Eric_Keenan_HW2

February 25, 2020

1 Ice Sheets and Climate - Eric Keenan - Homework # 2

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from datetime import datetime
from pandas.plotting import register_matplotlib_converters
register_matplotlib_converters()
```

In [2]:
```
# Function to load in aws data
def aws_timeseries(file_path, column_num):

    # File Path (absolute path)
    file_path = file_path

    # Get number of timesteps
    with open(file_path, 'r') as myfile:
        data = myfile.readlines()
    rows = len(data)

    # Get data
    time = []
    value = []
    for j in range(0, rows):
        line = data[j][:]

        # Parse the line
        line = line.split()
        time.append(datetime.strptime(line[0],'%Y/%m/%d'))
        value.append(float(line[column_num]))

    # Convert to Data Frame
    time_series = pd.DataFrame(value, index = time)
    time_series = time_series.replace(-999.50000, np.nan)

    # Trim to year
    time_series = time_series.truncate(before=pd.to_datetime("1998-01-01 00:00:00"), \
```

```
                                                       after=pd.to_datetime("1998-12-31 00:00:00"))

            return time_series
```

2 Problem 1

1.a. The temperature and wind sensors height above the snow surface changes with time as the station becomes burried. Thus to provide a consistent record the observations are corrected to 2 m and 10 m above the surface.

1.b. For this assignment we will look at year 1998.

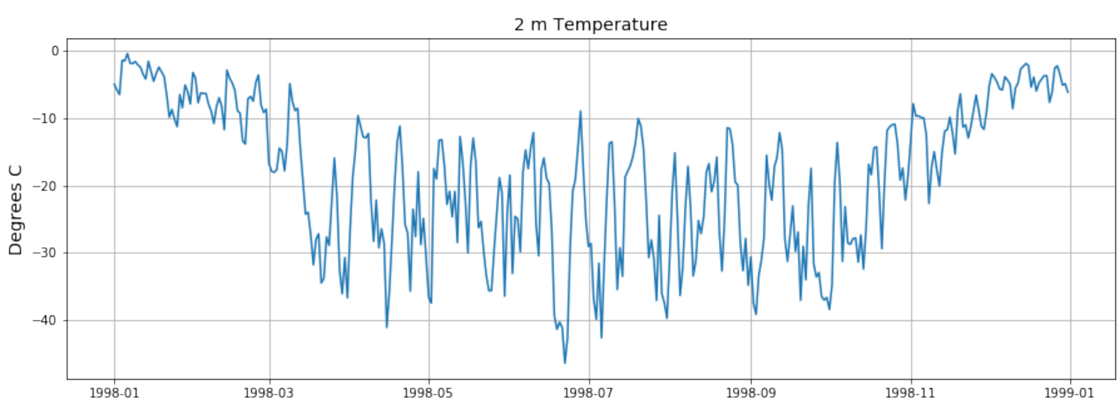
3 Problem 2 - AWS 4

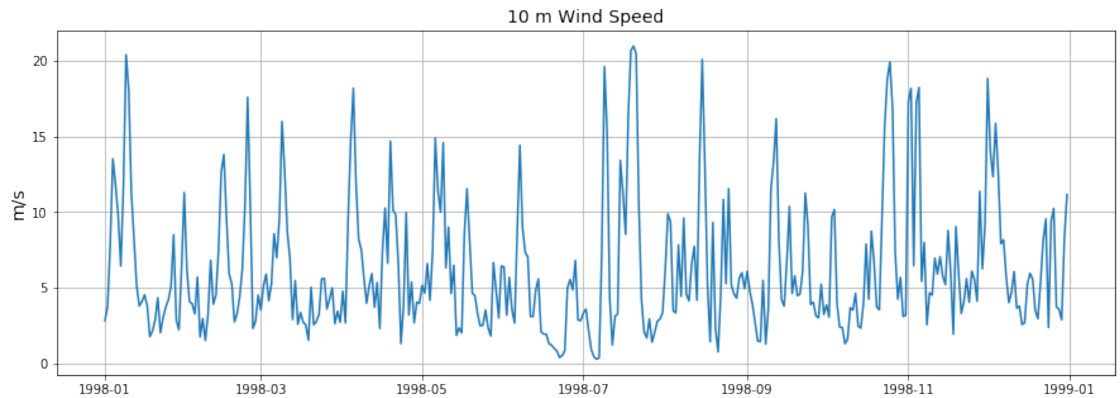
2.a. 2 m air temperature is generally higher in the summer and with larger variability in the winter than summer. At AWS 4 there is no readily apparent seasonal cycle in wind speed. However there is large variability in daily averaged wind speeds. Notably two periods of low wind speed in winter coincide with large surface inversions, likely due to a lack of turbulence caused by low wind speed.

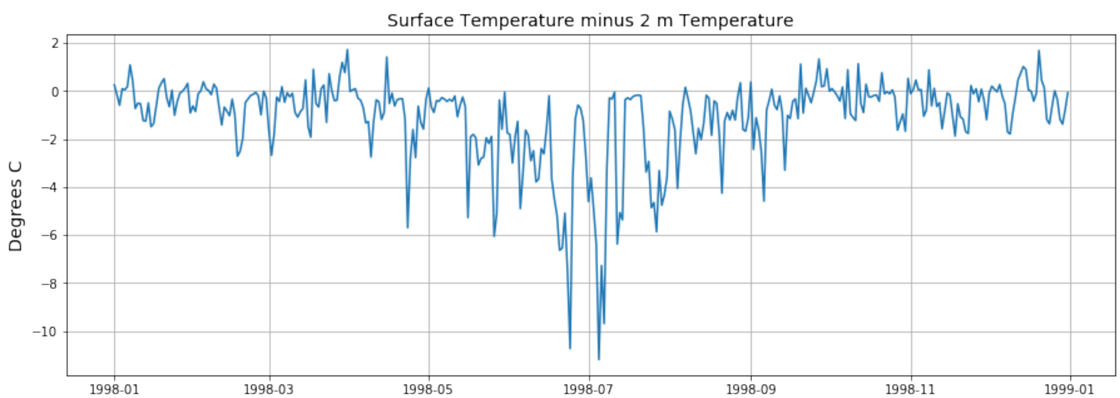
2.b. Difference between surface temperature and 2 m air temperature are largest on cold winter days with low wind speed. This is important for the sensible heat flux which is driven by a near surface temperature gradient! See scatter plots below. The difference is smallest on windy days.

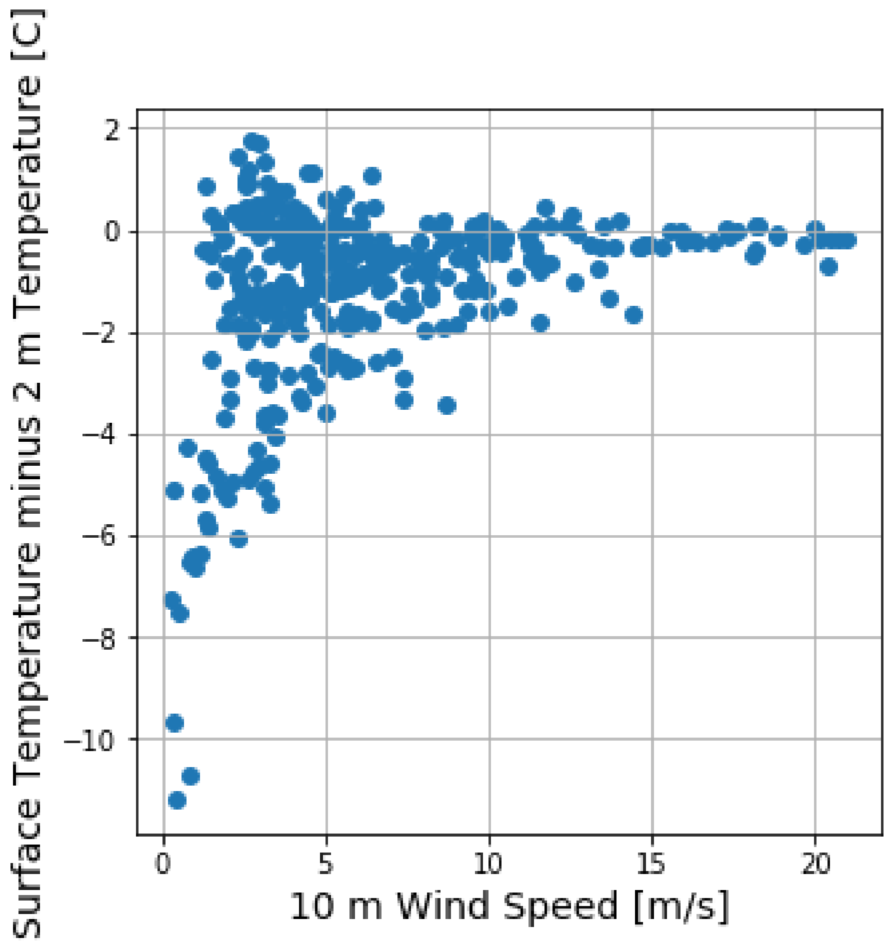
```
In [3]: # File paths
        aws4_path = "AWS4_DAY.txt"
        aws9_path = "AWS9_DAY.txt"

