lowEBMs: A Python implementation of low-dimensional energy balance models

Application example: Late Holocene temperature response

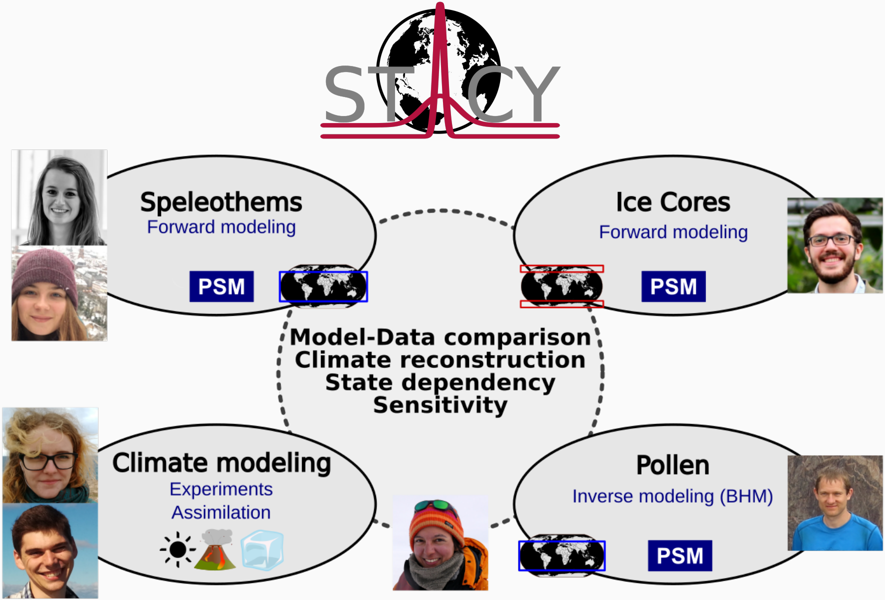
Benjamin Schmiedel

19.08.2019

Institute of Environmental Physics
University of Heidelberg

Geophysical Institute
University of Bergen

Motivation - STACY Group



Motivation - Bachelor Thesis

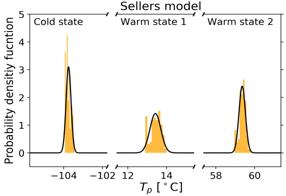
Climatic representation and simulation limits of two one-dimensional Energy-Balance Models

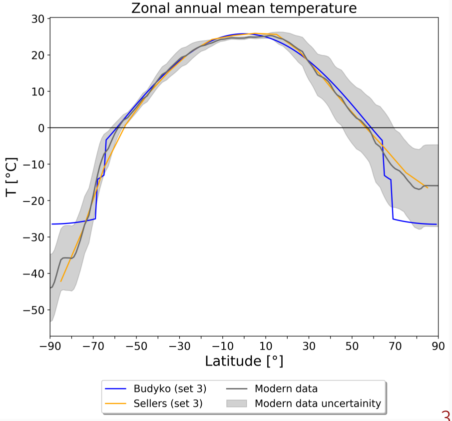
Bachelor Thesis in Physics
submitted by

Benjamin Schmiedel

born in Rheinfelden (Germany)

