Project # 1
Build your own thermodynamic sea ice model
LPHYS2265 2023

Martin Vancoppenolle*

March 5, 2023

In this exercise, you will build a thermodynamic sea ice model incrementally. In the end, your model should be able to simulate a full seasonal cycle of sea ice growth and melt[1].

What to deliver ?

We ask you to deliver 1) the model code routine in the language of your choice, 2) a very short, 1-page report (French is fine) with a brief description of your model (be brief – do not include equations, just say in a few lines which processes you have incorporated) and one 4-panel figure with the daily evolution of (a) ice thickness, (b) snow depth, (c) surface temperature and (d) water temperature, over 50 years, with oceanic heat flux = 5 W/m^2 and albedo = 0.8. The questions within the exercise are there to help you to develop understanding of your model.

Some general remarks.

- Use your **favourite computing language** (matlab, python, fortran, r, ...). In this tutorial, I assume you use matlab, but you can easily translate

*martin.vancoppenolle@locean.ipsl.fr

[1]This exercise was put together by Pr. Dirk Notz (Hamburg University) and I, for a sea ice class the University Centre in Svalbard, Longyearbyen, Norway.

the functions I provide into any other language. Contact me for any translation issue.

- **Use separate scripts** for the different parts of the exercise. That way you can reuse your code later, and correct your mistakes.
- **Code clearly**. Do one thing per line, not more. Give sensible **names** to your variables (not too short, not too long). Avoid single letters (except maybe for loop indices). Make sure your lines of code read as quickly as possible. If your code reads easily, errors become less likely.
- **Use comments**. That way, if you leave work aside for a week, you can easily dive into it again. Teachers/mates can also help you more easily.
- **References**. The formulas and understanding you will need can be found in slides of Chapter #7 (ice thermodynamics). You can also pick up information from the paper of Semtner (1976).
- Feel free to collaborate, but make sure you understand what you are doing :-).

First thing, open a script, give it a nice name, and start by defining the constants you will need:

Listing 1: Script Preamble

```
clear all; close all

% Physical constants
Lfus = 3.35e5; % Latent heat of fusion for water [J/kg]
rhoi = 917.;   % Sea ice density [kg/m3]
ki = 2.2;      % Sea ice thermal conductivity [W/m/K]
ks = 0.31;     % Snow thermal conductivity [W/m/K]
sec_per_day = 86400.; % Seconds in one day [s/day]

% Bottom boundary condition
T_bo = -1.8;   % Bottom temperature [C]
```

1 Ice growth with constant temperature (*1-2h*)

It is January, it is cold and dark, the air temperature remains at $-10°C$ for 30 days. The sea has started freezing.

1.1 Stefan and numerical solution for ice growth

Let us assume no ocean heat flux for a start. Define an array of ice thickness for the 30 days for which we want to calculate the ice-thickness evolution ($h_i = zeros(30, 1)$).

Set the initial ice thickness to 0.1 m, i.e. $h_i(1) = 0.1$. You should then use a do-loop to calculate the ice thickness for days 2 to 30. Inside your do-loop, calculate first the sensible heat flux in the ice Q_c based on the thickness of the previous day $h_i(day - 1)$. Then calculate the thickness change that this heat flux can cause within one day.

Add this thickness change to the thickness from the day before, and you have the ice thickness for the current day, $h_i(day)$. Then you go back to the beginning of the for loop to calculate the thickness on the next day etc.

1. Plot your ice thickness evolution for these 30 days
2. How thick does your ice get in the end?
3. How thick should the ice be after 30 days according to Stefan's law (Slide # 8)?

1.2 Add an ocean heat flux

Now assume that there is some heat flux coming from the water. Add some oceanic heat flux into your loop and see how it affects the ice growth.

1. Plot your ice thickness evolution over 30 days for an oceanic heat flux of $5 W/m^2$.
2. How thick does your ice get in the end for an oceanic heat flux of $5 W/m^2$?
3. How thick does your ice get in the end for an oceanic heat flux of $180.4 W/m^2$? Can you explain why?
4. Set your ice thickness to 1 m on day 1. Which oceanic heat flux do you need from the ocean to keep the ice at this thickness?

1.3 Add snow

Let's now assume that we had some snow fall on the ice in December. Our initial ice thickness is again 10 cm, our initial snow thickness is 5 cm and doesn't change. Modify your code so that it can handle the impact of snow for the calculation of the heat flux in the ice.

1. How thick does the ice get now if you don't have any oceanic heat flux?
2. Which heat flux do you need to keep the ice thickness constant at 1 m?
3. Which heat flux would you need without snow ?
4. What are the answers to these questions if your snow is 15 cm thick?