        # Load data (colum numbers: 4 = t2m, 8 = 10 m wind, 5 = surface temperature)
        aws4_t2m = aws_timeseries(aws4_path, 4) - 273.15
        aws4_ws10m = aws_timeseries(aws4_path, 8)
        aws4_ts = aws_timeseries(aws4_path, 5) - 273.15
        # aws4_ts = aws_timeseries(aws4_path, 6) - 273.15
        aws4_SHF = aws_timeseries(aws4_path, 15)
        aws4_LHF = aws_timeseries(aws4_path, 16)

        # Calculations
        aws4_tdiff = aws4_ts - aws4_t2m

        # Plot 2 meter temperature
        fig = plt.figure(figsize=(15,5))
        plt.plot(aws4_t2m)
        plt.title("2 m Temperature", fontsize=14)
        plt.ylabel("Degrees C", fontsize=14)
        plt.grid()
```

```python
# Plot 10 meter wind speed
fig = plt.figure(figsize=(15,5))
plt.plot(aws4_ws10m)
plt.title("10 m Wind Speed", fontsize=14)
plt.ylabel("m/s", fontsize=14)
plt.grid()

# Plot surface temperature minus 2 meter temperature
fig = plt.figure(figsize=(15,5))
plt.plot(aws4_tdiff)
plt.title("Surface Temperature minus 2 m Temperature", fontsize=14)
plt.ylabel("Degrees C", fontsize=14)
plt.grid()

# Plot wind speed vs temperature difference
fig = plt.figure(figsize=(5,5))
plt.scatter(aws4_ws10m, aws4_tdiff)
plt.xlabel("10 m Wind Speed [m/s]", fontsize=14)
plt.ylabel("Surface Temperature minus 2 m Temperature [C]", fontsize=14)
plt.grid()