2019





Motivation - Development

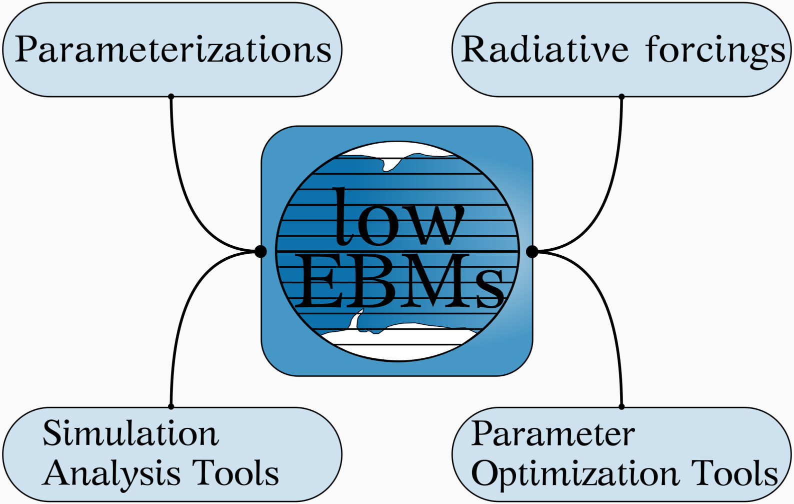
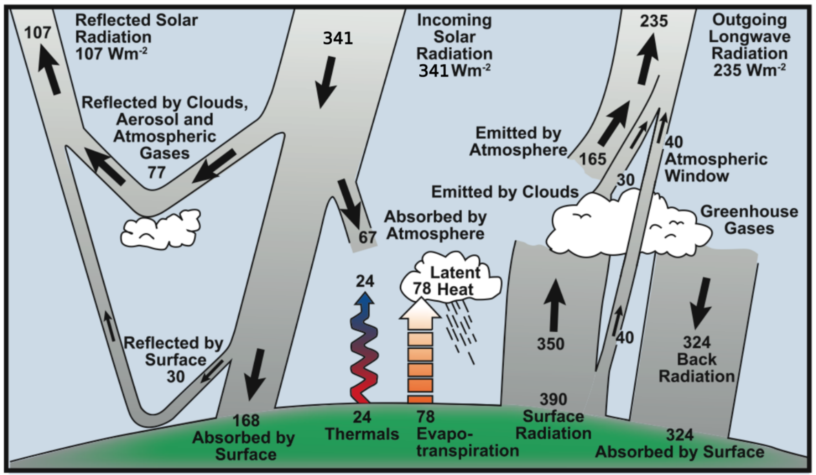