Now you have a little break and think about what you learnt... or something else!

2 Freeing surface temperature (*2-3h*)

2.1 Surface heat fluxes

Now we'll free the surface temperature. To do so, you will need to be able to specify the atmospheric heat fluxes. You are lucky, I did all the boring work and converted the Fletcher (1965) heat fluxes into SI units and did two regressions for the solar and the sum of non-solar (downwelling longwave, latent and sensible) heat fluxes (Q_{sol}, Q_{nsol}), respectively. So, just put the lines below at the end of your script.

Listing 2: Surface heat flux functions

```
function [Q_sol] = solar_flux(day)
  Q_sol = 314. * exp(-(day-164).^2/4608.);
end

function [Q_nsol] = non_solar_flux(day)
Q_nsol = 118. * exp(-0.5 * (day-206.).^2 / (53^2)) +
    179.;
end
```

You will also need a new set of physical constants

Listing 3: Extra script preamble

```
% Physical constants
...
epsilon = 0.99     % surface emissivity
sigma   = 5.67e-8 % Stefan-Boltzmann constant
Kelvin  = 273.15 % Conversion from Celsius to Kelvin
...
alb = 0.8;       % surface albedo
```

1. Make sure that you can use such functions by plotting the result when day goes from 1-365 and that you retrieve values close to Slide #15. If you are not confident in the fits, you can check that they match the compilation in Table 1, page 383 of the paper of Semtner (1976).

2.2 Calculate surface temperature

Assume we have sea ice with a thickness of h = 0.5 m, surface albedo $\alpha = 0.8$ and surface emissivity $\epsilon = 0.99$. The bottom temperature of the ice remains, as before, at the freezing point of sea water with a salinity of 34 g/kg, $T_f = -1.8°$ C.

The surface temperature changes according to the different fluxes from the atmosphere. The incoming atmospheric fluxes and the outgoing longwave flux from the ice surface are balanced by the conduction flux in the ice. We assume that the temperature gradient in the ice is linear all the time, and we get

$$Q_{sol}(1-\alpha) + Q_{nsol} - \epsilon\sigma T_{su}^4|_K = k \cdot \frac{T_{su} - T_f}{h} \tag{1}$$

where T_{su} is the surface temperature of the ice and all temperatures are measured in Kelvin. The sign convention is that all downward fluxes are positive.

One important remark about units (an often unexpected source of trouble). The easiest is to use Kelvin from now on, in all calculations, to avoid confusion. In particular, make sure you define bottom temperature in K.

Then, using a do loop for days between 1 and 365, calculate the surface temperature for each day of the year. Transform Equation (1) into a standard 4-th order polynomial (aT4 +bT3 +cT2 +dT +e = 0) and use Newton Raphson method (or roots function of matlab) to solve for T_{su}. If you use the built-in function (roots), you will get four roots, some of which are complex. You should take the maximum of the real part of the roots.

1. Plot the surface temperature for each day of the year.
2. The temperature is not physically sensible for ice in summer, because we do not incorporate the heat used by surface melting. Cap surface temperature to 273.15 K using an "if" or, probably more consise, a "max".

2.3 Couple temperature and thickness

2.3.1 Bottom growth and melt only

The big merge happens now. You can now copy the code that you used to calculate surface temperature into the code you wrote for exercise 1.3 that calculated the evolution of ice thickness. Now your temperature feels the effect of changing thickness, and your conductive heat flux feels the effect of changing temperature.

Set the initial ice thickness to 10 cm, initial snow thickness to 0 cm and oceanic heat flux to 2 W/m^2.

1. How does your ice thickness change throughout the year?

2. **Calendar**. Modify your code so as to calculate the evolution of ice thickness over, say, 100 years, with the same repeated forcing. One key trick is to calculate day of year, depending on the total number of days (N_d):

```
doy = mod(0:N_d-1,365)+1
```

   because the flux functions need a day between 1 and 365 and give idiot results otherwise.

2.3.2 Surface melting

Now your little code has basal growth and melt coupled to varying surface temperature. However, you still have no surface melt.

Set the number of years of simulation back to 1. Now you will calculate the energy available for melting. In practise, this happens when the surface temperature is 273.15 K or above. The net surface flux ($Q_{sol}(1-\alpha)+Q_{nsol}-LW^{\uparrow}-k(T_{su}-T_f)/h$) will give zero most of the year, and will be above zero when the surface temperature is 273.15K. Use this energy to calculate surface melting (an extra term of tendency for thickness).

- Plot the evolution of ice thickness throughout one year. What happens?

- Once you are happy, run 100 years of simulation.

- You can now play around with changing, for example, the emissivity, the albedo, the oceanic heat flux, etc... and see how your ice thickness evolves in time.