# Plot sensible heat flux vs temperature difference
fig = plt.figure(figsize=(5,5))
plt.scatter(aws4_SHF, aws4_tdiff)
plt.xlabel("Sensible Heat Flux [W/m^2]", fontsize=14)
plt.ylabel("Surface Temperature minus 2 m Temperature [C]", fontsize=14)
plt.grid()
```

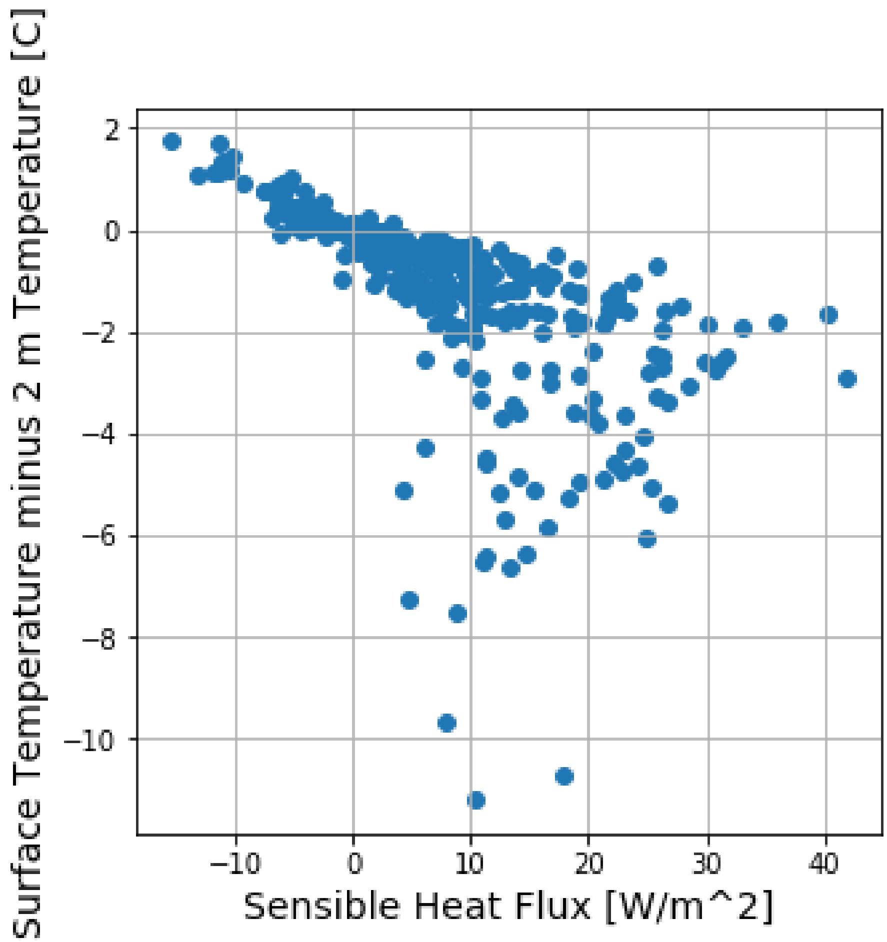



10 m Wind Speed
m/s
0
5
10
15
20
1998-01
1998-03
1998-05
1998-07
1998-09
1998-11
1999-01


Surface Temperature minus 2 m Temperature
Degrees C
2
0
−2
−4
−6
−8
−10
1998-01
1998-03
1998-05
1998-07
1998-09
1998-11
1999-01



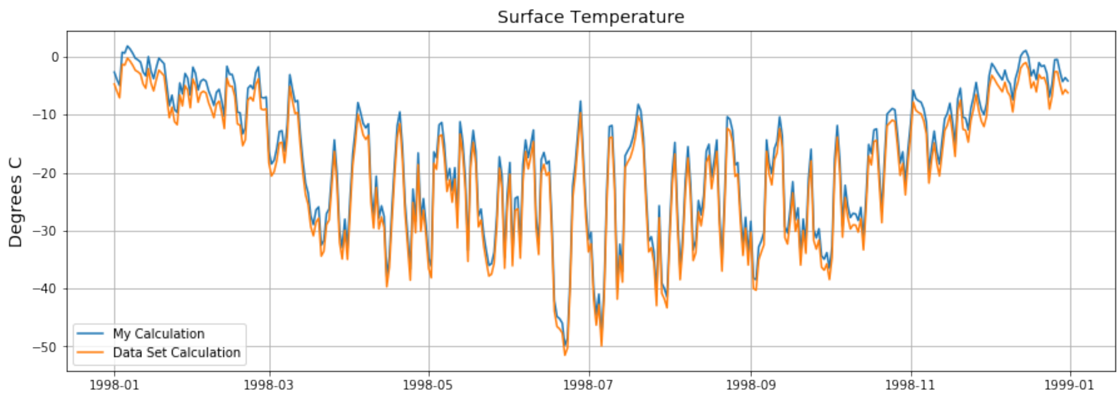
Surface Temperature minus 2 m Temperature [C]
2
0
−2
−4
−6
−8
−10
0
5
10
15
20
10 m Wind Speed [m/s]



4 Problem 3 - Calculate Surface Temperature at AWS 4

```
In [4]: # Calculate surface temperature 14
        aws4_OLWR = aws_timeseries(aws4_path, 14)
        aws4_ts_calc = np.power(aws4_OLWR / (0.97 * 5.67e-8), 0.25) - 273.15

        # Plot
        fig = plt.figure(figsize=(15,5))
        plt.plot(aws4_ts_calc, label='My Calculation')
        plt.plot(aws4_ts, label='Data Set Calculation')
        plt.title("Surface Temperature", fontsize=14)
        plt.ylabel("Degrees C", fontsize=14)
        plt.grid()
        plt.legend()
```

