README

Nitay Ben-Shachar

December 16, 2019

This package solves a 1D transient thermodynamic model for the temperature profile of sea ice/snow, based primarily on the air temperatures, which can be solved to arbitrary resolution both spatially and temporally, and has been coded in MATLAB. Snow can be incorporated but is assumed to have constant physical parameters. The growth of sea ice is determined by balancing heat fluxes at the ice-ocean interface. No oceanic heat flux is added but a custom ocean heat flux can be easily incorporated. A sharp interface between the ice and ocean is assumed. Salt transport is accounted for only in its variation of the ice's physical parameters.

The model has been validated using temperature array data taken in McMurdo Sounds, Antarctica in 1997 and has achieved an agreement to within 0.3 ^{o}C with the entire thermistor array when some oceanic heat flux was accounted for. More details are provided in the attached report.

This program was coded and tested with MATLAB R2019a. Note that the report might have slight discrepancies in the naming of files or other small details.

1 Model Details

The model used here is a 1D transient thermodynamical model. The goal of the model is to obtain a solution to the temperature profile in the sea ice and snow, and hence being able to estimate the ice growth rate and thickness over time, primarily based on the air temperatures. As we are modelling temperature transients, the core of the model is the heat equation which is solved throughout the sea ice/snow to determine the evolution of the temperature profiles. The heat equation is solved by making use of the Method of Lines and the mathematical technique of freezing the boundary, a very appropriate name in this situation. Both techniques are briefly detailed below, but are described in more details in the attached report.

The Method of Lines is the process of discritizing over space, meaning that some level of resolution must be chosen for the model. Rather than trying to solve for the temperature profile of the sea ice/snow at every depth, we consider a discrete number of points at various depths in the ice/snow and assume that the temperature profile is smooth between them, a relatively good assumption (as long as the resolution is high enough) as the heat equation is diffusive, smoothing the temperature gradients out. The spatial resolution of the model can be chosen to be arbitrarily high/low, tho discretizing at 5cm intervals is usually sufficiently accurate.

Freezing the Boundary is a mathematical tool that has been incorporated in this model for mathematical ease and elegance. This technique is especially useful when some solution is being obtained over an interval with a moving boundary, as we have in our situation, with the variable snow thickness and growing sea ice. Rather than adding simulation calculation points as the ice grows, we solve an equivalent mathematical problem over a fixed depth interval by defining the variables $\chi, \xi = \frac{z}{H(t)}$ where $H(t)$ is the ice/snow depth. By definition, the slab of sea ice always lies in corresponding values of χ, ξ between 0 and 1, and so solving the (modified) heat equation in terms of χ, ξ takes into account any sea ice growth or snow dump/melt. More mathematical details are provided in the report.

Some limitations of the model include:

- Salt transport is not properly treated, but rather a fixed salinity profile provided by the user is used (usually attained via ice cores). This decision was made as properly modelling salt transport cannot be done with a 1D model as brine channels connect and funnel the salt in a 3D manner. Brine affects the thermal properties of the sea ice as salt is a major contributor to the variability in its properties, see A.

- The thermal conductivity has a high uncertainty and has high dependence on salinity. The formula presented in appendix A from [4] is based on a model that fits the bulk of the data points but with some significant scatter.

- Snow density and hence its thermal properties vary greatly with age, wind, temperature, precipitation etc. For example, snow compresses greatly with age, changing its insulating powers. This is not taken into consideration due its high variability and rather an average value is used.

- The ice/snow to atmosphere interaction is not modelled rigorously, but rather the top layer of the ice (or snow if present) is assumed to be at the air temperature. This also means that we neglect any solar radiation terms (which can penetrate deep into the ice [2]) and any possibility of snow-ice forming (melted snow freezing into ice on top of the ice slab [1]).

- The ice/ocean interface is assumed to be a sharp interface and is not assumed to vary in location (due to the one dimensionality of the model).

2 Using the Package

In order to run the program, three files must be provided in the input folder. These files are:

salinity.csv: This file contains the salinity profile of the sea ice. The first column should be the depth (in meters) corresponding to the salinity (in grams per gram) in the second column. The depths should all be negative numbers, corresponding to depths below the sea level. If the salinity profile is not known leave this file blank and the salinity will default to 5ppt.

snow.csv: This file contains the snow thickness over time. The first column should contain the time in days corresponding to the snow thickness in the right column. The snow thicknesses are then interpolated linearly between the provided values.

airTemps.csv: This file contains the air temperatures over time. Provide the time in days in the first column and the corresponding air temperature in degrees C in the second column.

