df6_3)) :

    DensityMg6_3 = df6_3["mass"][0:6]*10**(-3)
    k6_3 = 2.22362*DensityMg6_3**(1.885)
    q6_3 = k6_3[0:6]*E6_3[0:6]
    q6_3 = q6_3[0]

q6_3 = pd.DataFrame(q6_3)
q6_3 = q6_3.set_index(df6_3["depth"][0:6])
q6_3.columns = ["16/06"]
q6_3.plot(legend=True)


print(q6_3)

#%% q7_1

E7_1 = []
q7_1 = []
for i in np.arange(1, len(df7_1)) :

    dTdz_vals7_1 = (df7_1["T"][i] - df7_1["T"][i-1])/0.10
    E7_1.append(dTdz_vals7_1+273)
E7_1 = pd.DataFrame(E7_1)

for i in np.arange(0, len(df7_1)) :

    DensityMg7_1 = df7_1["mass"][0:7]*10**(-3)
    k7_1 = 2.22362*DensityMg7_1**(1.885)
    q7_1 = k7_1[0:7]*E7_1[0:7]
    q7_1 = q7_1[0]
```

```
q7_1 = pd.DataFrame(q7_1)
q7_1 = q7_1.set_index(df7_1["depth"][0:7])
q7_1.columns = ["20/06"]
q7_1.plot(legend=True)


print(q7_1)


#%% q7_2

E7_2 = []
q7_2 = []
for i in np.arange(1, len(df7_2)) :

    dTdz_vals7_2 = (df7_2["T"][i] - df7_2["T"][i-1])/0.10
    E7_2.append(dTdz_vals7_2+273)
E7_2 = pd.DataFrame(E7_2)

for i in np.arange(0, len(df7_2)) :

    DensityMg7_2 = df7_2["mass"][0:9]*10**(-3)
    k7_2 = 2.22362*DensityMg7_2**(1.885)
    q7_2 = k7_2[0:9]*E7_2[0:9]
    q7_2 = q7_2[0]

q7_2 = pd.DataFrame(q7_2)
q7_2 = q7_2.set_index(df7_2["depth"][0:9])
q7_2.columns = ["20/06"]
q7_2.plot(legend=True)

print(q7_2)



#%% q8_1

E8_1 = []
q8_1 = []
for i in np.arange(1, len(df8_1)) :

    dTdz_vals8_1 = (df8_1["T"][i] - df8_1["T"][i-1])/0.10
    E8_1.append(dTdz_vals8_1+273)
E8_1 = pd.DataFrame(E8_1)

for i in np.arange(0, len(df8_1)) :
    DensityMg8_1 = df8_1["mass"][0:5]*10**(-3)
    k8_1 = 2.22362*DensityMg8_1**(1.885)
    q8_1 = k8_1[0:5]*E8_1[0:5]
    q8_1 = q8_1[0]
q8_1 = pd.DataFrame(q8_1)
q8_1 = q8_1.set_index(df8_1["depth"][0:5])
q8_1.columns = ["26/06"]
q8_1.plot(legend=True)

print(q8_1)



#%% q8_2

E8_2 = []
q8_2 = []
for i in np.arange(1, len(df8_2)) :

    dTdz_vals8_2 = (df8_2["T"][i] - df8_2["T"][i-1])/0.10
    E8_2.append(dTdz_vals8_2+273)
E8_2 = pd.DataFrame(E8_2)

for i in np.arange(0, len(df8_2)) :

    DensityMg8_2 = df8_2["mass"][0:6]*10**(-3)
    k8_2 = 2.22362*DensityMg8_2**(1.885)
    q8_2 = k8_2[0:6]*E8_2[0:6]
    q8_2 = q8_2[0]

q8_2 = pd.DataFrame(q8_2)
q8_2 = q8_2.set_index(df8_2["depth"][0:6])
q8_2.columns = ["26/06"]
q8_2.plot(legend=True)

plt.plot(q8_2)
print(q8_2)



#%% q8_3

E8_3 = []
q8_3 = []
for i in np.arange(1, len(df8_3)) :

    dTdz_vals8_3 = (df8_3["T"][i] - df8_3["T"][i-1])/0.10
    E8_3.append(dTdz_vals8_3+273)
```

```
E8_3 = pd.DataFrame(E8_3)

for i in np.arange(0, len(df8_3)) :

    DensityMg8_3 = df8_3["mass"][0:5]*10**(-3)
    k8_3 = 2.22362*DensityMg8_3**(1.885)
    q8_3 = k8_3[0:5]*E8_3[0:5]
    q8_3 = q8_3[0]

q8_3 = pd.DataFrame(q8_3)
q8_3 = q8_3.set_index(df8_3["depth"][0:5])
q8_3.columns = ["26/06"]
q8_3.plot(legend=True)

plt.plot(q8_3)
print(q8_3)

#%% To make plot look nice.


