

# An Introduction to Climate Modeling

## Milestone 2 - Albedo, Solar Forcing and Visualization in Time

### 1 Compute Albedo and Solar Forcing Terms and Visualize in Time

In milestone 2 you will program the calculation of all necessary parameters for the calculation of the time and space dependent solar forcing and then visualize it as an animation over a period of one year in 48 time steps. For this you will need the functions `read_geography` and `robinson_projection` from milestone 1.

1. Write a function `calc_albedo` which outputs the albedo  $\alpha_{ij}$  for the points  $(\theta_i, \varphi_j)$  as a matrix  $A \in \mathbb{R}^{65 \times 128}$  depending on the earth surface type  $T \in \mathbb{N}^{65 \times 128}$  (see milestone 1). For the calculation use the albedo scheme below as proposed by Zhuang et al. <sup>1</sup>.

Land mask	Albedo
Land	$0.30 + 0.12P_2(x)$
Ocean	$0.29 + 0.12P_2(x)$
Sea ice	0.60
Snow	0.75

$P_2(x) = \frac{(3x^2-1)}{2}$  for  $x = \sin(\theta)$ , where  $\theta$  is the latitude in radians

2. In the table below, besides atmospheric parameters, constants are given for each type of the earth's surface, which are used for the calculation of the effective heat capacity. The effective heat capacity of a grid cell results from the sum of the heat capacity of the respective surface type and the heat capacity of the overlying atmospheric column. Since the climate model measures time in years, this heat capacity has to be divided by the number of seconds in a year  $3.15576 \times 10^7$  s. Implement a function `calc_heatcapacity` analogous to `calc_albedo` to calculate the geography-dependent heat capacity per surface area using the given constants from the table below.

Types	Density $\rho$ in $\text{kg m}^{-3}$	Specific heat capacity in $\text{J kg}^{-1}\text{K}^{-1}$	Depth in m
Atmosphere	1.225	1000	3850
Ocean mixed layer	1030	4000	70
Sea ice	917	2000	1.5
Land	1350	750	1.0
Snow	400	880	0.5

3. Implement a function `read_true_longitude` to read in the true longitude  $\lambda$  for each time step from the input file `True_Longitude.dat` and output it as a vector  $\lambda \in \mathbb{R}^{48}$ .
4. Write a function `insolation` that calculates the insolation  $S(\theta, t)$  as in Berger <sup>2</sup>. Input arguments of the function should be:

<sup>1</sup>K. Zhuang, G.R. North, M.J. Stevens, *A NetCDF version of the two-dimensional energy balance model based on the full multigrid algorithm*, SoftwareX, Vol. 6, pp. 198-202, July 7, 2017.

<sup>2</sup>A. L. Berger, *Long-Term Variations of Daily Insolation and Quaternary Climatic Changes*, Journal of the Atmospheric Sciences, Vol. 35, pp. 2362-2367, 1978.

- $\theta$ : latitude,
- $S_0$ : solar constant,
- $e$ : eccentricity,
- $\epsilon$ : obliquity,
- $\tilde{\omega}$ : precession distance,
- $\lambda$ : true longitude.

The insolation at latitude  $\theta$  and time  $t$  is given by:

$$S(\theta, t) = \begin{cases} 0 & \text{for } (\theta, t) \text{ without sunrise,} \\ S_0 \rho(t) \sin(\theta) \sin(\delta(t)) & \text{for } (\theta, t) \text{ without sunset,} \\ \frac{S_0 \rho(t)}{\pi} \left( H_0(t) \sin(\theta) \sin(\delta(t)) + \cos(\theta) \cos(\delta(t)) \sin(H_0(t)) \right) & \text{else,} \end{cases}$$

where  $\delta(t)$  is the declination of the sun with

$$\sin(\delta(t)) = \sin(\epsilon) \sin(\lambda(t)).$$

Here,  $\rho$  is the normalized distance between earth and sun squared,

$$\rho(t) = \left( \frac{1 - e \cos(\nu(t))}{1 - e^2} \right)^2,$$

with eccentricity  $e$  and the positional angle of the Earth on its orbit around the Sun  $\nu(t) = \lambda(t) - \tilde{\omega}$  as given counterclockwise from aphelion.

$$H_0(t) = \arccos(-\tan(\theta) \tan(\delta(t)))$$

describes the absolute angle of the sun at sunrise and sunset. Make sure that your implementation handles values  $\pm\infty$  of the tangent at the poles correctly.

5. Write a function `calc_solar_forcing` with input arguments

- albedo values at the grid points,
- $S_0$ : solar constant with default value  $S_0 = 1371.685$ ,
- $e$ : eccentricity with default value  $e = 0.01674$ ,
- $\epsilon$ : obliquity with default value  $\epsilon = 0.409253$ ,
- $\tilde{\omega}$ : precession distance with default value  $\tilde{\omega} = 1.783037$ ,
- $(\lambda_j)_{j=1}^{n_t} \in \mathbb{R}^{n_t}$ : true longitude for each time step.

Note that these default orbital parameters correspond to the year 1950. This function should calculate and return the solar forcing  $S_{sol}(\theta_i, \varphi_j, t_k)$  at each grid point  $(\theta_i, \varphi_j)$  and for each time step  $t_k$ . The return value should be a three-dimensional array, where the first dimension corresponds to the latitude, the second to the longitude, and the third to the time step. The solar forcing is given by

$$S_{sol}(\theta, \varphi, t) = S(\theta, t) \alpha_c(\theta, \varphi).$$

Here,  $S$  is the insolation defined above and  $\alpha_c$  is the coalbedo given by  $\alpha_c = 1 - \alpha$ , where  $\alpha$  is the albedo.

6. To visualize the solar forcing as an animation, calculate the albedo and the solar forcing for a grid size of  $65 \times 128$  (with the data file from milestone 1) and 48 time steps within a year. Then plot your result using the function `robinson_projection` from milestone 1.

## 2 Control Solutions

Surface Albedo of the Earth



Figure 1: Plot of the albedo

Surface Heat Capacity



Figure 2: Plot of the heat capacity

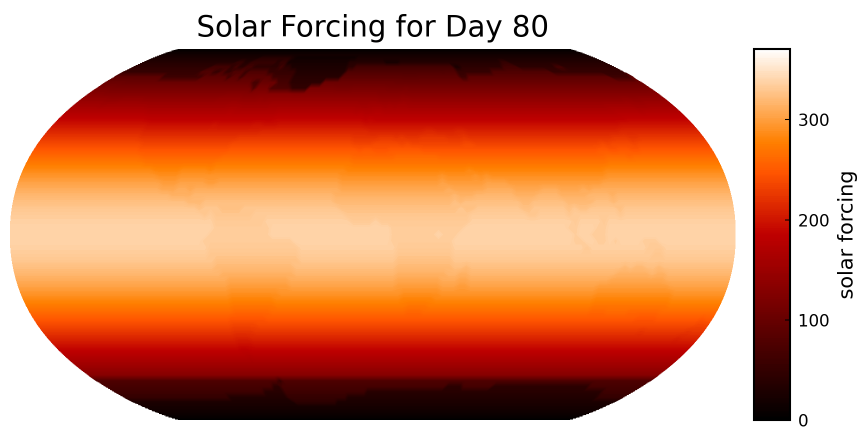


Figure 3: Example plot for the solar forcing on day 80 of the year