As well as the above files, some parameters and initial conditions should be adjusted in the file seaIceModel.m. These values are clearly labeled near the top of the file.

num_of_points_snow/ice: These constants determine the number of calculation points that are used in discritizing over the depth of the snow/ice for the simulation. A constant number of points is used meaning that the resolution of the model changes over time, for example, setting num_of_points_ice = 10 for a simulation were the ice slab grows from -0.2m to -0.5m means that the model's resolution changes from 50points/meter to 20 points/meter. If the output temperature profiles seem unrealistic, for example with sharp changes, a likely solution would be to increase the spatial resolution of the simulation by increasing these constants.

odeset: This constant determines the accuracy that the ODE solver takes, so determines the temporal accuracy of the simulation. The MATLAB ODE solvers are used to determine the numerical solutions to the temperature transients of the sea ice/snow and choose a time resolution that corresponds to an arbitrarily accurate solution specified by odeset. 'RelTol' determines the rerror allowed for each variable with respect to that variable at each calculation cycle. 'AbsTol' determines the absolute numerical error allowed for each variable. See the MATLAB documentation for further settings that could be selected.

final_t: This is the number of seconds that the solver should run for. Select the duration for which the solution is desired. Time being zero is set to the first line in each input file (airTemps.csv, snow.csv).

time_offset: The solver will begin the solution from the point time_offset, with t=0 being defined at the beginning of the input files.

initial_depth: This is the initial depth of the ice, at the point in time specified by time_offset. This value should be negative and measured in meters.

present_dt: This is the temporal resolution (in seconds) in which the solutions will be reported.

z_res_ice: This is the spatial resolution (in meters) in which the sea ice temperature profile solutions will be reported.

z_res_snow: This is the spatial resolution (in meters) in which the snow temperature profile solutions will be reported.

Note: Increasing present_dt, z_res_ice and z_res_snow does not increase the accuracy of the solutions, only the resolution in which the solutions are reported. To increase the accuracy increase the number of calculation points in the model and decrease the error tolerance (odeset).

T_freezing: This is the freezing temperature of the sea water, taken to be constant. This can be adjusted if the salinity of the water is well known. Further, it can be extended into a subroutine that adjusts the freezing point depending on the salinity, depth or whatever the other variables of interest are.

V_a: This is the proportion of incorporated air in the freezing sea ice, taken to be constant. Again, this can be made to depend on any number of parameters via a subroutine.

Once all the necessary variables have been provided, with the MATLAB working directory set to the folder containing the file seaIceModel.m, run the simulation by typing model_output = seaIceModel() into the MATLAB command line. The solutions will be stored in the variable model_output as detailed here:

model_output.time: A 1D array containing the values of time (in seconds), starting from 0 corresponding to the time of time_offset, at which the remainder of the solutions are given. This temporal resolution can be adjusted with the constant present_dt.

model_output.z_ice: A 1D array containing the depths associated with the values presented in the temperature profile of ice.

model_output.z_snow: A 1D array containing the depths associated with the values presented in the temperature profile of snow.

model_output.temp_profile_snow: A 2D array containing the solution to the temperature profile in the snow as a function of time. The temperature profile at a specific time is given by temp_profile_snow(time_index, :) where time_index is an integer corresponding to the time of interest. The temperature profile is calculated at time increments of present_dt and so time_index=$\frac{time}{present\_dt}$. Points in the air are filled with a dummy value of 0.

model_output.snow_depth: A 1D array containing the snow thickness over the desired times. This will merely be an interpolation of the provided snow thickness at the points of interest.

model_output.temp_profile_ice: A 2D array with the corresponding solution of the ice temperature profile at the times of interest, similar to temp_profile_snow. Points in the water are filled with a dummy value of the freezing temperature.

model_output.ice_depth: A 1D array containing the solution to the ice depth over time.

3 Files

seaIceModel.m This is the central function that administrates the rest of the simulation. It has three primary purposes being initialising all the necessary variables, obtaining a solution with the user provided data and finally re-organizing the solution in the necessary format.

3.1 input

This is the folder under which the user should place the necessary data needed for the simulation. Three files should be placed here: salinity.csv, snow.csv and airTemps.csv. Each should be formatted as detailed above in section 2.

3.2 user

In this folder are the subroutines associated with handling the user data.

calcAirTemp.m This function returns the air temperature at a specific time by interpolating between the user specified air temperatures. The request time should be measured in seconds and begins at zero at the first time specified in the airTemps.csv. If time_offset was initialised to anything but zero this is accounted for.

calcSalinity.m This function returns the salinity at a given depth, handling the input from salinity.csv, by interpolating on the salinities given in salinity.csv.

snowThickness.m This function returns the snow thicknesses as a given time by interpolating between the user specified snow thicknesses in snow.csv.

snowGrowthRate.m This function returns the growth rate of the snow at a particular time by linearly interpolating between the snow thicknesses provided and returning the slope of the linear interpolation it lies on. This is used in the iteration of the variables when solving for the temperature profile of the snow in equation 30 of the report.