At the end of this exercise, your model is able to simulate **multi-year sea ice**, without snow. Take a break :-)

3 Add surface ocean and snow (2-3h)

3.1 Add surface ocean

Set your model into ice-free conditions. Try to make your model such that ice totally melts in summer in the first year of simulation. To do so, set $Q_w = 5$ W/m^2 and α=0.6. Once you achieve such a state, you should see your that your ice thickness becomes negative.

As you now have seen, results are no longer realistic results in such a regime: the ice thickness becomes 0 or negative. To overcome that issue, one has to add an oceanic mixed layer to the model, characterized by a temperature T_w. Let us initialize the mixed layer temperature at the freezing temperature (=T_b). As long as the ice cover is present, T_w does not change.

Initial warming. Once the ice thickness becomes negative, the ice thickness should be reset to 0. The heat that is used to do so should instead be used to heat the ocean. Assume a depth of the mixed layer of $h_w = 50$ m, a heat capacity of the water of 4000 J / (kg K), an albedo of the water of 0.1 and a water density of 1025 kg/m^3. T_w will now increase once the ice disappears.

Water temperature calculation. The next problem you will encounter is that your model crashes, because you try to calculate ice growth and melt, which does not work if $h_i = 0$. What you will do is to split your do loop in two conditional blocks. One block will correspond to ice-covered conditions (the block you already have), and another block of code will be dedicated to ice-free conditions. In the latter, you will recompute the surface energy budget over the ocean (accounting for the ocean albedo) and use this heat to change T_w.

Refreezing. The final problem you will encounter occurs at the end of the ice-free season. Nothing happens and the water temperature can decrease indefinitely. Instead, once the water temperature drops below freezing, reset it to freezing temperature, use the excess heat to grow some new ice.

1. Let the model run for 10 years. How thick does your ice get in winter? Are there still year-to-year changes?

2. When does the ocean become ice free?

3. By how much do you have to reduce the non-solar flux to get multi-year ice?

4. By how much do you have to increase the non-solar fluxes to have an ice-free Arctic all year round?

Now, your model is able to simulate **first-year and multi-year sea ice regimes**, but still without snow.

3.2 Add snow [optional!]

Adding snow is the final step to reach what Semtner proposed in his paper as a minimum set of physics for sea ice thermodynamics. In the present exercise, we leave it as optional. Yet, we have seen that snow significantly reduces the ice-growth rate and therefore, incorporating snow will make your model more realistic.

To consider snow, add a **snow depth** array h_s. We will consider two terms in the snow mass budget equation: snowfall and melting. Set your albedo to 0.8 and oceanic heat flux to 5 W/m^2 to set your model into a multi-year regime which is easier to handle.

For **snowfall**, we will follow the approach that Maykut and Untersteiner (1971) used for their model. They assumed total 30 cm of snow fall between August 20 and October, 5 cm of total snowfall between November and April, and 5 cm snowfall in May. You can use a function similar to heat fluxes to compute snowfall.

Regarding **snow melt**, we will do as follows. As long as you have snow on the ice, once the surface temperature reaches 0°C, you first have to melt the snow before you can melt the ice. Do not forget that snow is lighter than sea ice, so it costs less energy to melt it. The density of the snow is 330 kg/m^3 on average.

One small but annoying **bug** may arise once snow and ocean are active together. When the ice that refreezes right after the ice-free period is very thin and has snow on top of it, the conduction flux can become too small and re-melt the ice immediately. One way to overcome this is to impose a minimum ice thickness value (10 cm) in the calculation of the heat conduction flux. This problem could reflect that the time step we use is too large. This could also reflect wrong model physics. For example, that heat is not fully conserved in our model, or that heat transfer when ice is thin is wrongly represented. I don't really know exactly[2].

Finally, if you feel like, you could even add **snow-ice** formation. To do so, calculate if the snow-ice interface is below sea level and turn the snow below the sea level into so-called snow-ice (simply remove it from the snow thickness and add it to the ice thickness).

1. Investigate how your model responds to doubled snow-fall rate.

4 Epilogue

You now have a model that can represent first-year and multi-year ice cycles. You might be a bit puzzled by the fact that model strongly responds to forcing and albedo. We will further explore that in the second part of the exercise.

[2]Note that such a situation is typical of the world of earth system models. Often, some bugs arise when the complexity of the model increases. The inconsistencies become sometimes influential. Modelers have to figure out how to resolve those. This type of choice is subjective and is part of what can be qualified as model uncertainty.

References

Semtner, A. J. (1976), A model for the thermodynamic growth of sea ice in numerical investigations of climate, *Journal of Physical Oceanography*, *6*, 379–389.