Sea Ice Parameters in Antarctica: Seasonal variations in the Sea-ice Thickness

Shreya Trivedi
mailto:shreyatrivedi26@ucla.edu

March 2, 2020

Abstract

Antarctic sea ice plays an important role in regulating the heat, moisture and moisture exchanges between the atmosphere and the ocean, and salinity of the ocean. Current studies of Antarctica confirm the area-integrated total sea ice extent grew to record maximum values in four of the last six years, whilst the 2016-17 summer has been marked by record low ice cover. The reasons for the variations in the sea ice extent are not fully understood. Given the important role of Antarctic sea ice; in this era of climate change it is vital to understand the factors contributing to its variability. Current work focuses on the sea ice concentration, which allows an understanding of the areal variation of the ice. The sea ice thickness (SIT) which allows an understanding of the variability of the volume of sea ice has not been explored to the same extent because there was a lack of data. This study uses a newly-released circum-Antarctic SIT dataset to examine the variability in Antarctic SIT and to investigate the relationships between the SIT and climatic variables such as surface and ocean temperatures as well as convergence and divergence in ocean flows over a period of 2002-2011. This study gives a first look at the spatial and temporal variability of the SIT showing how it varies between the continent and sea-ice edge, with SIT favoring increase towards the landmass. Temporal variation shows maximum SIT values concentrated at the start of spring when the sea-ice accumulation is at its peak.

Keywords: *; Sea-ice Thickness ; Climate Change; Seasonal Variability*

Contents

List of Figures

1 Introduction

The Antarctic sea-ice plays is of global significance owing to its role in the global ocean's uptake of heat, and potentially in the penetration of warm, ocean water to the base of Antarctic ice shelves. This makes the study of Antarctic sea ice and its spatial and temporal variations important, primarily due to two reasons, firstly, there are very limited studies which have been conducted in the research field focusing mainly upon the reasons and the climatological causalities behind the changing sea-ice dynamics in Antarctica. Secondly, to study the tele-connections which might impact the regional and global climatic interactions as a result of variations taking place in Southern Oceans. In todays's scenario where Climate Change is considered as the 'prime suspect' for changing climatological dynamics, it becomes all the more important to asses the reasons for the sea-ice variability in order to ascertain whether the observed changes are due to natural variability alone, or represent a forced anthropogenic response.

2 Methodology and Datasets

2.1 Dataset

The dataset format used in the study is called the NetCDF format (with the extension .nc). The Network Common Data Form, or netCDF, is an interface to a library of data access functions for storing and retrieving data in the form of arrays. An array is an n-dimensional (where n is 0, 1, 2, ...) rectangular structure containing items which all have the same data type (e.g., 8-bit character, 32-bit integer). A scalar (simple single value) is a 0-dimensional array.

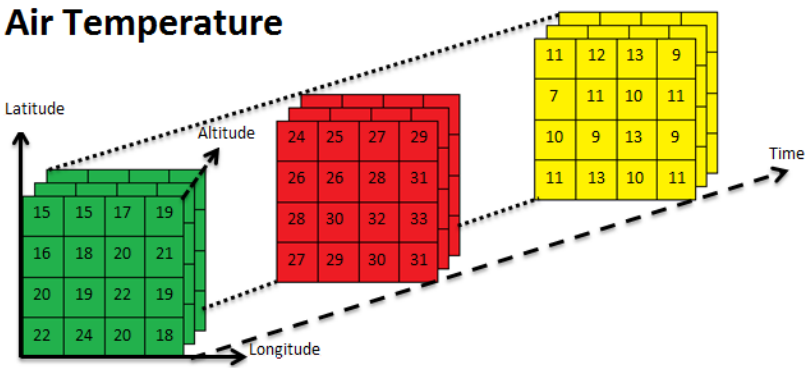

Figure 1: NetCDF format: Reading a climate variable

The study uses gridded datasets for making seasonal comparisons of the Sea-ice Thickness(SIT) values around Antarctica. A surface air temperature

dataset is also used to derive relationships between sea-ice parameters and global temperatures.
The dataset is in the form of the spatially gridded values over the time period of 2002-2011. The dataset only incorporates two seasons namely, Spring and Winters (the months of May, June, July, August, September and October)

2.2 Python Scripting

Python can read and analyse NetCDF files via two inbuilt packages namely *netCDF4 and xarray*. These packages make it very handy to iterate through this 3-D gridded files and and read the values located at every coordinate. (intersection of the latitude and longitude for sliced for every time-step.)

```

import netCDF4 as nf # This package helps to read the 3-D gridded
    (.nc) format.
import datetime as dt #This helps to convert the time-steps in the
    netcdf files into readble date format
import numpy as np
import matplotlib.pyplot as plt #For plotting spatial variables
from mpl_toolkits.basemap import Basemap #Provides basemaps to be
    used as reference backgrounds for the
#plotting the globally referenced climate variables over space and
    time
```