```
Out[4]: <matplotlib.legend.Legend at 0x1278b22e8>
```



5 Problem 4 - Radiative and Turbulent Fluxes

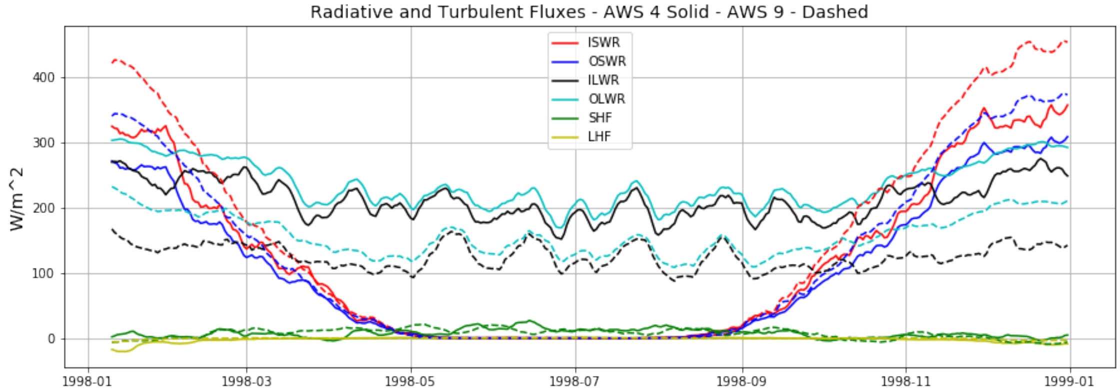
4.a. Seasonal Cycles: Radiative fluxes have a strong seasonal cycle, higher in the summer, lower in the winter due to the rising sun and warming of the atmosphere and snow surface. The sensible heat flux is also higher in winter. The ground heat flux appears to have no signficant seasonal cycle. Latent heat flux is either very small or zero in winter (no phasse changes) but then negative in summer. This can likely be explained by refreezing of surface meltwater or deposition.

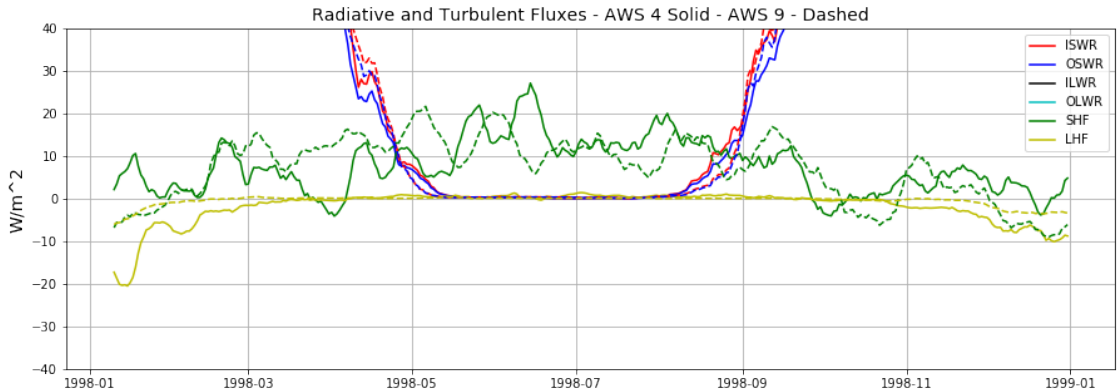
4.b. Similarities and Differences: Both sites have a strong seasonal cycle in shortwave radiation because of their high latitude, however AWS 9 has significantly higher incoming SWR because of higher elevation, despite a more southerly latitude. AWS 4 has higher longwave fluxes because of a warmer climate regime. In summer AWS 4 has a larger SHF because of a warmer atmosphere, however, the SHF can be positive in winter as both sites, indicating the atmosphere can be warmer than the surface (surface inversion) at both locations. The LHF is larger in magnitude at AWS 4 because of either larger meltwater refreezing or deposition.

```
In [5]: # Load data and perform 10 day running means
        aws4_ISWR = aws_timeseries(aws4_path, 11).rolling(window=10).mean()
        aws4_OSWR = aws_timeseries(aws4_path, 12).rolling(window=10).mean()
        aws4_ILWR = aws_timeseries(aws4_path, 13).rolling(window=10).mean()
        aws4_OLWR = aws_timeseries(aws4_path, 14).rolling(window=10).mean()
        aws4_SHF = aws_timeseries(aws4_path, 15).rolling(window=10).mean()
        aws4_LHF = aws_timeseries(aws4_path, 16).rolling(window=10).mean()
        aws4_GHF = aws_timeseries(aws4_path, 17).rolling(window=10).mean()

        aws9_ISWR = aws_timeseries(aws9_path, 11).rolling(window=10).mean()
        aws9_OSWR = aws_timeseries(aws9_path, 12).rolling(window=10).mean()
        aws9_ILWR = aws_timeseries(aws9_path, 13).rolling(window=10).mean()
        aws9_OLWR = aws_timeseries(aws9_path, 14).rolling(window=10).mean()
        aws9_SHF = aws_timeseries(aws9_path, 15).rolling(window=10).mean()
        aws9_LHF = aws_timeseries(aws9_path, 16).rolling(window=10).mean()
```

```
aws9_GHF = aws_timeseries(aws9_path, 17).rolling(window=10).mean()

# Plot AWS 4
fig = plt.figure(figsize=(15,5))
plt.plot(aws4_ISWR, 'r', label="ISWR")
plt.plot(aws4_OSWR, 'b', label="OSWR")
plt.plot(aws4_ILWR, 'k', label="ILWR")
plt.plot(aws4_OLWR, 'c', label="OLWR")
plt.plot(aws4_SHF, 'g', label="SHF")
plt.plot(aws4_LHF, 'y', label="LHF")