3.3 params

This folder contains the functions that return the physical parameters of the sea ice/snow based on the local conditions (such as temperature, salinity etc.). The formulas are all provided in appendix A along with references.

3.4 utils

iceGrowthConstant.m This function returns the constant in formula 23 of the report. This constant is given by k/(L rho).

importUserData.m This function imports the data entered in the 'input' folder and saves it in matrices ready to be used. It also preforms some preliminary checks to ensure the data was entered in the correct format.

surfaceTemp.m This function returns the temperature between the ice and the snow in order to balance the heat flows and ensure the interface is in thermal equilibrium. This is to ensure equation 24 of the report is satisfied.

iterateDEs.m This is the function that returns the rate of change of the temperature profiles and ice depth for a given temperature profile of the snow and ice. This is the function passed to the MATLAB ODE solver. This function finds the rate of change of each point in the snow/ice profile via the heat equation (equation 30 in the report). The ice growth rate is determined with the boundary condition given in equation 23 in the report.

4 Troubleshooting

4.1 Unable to reduce time step below minimum amount

If a MATLAB warning that the minimum time step was hit while trying to solve the ODEs, the resolution of the simulation must be increased by increasing the number of calculation points num_of_points_ice/snow.

4.2 Sharp jump in the snow temperature profile

This is due to the solver completely jumping over a critical time where the snow has been changing rapidly. This can be fixed by reducing the error tolerance or adding a minimum time step being a fraction of the length of the rapid snow melt/dump. See the MATLAB documentation for odeset for possible ways of increasing the solver accuracy and ensuring that it doesn't skip the points of interest.

A Physical Parameters and Constants

All quantities are given in SI units. Temperature is measured in degrees Celsius. Salinity is given in units of grams per gram, note that this is a factor of 1000 smaller than other standard units such as psu, ppt (parts per thousand). Each of the following quantities is calculated in its own subroutine in the MATLAB folder.

–have to make sure these are all correct, like k ice has changed–

A.1 Ice Properties

- Thermal Conductivity [4] [W/(m K)]:

$$k_{ice} = \frac{\rho}{\rho_{ice}} \left(2.11 - 0.011T + 0.09\frac{S}{T} - \frac{\rho - \rho_{ice}}{1000} \right)$$

where ρ_{ice} is the density of pure ice taken to be 917 kg/m^3 as in [].

- Heat Capacity [3] [J/(kg K)]:

$$c_{ice} = 1000 \left(2.113 + 0.0075T - 0.0034S \times 1000 + 0.00008ST \times 1000 + 18.04 \frac{S \times 1000}{T^2} \right)$$

- Density [5] [kg/(m^3)]

$$\rho_{ice} = (1 - V_a) \left(1 - \frac{4.51S}{T} \right) 917$$

- Latent Heat of Fusion [5] [J/kg]:

$$L_{ice} = 4184 \left(79.68 - 0.505T - 27.3S + 4311.5 \frac{S}{T} \right)$$

A.2 Snow Properties

- Density [2] [kg/m^3]: $\rho_{snow} = 330$
- Thermal Conductivity [5] [W/(m K)]:

$$k_{snow} = 0.0688 \exp \left(0.0088T + 4.6682 \frac{\rho_{snow}}{1000} \right)$$

- Heat Capacity [5] [J/(kg K)]:

$$c_{snow} = (2.7442 + 0.1282(T + 273.15)) \times \frac{18.02}{1000}$$

References

[1] Matti Lepparanta. "A review of analytical models of sea-ice growth". In: *Atmosphere-Ocean* 31.1 (1993), pp. 123–138. URL: https://doi.org/10.1080/07055900.1993.9649465.

[2] Gary A. Maykut and Norbert Untersteiner. "Some results from a time-dependent thermodynamic model of sea ice". In: *Journal of Geophysical Research (1896-1977)* 76.6 (1971), pp. 1550–1575. DOI: 10.1029/JC076i006p01550. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/JC076i006p01550.

[3] D. J. Pringle, H. J. Trodahl, and T. G. Haskell. "Direct measurement of sea ice thermal conductivity: No surface reduction". In: *Journal of Geophysical Research: Oceans* 111.C5 (2006). DOI: 10.1029/2005JC002990. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2005JC002990.

[4] D. J. Pringle et al. "Thermal conductivity of landfast Antarctic and Arctic sea ice". In: *Journal of Geophysical Research: Oceans* 112.C4 (2007). DOI: 10.1029/2006JC003641. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2006JC003641.

[5] Yin-Chao Yen. "Review of Thermal Properties of Snow, Ice and Sea Ice". In: (June 1981). URL: https://apps.dtic.mil/dtic/tr/fulltext/u2/a103734.pdf.