Listing 1: Importing required packages

- **Reading and Iterating through the variables:**

  Both *xarray and netcdf* packages help in reading the gridded files by converting them into arrays or dictionaries respectively. The functions in the package help in reading the variables and extracting them to be used individually.

```

data_temp = nf.Dataset("air.mon.mean_1979-2015.nc") #importing
    datasets using netcdf

#defining the variables using the ncdf4 library
time_temp = data_temp.variables["time"]
lon = data_temp.variables["lon"]
lat = data_temp.variables["lat"]
level = data_temp.variables["level"]
air = data_temp.variables["air"]


#Creating the time stamps in the form of real time date format:

time_start_temp = nf.date2index(dt.datetime(1979,1,1),time_temp,
    select="nearest")
time_end_temp = nf.date2index(dt.datetime(2015,12,31),time_temp,
    select="nearest")
```

```
dates= nf.num2date(time_temp[:],time_temp.units)
time= ([date.strftime('%Y-%m-%d') for date in dates[:]])
```

Listing 2: Reading variables in NetCDF

- **Data Slicing: Selection of desired time or pressure levels:**

  Selecting only the months of May, June, July, August, September and October and the years from 2002-2011. The filtering of the dates is done so that comparison can be made with the another NetCDF file which incorporates the sea-ice values for the above timescale.

  Since the NetCDF files contain more than one pressure levels, desired pressure level was extracted from the matrix which was relevant for the study. The pressure level at **index 2** is 850 mb which is the near surface pressure level. The variables in a NetCDF files are read in the format *current shape = 444(time), 17(levels), 73(lat), 144(long).*

```
#selecting the months:
new_mon_temp=[]
for ii in range(len(time)):
    if time[ii][5:7]=='05'or time[ii][5:7]=='06' or time[ii][5:7]==
    '07' or time[ii][5:7]=='08' or time[ii][5:7]=='09' or time[ii
    ][5:7]=='10':
        new_mon_temp.append(time[ii])


#selecting the years:
new_time_temp=[]
for ii in range(len(new_mon_temp)):
    if new_mon_temp[ii][0:4]=='2002'or new_mon_temp[ii][0:4]=='2003
    ' or new_mon_temp[ii][0:4]=='2004' or new_mon_temp[ii][0:4]=='
    2005' or new_mon_temp[ii][0:4]=='2006' or new_mon_temp[ii
    ][0:4]=='2007' or new_mon_temp[ii][0:4]=='2008' or new_mon_temp
    [ii][0:4]=='2009' or new_mon_temp[ii][0:4]=='2010' or
    new_mon_temp[ii][0:4]=='2011':
        new_time_temp.append(new_mon_temp[ii])


#selecting the 850mb pressure level:
air_temp_850hpa=air_temp[:,2,:,:]
```

Listing 3: Data Slicing (using temperature dataset)

- **Statistical operations on Gridded Variables:**

  The study required the use of temporal trends of the data for comparing variations in a variable over a specified period of time. For this purpose, functions were defined that iterated through the lat-lon matrices corresponding to every time-step and calculated an area-averaged value against each time-step in the data. These are called as *Area-Averaged Values.*

```
#Mean:

def matrix_mean(timestep_matrix):
    temp_mean=[]
    for ii in range(0,len(timestep_matrix)):
        mean = np.mean(timestep_matrix[ii])
        temp_mean.append(mean)
    return(temp_mean)

#Standard deviation:

def matrix_std(timestep_matrix):
    temp_std=[]
    for ii in range(0,len(timestep_matrix)):
        std = np.std(timestep_matrix[ii])
        temp_std.append(std)
    return(temp_std)

#Maximum values:

def matrix_max(timestep_matrix):
    temp_max=[]
    for ii in range(0,len(timestep_matrix)):
        maxm = np.max(timestep_matrix[ii])
        temp_max.append(maxm)
    return(temp_max)

#Minimum values:

def matrix_min(timestep_matrix):
    temp_min=[]
    for ii in range(0,len(timestep_matrix)):
        minm = np.min(timestep_matrix[ii])
        temp_min.append(minm)
    return(temp_min)
```