# Plot AWS 9
plt.plot(aws9_ISWR, 'r--')
plt.plot(aws9_OSWR, 'b--')
plt.plot(aws9_ILWR, 'k--')
plt.plot(aws9_OLWR, 'c--')
plt.plot(aws9_SHF, 'g--')
plt.plot(aws9_LHF, 'y--')
plt.legend()
plt.grid()
plt.title("Radiative and Turbulent Fluxes - AWS 4 Solid - AWS 9 - Dashed", fontsize=14
plt.ylabel("W/m^2", fontsize=14)

# Plot AWS 9 zoom on on turbylent fluxes
fig = plt.figure(figsize=(15,5))
plt.plot(aws4_ISWR, 'r', label="ISWR")
plt.plot(aws4_OSWR, 'b', label="OSWR")
plt.plot(aws4_ILWR, 'k', label="ILWR")
plt.plot(aws4_OLWR, 'c', label="OLWR")
plt.plot(aws4_SHF, 'g', label="SHF")
plt.plot(aws4_LHF, 'y', label="LHF")

# Plot AWS 9
plt.plot(aws9_ISWR, 'r--')
plt.plot(aws9_OSWR, 'b--')
plt.plot(aws9_ILWR, 'k--')
plt.plot(aws9_OLWR, 'c--')
plt.plot(aws9_SHF, 'g--')
plt.plot(aws9_LHF, 'y--')
plt.legend()
plt.grid()
plt.ylim([-40, 40])
plt.title("Radiative and Turbulent Fluxes - AWS 4 Solid - AWS 9 - Dashed", fontsize=14
plt.ylabel("W/m^2", fontsize=14)
```

```
Out[5]: Text(0, 0.5, 'W/m^2')