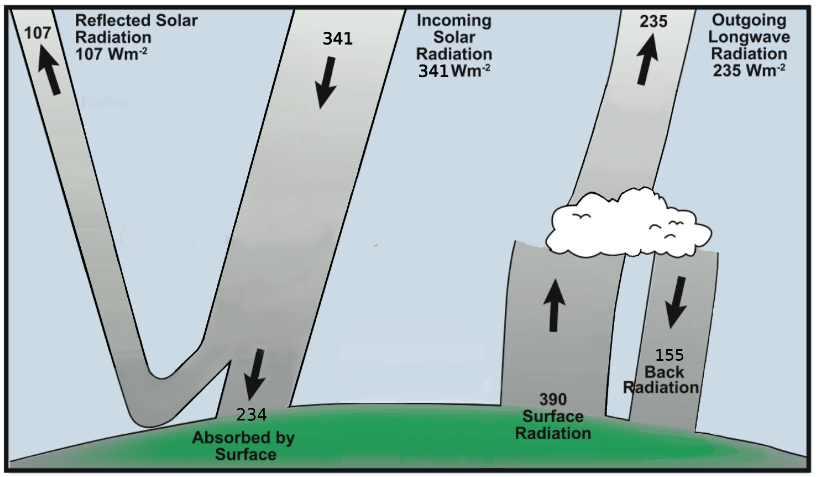
Table of Content

Theoretical background - Energy balance



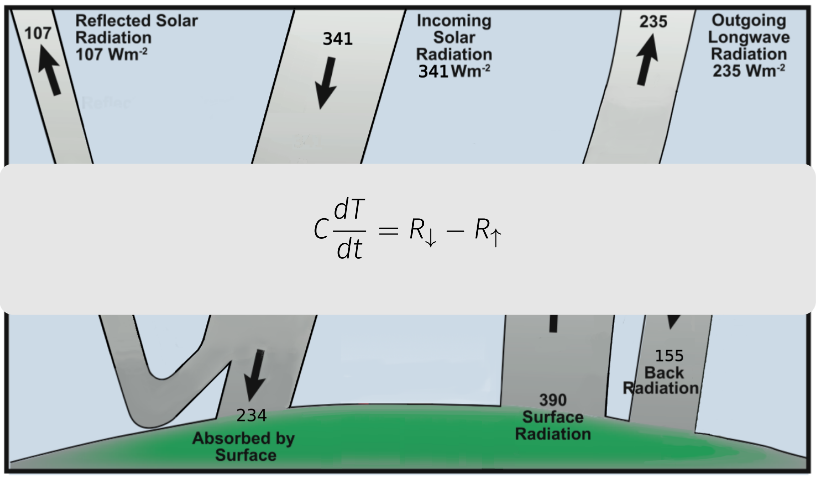
[Kiehl and Trenberth 1997]

Theoretical background - Energy balance



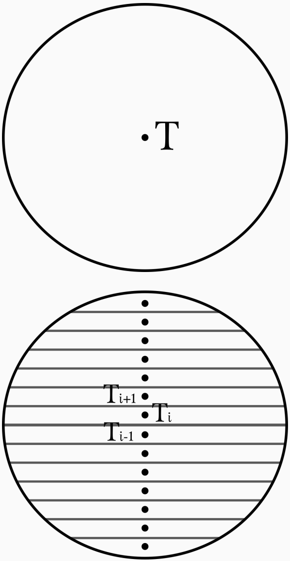
[Kiehl and Trenberth 1997, modified]

Theoretical background - Energy balance



$$C\frac{dT}{dt} = R_{\downarrow} - R_{\uparrow}$$

[Kiehl and Trenberth 1997, modified]