# ============================================
#
# q1_1 = pd.DataFrame(q1_1)
# q1_1.columns = ["07/05"]
# q1_2 = pd.DataFrame(q1_2)
# q1_2.columns = ["07/05"]
# q1_3 = pd.DataFrame(q1_3)
# q2_1 = pd.DataFrame(q2_1)
# q2_1.columns = ["16/05"]
# q2_2 = pd.DataFrame(q2_2)
# q2_2.columns = ["16/05"]
# q2_3 = pd.DataFrame(q2_3)
# q2_3.columns = ["16/05"]
#
# q3_1 = pd.DataFrame(q1_1)
# q3_1.columns = ["22/05"]
# q3_2 = pd.DataFrame(q1_2)
# q3_2.columns = ["22/05"]
# q3_3 = pd.DataFrame(q1_3)
# q3_3.columns = ["22/05"]
# q4_1 = pd.DataFrame(q2_1)
# q4_1.columns = ["06/06"]
# q4_2 = pd.DataFrame(q2_2)
# q4_2.columns = ["06/06"]
# q4_3 = pd.DataFrame(q2_3)
# q4_3.columns = ["06/06"]
#
# q5_1 = pd.DataFrame(q1_1)
# q5_1.columns = ["11/06"]
# q5_2 = pd.DataFrame(q1_2)
# q5_2.columns = ["11/06"]
# q5_3 = pd.DataFrame(q1_3)
# q5_3.columns = ["11/06"]
# q6_1 = pd.DataFrame(q2_1)
# q6_1.columns = ["16/06"]
# q6_2 = pd.DataFrame(q2_2)
# q6_2.columns = ["16/06"]
# q6_3 = pd.DataFrame(q1_3)
# q6_3.columns = ["16/06"]
#
# q7_1 = pd.DataFrame(q7_1)
# q7_1.columns = ["20/06"]
# q7_2 = pd.DataFrame(q7_2)
# q7_2.columns = ["20/06"]
# q8_1 = pd.DataFrame(q8_1)
# q8_1.columns = ["26/06"]
# q8_2 = pd.DataFrame(q8_2)
# q8_2.columns = ["26/06"]
# q8_3 = pd.DataFrame(q8_3)
# q8_3.columns = ["26/06"]
# =================================================

plt.figure()
ax = q1_1.plot(legend=False,color="tab:blue")
q1_2.plot(ax=ax,color="tab:blue",legend=False)
q1_3.plot(ax=ax,color="tab:blue",legend=False)
q2_1.plot(ax=ax,color="tab:orange",legend=False)
q2_2.plot(ax=ax,color="tab:orange",legend=False)
q2_3.plot(ax=ax,color="tab:orange",legend=False)
q3_1.plot(ax=ax,color="tab:green",legend=False)
q3_2.plot(ax=ax,color="tab:green",legend=False)
q3_3.plot(ax=ax,color="tab:green",legend=False)
q4_1.plot(ax=ax,color="tab:red",legend=False)
q4_2.plot(ax=ax,color="tab:red",legend=False)
q4_3.plot(ax=ax,color="tab:red",legend=False)
q5_1.plot(ax=ax,color="tab:purple",legend=False)
q5_2.plot(ax=ax,color="tab:purple",legend=False)
q5_3.plot(ax=ax,color="tab:purple",legend=False)
q6_1.plot(ax=ax,color="tab:brown",legend=False)
q6_2.plot(ax=ax,color="tab:brown",legend=False)
q6_3.plot(ax=ax,color="tab:brown",legend=False)
q7_1.plot(ax=ax,color="tab:pink",legend=False)
q7_2.plot(ax=ax,color="tab:pink",legend=False)
q8_1.plot(ax=ax,color="tab:grey",legend=False)
q8_2.plot(ax=ax,color="tab:grey",legend=False)
q8_3.plot(ax=ax,color="tab:grey",legend=False)
```

```



blue = mpatches.Patch(color=’tab:blue’, label=’07/05/21’)
green= mpatches.Patch(color=’tab:orange’, label=’16/05/21’)
pink = mpatches.Patch(color=’tab:green’, label=’22/05/21’)
purple = mpatches.Patch(color=’tab:red’, label=’06/06/21’)
cyan = mpatches.Patch(color=’tab:purple’, label=’11/06/21’)
orange = mpatches.Patch(color=’tab:brown’, label=’16/06/21’)
black = mpatches.Patch(color=’tab:pink’, label=’20/06/21’)
red = mpatches.Patch(color=’tab:grey’, label=’26/06/21’)



plt.legend(bbox_to_anchor=(1.0, 1.0), handles=[blue,green,pink,purple,
   ↪ cyan,orange,black,red])
plt.title("Heat␣Flow␣in␣Snow␣Pack")
plt.ylabel("W", rotation=0)
plt.xlabel("Snow␣Depth␣(cm)␣␣(0cm:␣snow␣surface.)")
plt.show()
```

[language=Python]