```





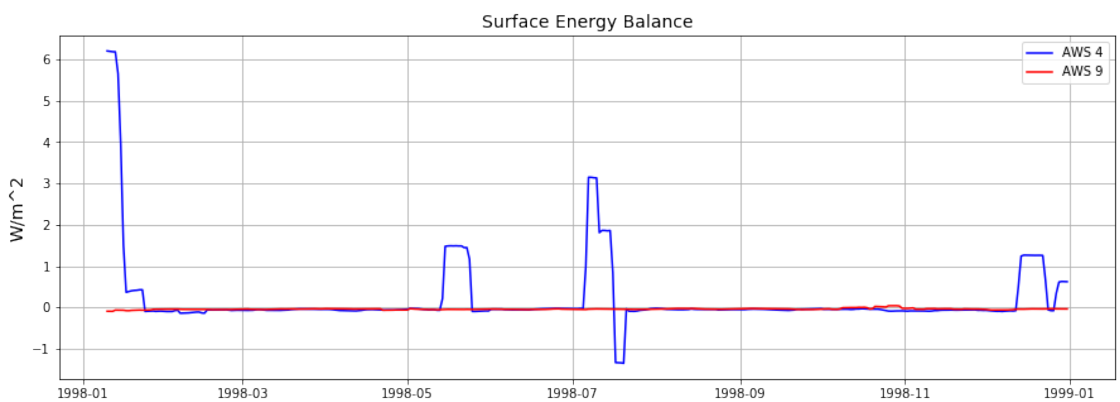
6 Problem 5 - Surface Energy Balance

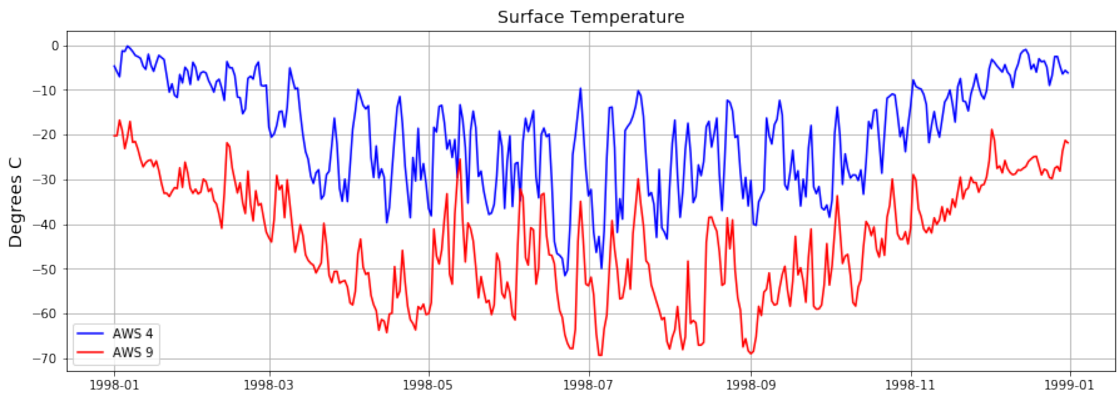
The surface energy balance is typically zero meaning that the surface is in thermal equilibrium. Surface energy balance is positive on some days because the outgoing longwave cannot keep up with the other fluxes, snow cannot heat up past zero degrees! On days with a positive SEB, the excess energy goes to melt or sublimation. I'm very surprised to see sustained positive SEB in winter at AWS 4. I think this is likely some sort of processing or instrument error as air temperature are well below zero durring these times. Or could a positive SEB be attributed to sublimation of surface snow?

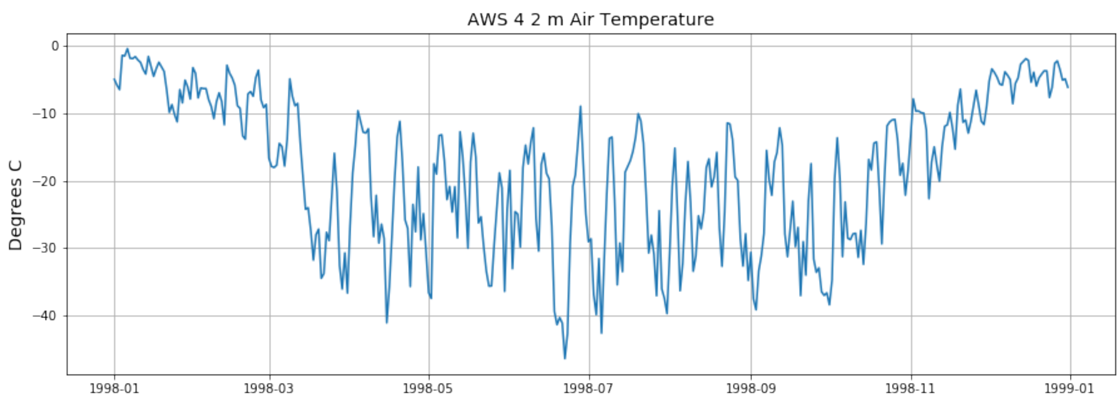
In [6]:
```
# Caclulate SEB
aws4_SEB = aws4_ISWR - aws4_OSWR + aws4_ILWR - aws4_OLWR + aws4_SHF + aws4_LHF + aws4_(
aws9_SEB = aws9_ISWR - aws9_OSWR + aws9_ILWR - aws9_OLWR + aws9_SHF + aws9_LHF + aws9_(