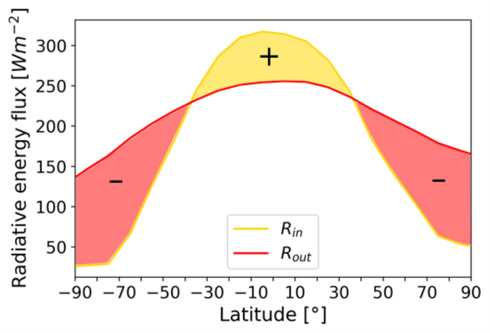
Zero dimensional

$$C\frac{dT}{dt} = R_{\downarrow} - R_{\uparrow}$$

One dimensional

$$C\frac{dT_i}{dt} = R_{\downarrow,i} - R_{\uparrow,i}$$

Theoretical Background - Latitudinal Transfer



$$C\frac{dT_i}{dt} = R_{\downarrow,i} - R_{\uparrow,i} + F_{transfer,i}$$

Theoretical Background - Parameterizations

$\boxed{R_\downarrow}$

$R_\downarrow^{(1)} = (1-\alpha) \cdot Q$

- $\alpha = const.$
- $\alpha(\phi)$
- $\alpha(T)$

$\boxed{R_\uparrow}$

$R_\uparrow^{(1)} = \epsilon\sigma \cdot T^4$

- Sellers
- Stefan-Boltzmann

$R_\uparrow^{(2)} = A + B \cdot T$

- NoClouds
- Clouds

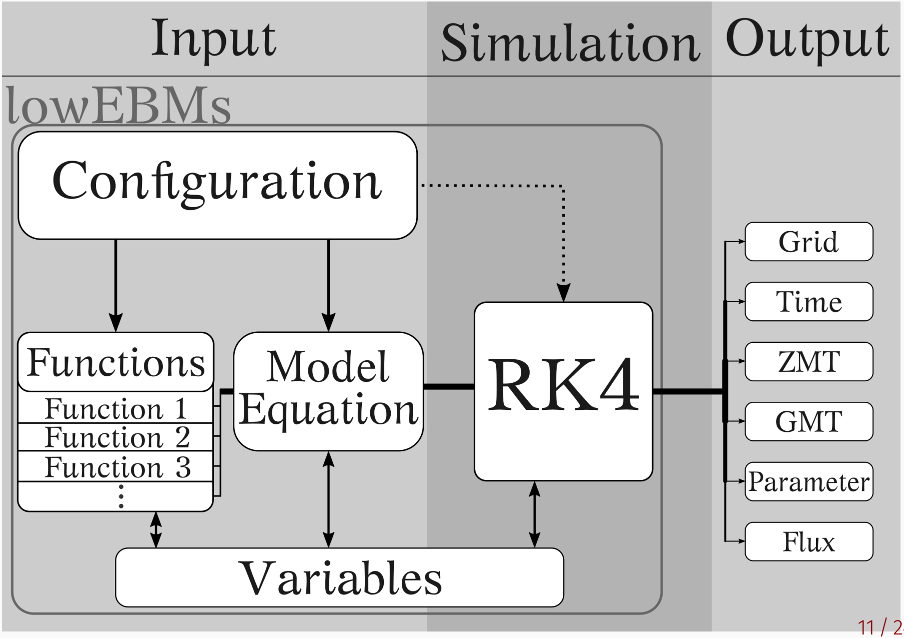
$\boxed{F_{transfer}}$

$F_{transfer}^{(1)} = \gamma(T(\phi) - T_0)$

$F_{transfer}^{(2)} = \Delta cL + \Delta C + \Delta O$

- ΔcL: water vapour
- ΔC: atmospheric
- ΔO: oceanic

Implementation details - workingmode



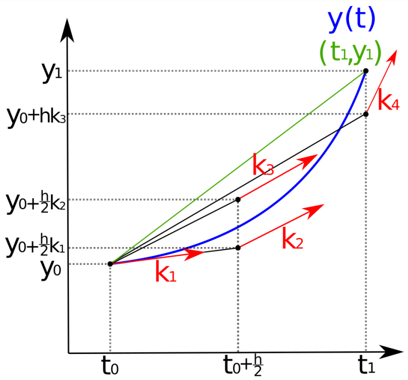
Implementation details - configuration

File Edit View Search Tools Documents Help

```
[rk4input]
number_of_integration=365*120
stepsize_of_integration=60*60*24
spatial_resolution=10
both_hemispheres=True
latitudinal_circle=False
latitudinal_belt=True

eq_condition=False
eq_condition_length=100
eq_condition_amplitude=1e-3

data_readout=10
number_of_externals=0

[initials]
time=1800*365*60*60*24
zmt=273+15
gmt=273+15
initial_temperature_cosine=True
initial_temperature_amplitude=30
initial_temperature_noise=False
initial_temperature_noise_amplitude=5

[func0]
func=flux_down.insolation
Q=1366.14
m=1
dQ=0

albedo=albedo.dynamic_sel
albedoread=False
albedoparam=[0,0]

noise=False
noiseamp=342*0.03
noisedelay=1
seed=True
seedmanipulation=0
```

- Input is seperated into sections
- Each sections variables define the simulation
- [rk4input] defines the numerical integration
- [initial] defines the initial condition
- [func0],..,[func4] define the parameterization