Listing 4: Defining functions for operating on gridded variables

- **Creating Visualizations using NetCDF:** In order to create trend lines for the gridded variables, outputs from the above statistical functions were used and then plotted using the library *matplotlib*.

```

def visualize(timestep_matrix):

    #1) Finding mean, max, min and standard deviation
    max_global = matrix_max(timestep_matrix)
    min_global = matrix_min(timestep_matrix)
    mean_global = matrix_mean(timestep_matrix)
    std_global = matrix_std(timestep_matrix)

    #2) create our three subplot holders
    fig = matplotlib.pyplot.figure(figsize=(20.0,10.0))
    axes1= fig.add_subplot(2,2,1)
    axes2= fig.add_subplot(2,2,2)
    axes3= fig.add_subplot(2,2,3)
```

```
    axes4= fig.add_subplot(2,2,4)

    #3) plot each of them
    fig.suptitle('Global Temperature Trends and Statistics:
    1979-2015', fontsize=25)
    axes1.set_ylabel('Average')
    axes1.plot(mean_global)

    axes2.set_ylabel('Maximum')
    axes2.plot(max_global)

    axes3.set_ylabel('Minimum')
    axes3.plot(min_global)
    axes3.set_xlabel('No. of Timesteps')

    axes4.set_ylabel('Standard Deviation')
    axes4.plot(std_global)
    axes4.set_xlabel('No. of Timesteps')

    #4) we want to show the plot
    fig.tight_layout
    matplotlib.pyplot.show()
```

Listing 5: Defining functions for creating Visualizations

The above code created following output:

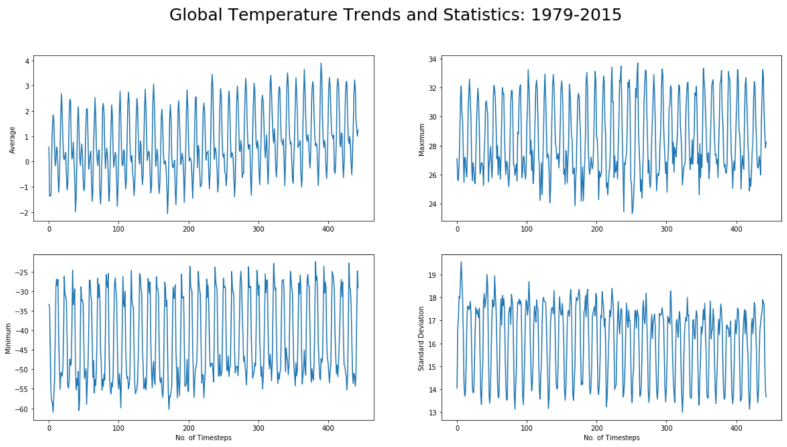

Figure 2: Visualizing statistics for a Gridded data (Air Temperature)

3 Results

3.1 Temporal trends: Relationship between Air Temperature and Sea-ice Thickness (SIT)

Studying Antarctic sea-ice thickness variability is the primary objective of the study. Besides looking at the variability trends in the dataset, the research would also try to explore the inter-relationship between the sea-ice parameters and other climatic parameters. This would help in building greater understanding of the factors controlling the changes in the sea-ice patterns across seasons and years.

A temporal trend analysis was conducted to look at the variations of the sea-ice thickness with the surface temperatures at 850 hpa.

```
#One mean value obtained for each matrix (mesh of latitude and
    longitude) for every timestep in the gridded data.

#mean Sea Ice Thickness (SIT) array

SIT_mean=[]
for ii in range(len(SIT)):
    mean_SIT = np.nanmean(SIT[ii])#This is done to deal with the
    very high NaN values in the SIT dataset.
    SIT_mean.append(mean_SIT)
print(SIT_mean)

#mean temperature array

temp_mean=[]
for ii in range(len(air_temp_850hpa)):
    mean = np.mean(air_temp_850hpa[ii])
    temp_mean.append(mean)
print(temp_mean)


# Plotting mean values on the same plot for comparisons:

fig = plt.figure(figsize=(16.0,10.0))
plt.plot(time,SIT_mean, marker='o', markerfacecolor='blue',
    markersize=10, color='skyblue', linewidth=4,label='SIT')
plt.plot(time,temp_mean, marker='', color='red', linewidth=2,
    linestyle='dashed', label="Temperatures")
plt.xticks(np.arange(0, 55, 4.0)) #axis intervals defined
plt.xlabel('Dates') #these are dates with a well defined date
    format
plt.ylabel('Mean values')
plt.title('Global Temperatures and SIT trends:Spring and Winter
    Seasons (2002-11)')
plt.legend(loc='best') #It chooses the best positioning for legend
plt.show()
```

Listing 6: Comparing two gridded variables

The above code created following output:

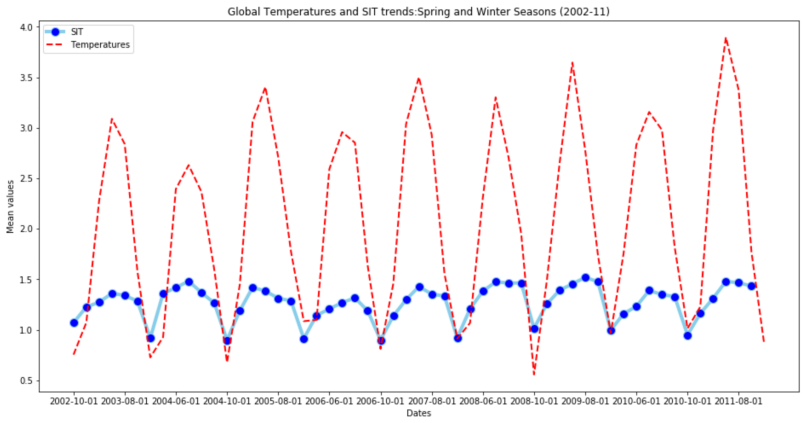

Figure 3: Trend Plot comparing SIT and air temperatures at 850hpa)

3.2 Spatial analysis: Studying spatial extents and variations of SIT

Along with temporal trends or change in the SIT values with time, it is very important to note the spatial variations of SIT for a particular time period. The spatial variations also help in understanding the changes in the extent and thickness of the sea-ice parameters during a particular season. Hence, Spatial Variability studies are very important for studying seasonal patterns.
While dealing with spatial plots, another crucial point which plays an important role in the visualization is the use of relevant projections for the maps. In our case, **Polar Projections** are the best for representing the South Pole. For this purpose, the *Basemap feature* is used for all spatial plots.

```
#Defining a function for creating Spatial Plots

def plt_map(data):
    m = Basemap(projection='spstere',boundinglat=-55,lon_0=90,
    resolution='l') #Creates a basemap with south pole oriented
    projection with the domain extending from 55S to 90S.
    x, y = m(lon, lat) #Creates a mesh out of the lat-lon matrix
    fig = plt.figure(figsize=(15,7))
    m.fillcontinents(color='white',lake_color='gray') #specifics
    for the basemap aesthetic
    m.drawcoastlines()
    m.drawparallels(np.arange(-80.,81.,20.)) #spacing and extent of
     latitudes
    m.drawmeridians(np.arange(-180.,181.,20.)) #spacing and extent
    of longitudes
    m.drawmapboundary(fill_color='skyblue')
    m.contourf(x,y,data,40,cmap=plt.cm.get_cmap('inferno'));#Using
    the fill countor method of plotting
    plt.title("Seasonal Variations in SIT for October,2002")
    plt.colorbar()
```

```
15
16 #Plotting the spatial map of SIT for the first timestep at index 0:
17 plt_map(SIT[0])
```

Listing 7: Creating Spatial Plots

The above code created following output:

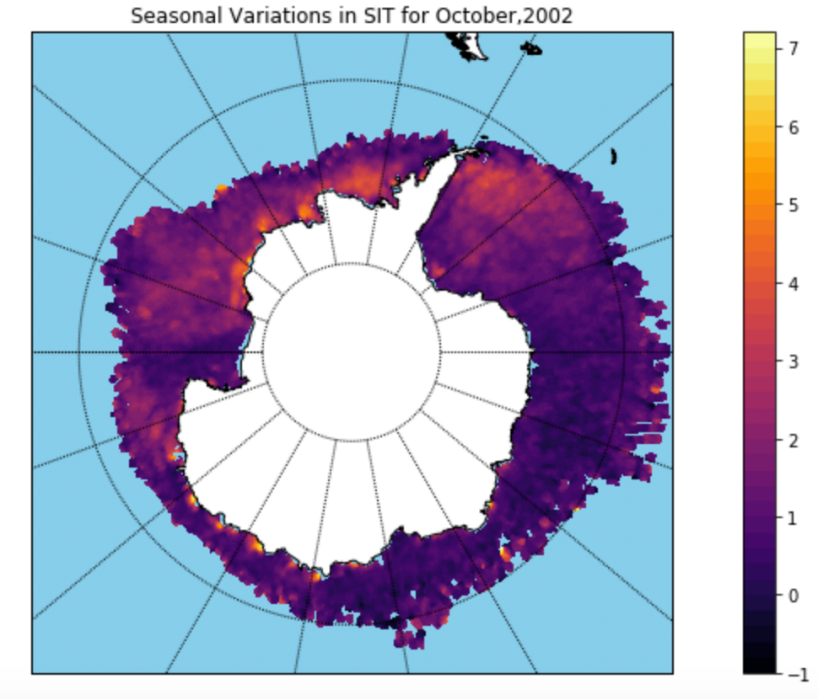

Figure 4: Spatial Variations in SIT for October, 2002

4 Discussion and Conclusions

Several analyses are yet to be done in order to obtain the desired results leading to a well-defined conclusion.