# Plot SEB
fig = plt.figure(figsize=(15,5))
plt.plot(aws4_SEB, 'b', label="AWS 4")
plt.plot(aws9_SEB, 'r', label="AWS 9")
```

```
plt.grid()
plt.legend()
plt.title("Surface Energy Balance", fontsize=14)
plt.ylabel("W/m^2", fontsize=14)

# Plot surface temperature
aws9_ts = aws_timeseries(aws9_path, 5) - 273.15
fig = plt.figure(figsize=(15,5))
plt.plot(aws4_ts, 'b', label="AWS 4")
plt.plot(aws9_ts, 'r', label="AWS 9")
plt.grid()
plt.legend()
plt.title("Surface Temperature", fontsize=14)
plt.ylabel("Degrees C", fontsize=14)

# Plot air temperature at AWS 4
fig = plt.figure(figsize=(15,5))
plt.plot(aws4_t2m)
plt.grid()
plt.title("AWS 4 2 m Air Temperature", fontsize=14)
plt.ylabel("Degrees C", fontsize=14)
```

Out[6]: Text(0, 0.5, 'Degrees C')







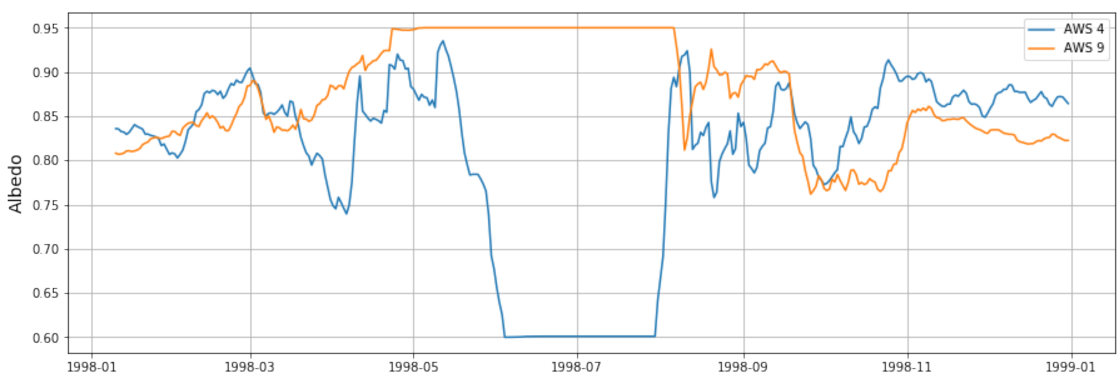
7 Problem 6 - Albedo

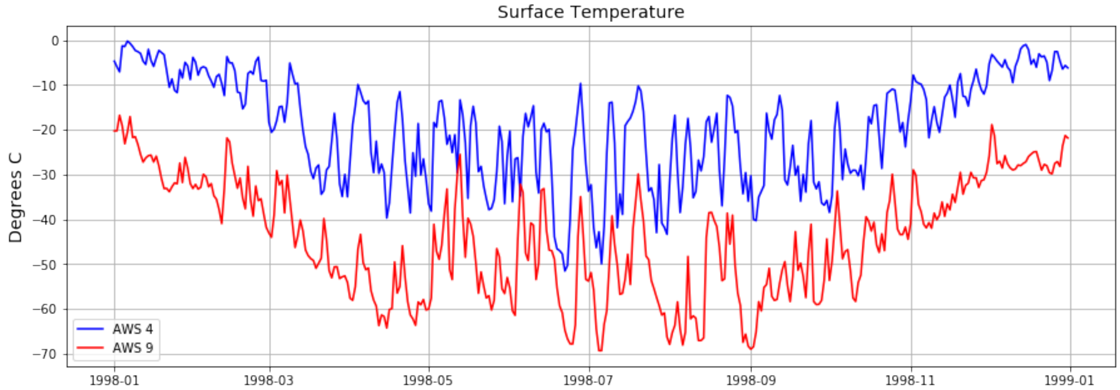
Note that I am plotting 10 day running means. For this analysis I focus on high summer (defined as January, February, November, and December) as this is when surface melting is most likely to occur, also becuase albedo is poorly defined in winter (dividing by zero). AWS 9 tends to have lower surface albedo than AWS 4. I actually find this a bit surprising given the higher solar zenith angles at AWS 9 and that surface melt at AWS 4 is much more likely than at AWS 9. However, perhaps more frequent fresh snowfall and therefore smaller snow grains could explain the generally higher surface albedo at AWS 4. Noteblly AWS 4 has more albedo variability, perhaps because more frequent cycles of surface melt and new snowfall. Interestingly, at both stations albedo decreases moving closer to the middle of summer. I wonder if this is because of melt or becuase lower solar zenith angles. You can see that periods of surface melting at AWS 4 correspond to reductions in surface albedo, whereas AWS 9 does not have surface melting

```
In [7]: # Load data
        aws4_ISWR = aws_timeseries(aws4_path, 11).rolling(window=10).mean()
        aws4_OSWR = aws_timeseries(aws4_path, 12).rolling(window=10).mean()
```

```
aws9_ISWR = aws_timeseries(aws9_path, 11).rolling(window=10).mean()
aws9_OSWR = aws_timeseries(aws9_path, 12).rolling(window=10).mean()

# Calculate albedo
aws4_albedo = aws4_OSWR / aws4_ISWR
aws9_albedo = aws9_OSWR / aws9_ISWR

# Plot
fig = plt.figure(figsize=(15,5))
plt.plot(aws4_albedo, label="AWS 4")
plt.plot(aws9_albedo, label="AWS 9")
plt.ylabel("Albedo", fontsize=14)
plt.grid()
plt.legend()

# Plot surface temperature
fig = plt.figure(figsize=(15,5))
plt.plot(aws4_ts, 'b', label="AWS 4")
plt.plot(aws9_ts, 'r', label="AWS 9")
plt.grid()
plt.legend()
plt.title("Surface Temperature", fontsize=14)
plt.ylabel("Degrees C", fontsize=14)
```

```
Out[7]: Text(0, 0.5, 'Degrees C')
```





Surface Temperature
Degrees C
0
−10
−20
−30
−40
−50
−60
−70
1998-01
1998-03
1998-05
1998-07
1998-09
1998-11
1999-01
AWS 4
AWS 9