Implementation details -
Runge-Kutta 4th order scheme



$$k_1 = f(t_0, y_0)$$

$$k_2 = f(t_0 + \frac{h}{2}, y_0 + \frac{h}{2} \cdot k_1)$$

$$k_3 = f(t_0 + \frac{h}{2}, y_0 + \frac{h}{2} \cdot k_2)$$

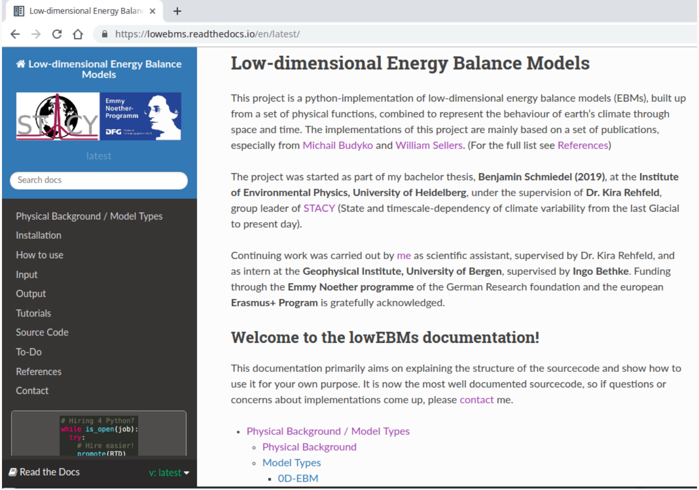
$$k_4 = f(t_1, y_0 + h \cdot k_3)$$

$$\Phi = \frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4$$

$$y_1 = y_0 + \Phi(f, t_0, y_0, h) \cdot h$$

- iteratively solve the model equation (with $f(t,y) \mathrel{\hat{=}} \frac{dT}{dt}$)

Implementation details - documentation


Low-dimensional Energy Balan

https://lowebms.readthedocs.io/en/latest/

Low-dimensional Energy Balance Models

latest

Search docs

Physical Background / Model Types
Installation
How to use
Input
Output
Tutorials
Source Code
To-Do
References
Contact

Read the Docs v: latest

Low-dimensional Energy Balance Models

This project is a python-implementation of low-dimensional energy balance models (EBMs), built up from a set of physical functions, combined to represent the behaviour of earth's climate through space and time. The implementations of this project are mainly based on a set of publications, especially from Michail Budyko and William Sellers. (For the full list see References)

The project was started as part of my bachelor thesis, **Benjamin Schmiedel (2019)**, at the **Institute of Environmental Physics, University of Heidelberg**, under the supervision of **Dr. Kira Rehfeld**, group leader of STACY (State and timescale-dependency of climate variability from the last Glacial to present day).

Continuing work was carried out by me as scientific assistant, supervised by Dr. Kira Rehfeld, and as intern at the **Geophysical Institute, University of Bergen**, supervised by **Ingo Bethke**. Funding through the **Emmy Noether programme** of the German Research foundation and the european **Erasmus+ Program** is gratefully acknowledged.

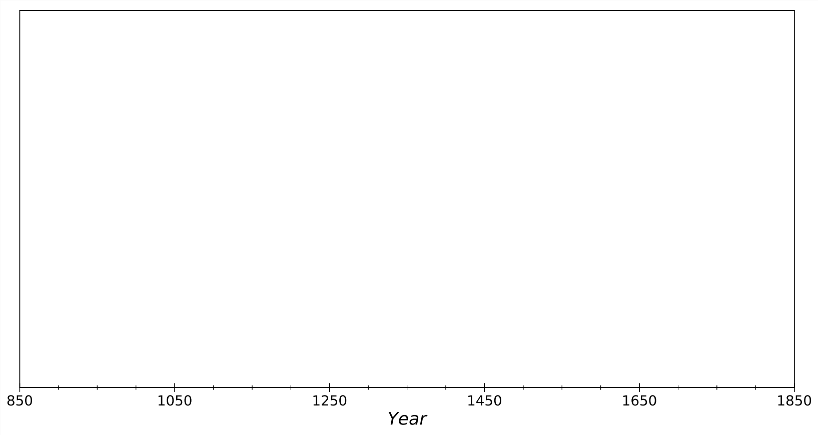
Welcome to the lowEBMs documentation!

This documentation primarily aims on explaining the structure of the sourcecode and show how to use it for your own purpose. It is now the most well documented sourcecode, so if questions or concerns about implementations come up, please contact me.

- Physical Background / Model Types
  - Physical Background
  - Model Types
    - 0D-EBM

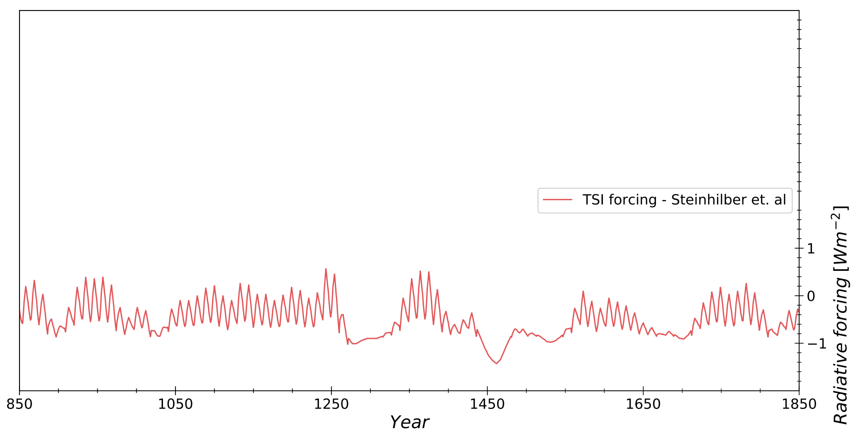
```
https://lowebms.readthedocs.io/en/latest/
```

Application period - Late Holocene



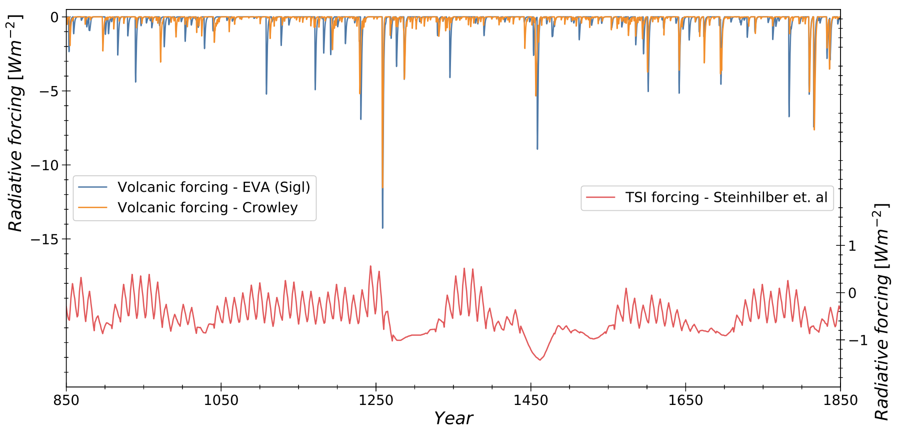
[Steinhilber 2009, Sigl 2015, Toohey 2016, Crowley 2008]

Application period - Late Holocene

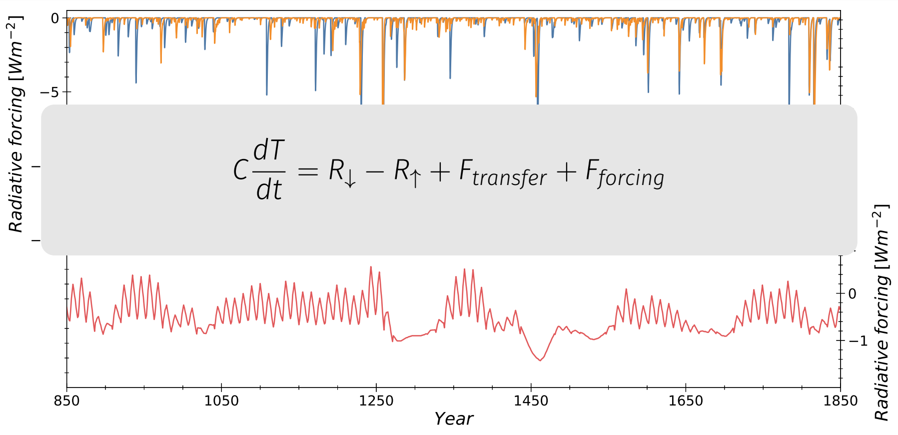


[Steinhilber 2009, Sigl 2015, Toohey 2016, Crowley 2008]

Application period - Late Holocene



[Steinhilber 2009, Sigl 2015, Toohey 2016, Crowley 2008]

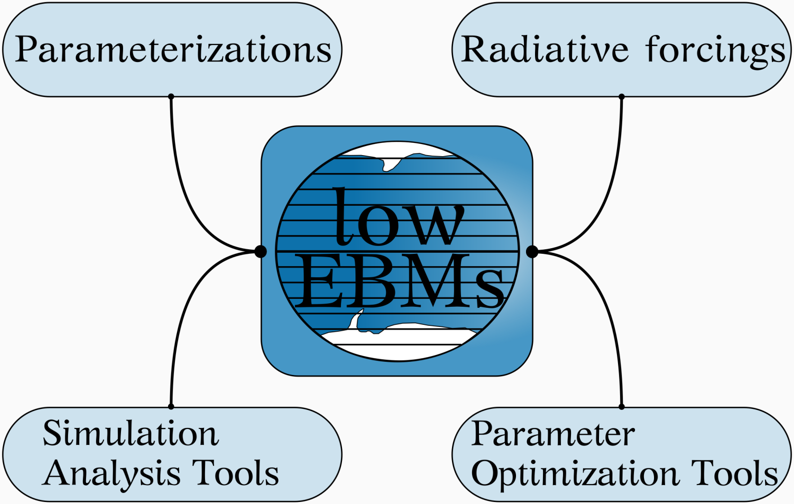


$$C\frac{dT}{dt} = R_{\downarrow} - R_{\uparrow} + F_{transfer} + F_{forcing}$$

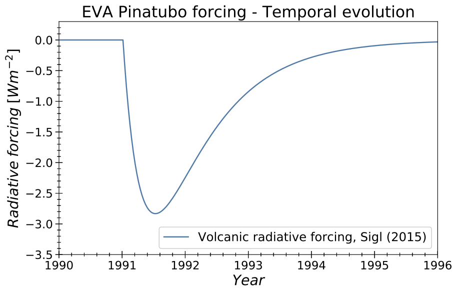
[Steinhilber 2009, Sigl 2015, Toohey 2016, Crowley 2008]

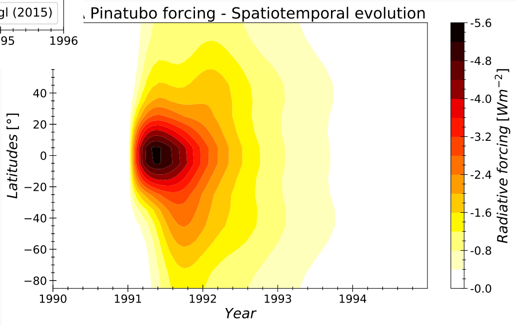
Jupyter notebook demonstration

Screen Change

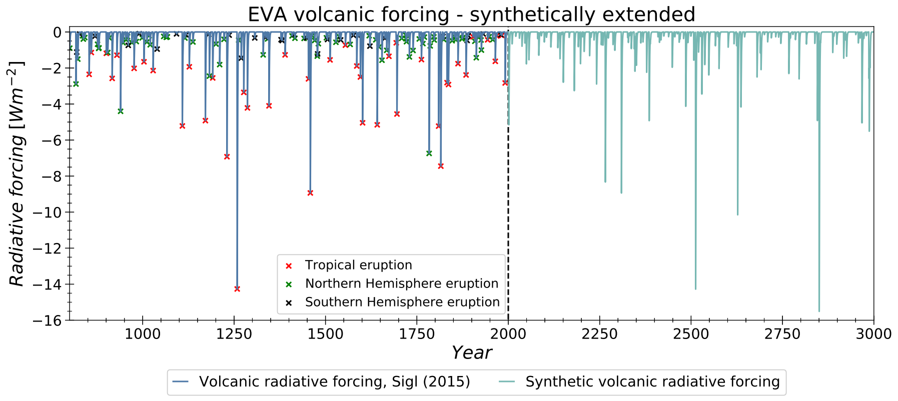

Parameterizations
Radiative forcings
low
EBMs
Simulation
Analysis Tools
Parameter
Optimization Tools

Additional features - VolcanicForcingGenerator

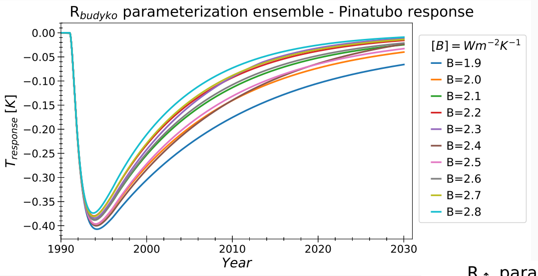


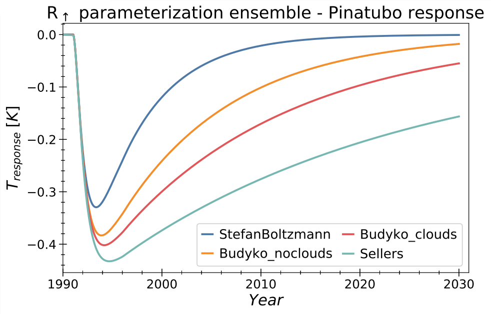


Additional features - VolcanicForcingGenerator



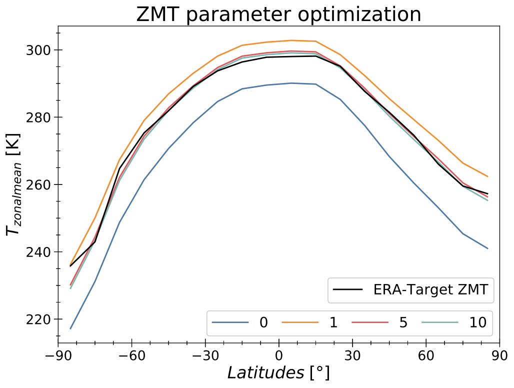
Additional features - Simulation analysis tools: Ensemble

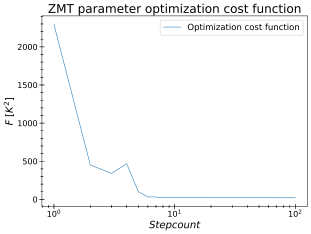


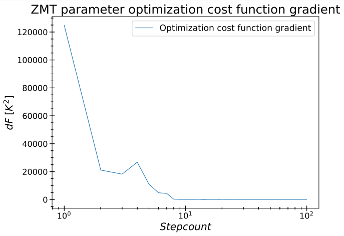


Additional features - Parameter optimization tools

- Optimization Method: Gradient descent
- Cost function: Least squares

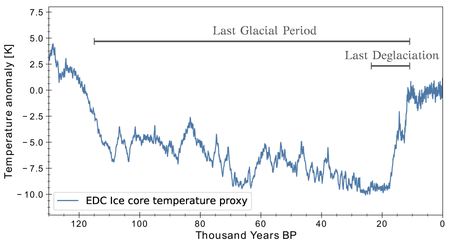


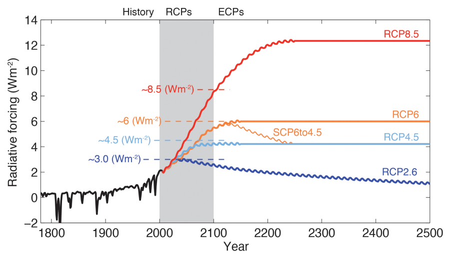




Outlook

- Tune parameterization to adapt to GCM/observation data
- Perform simulations over long timescales (LGP, LGM, Pliocene)
- Test various parameterizations and their climatic behaviour





[IPCC 2013, Jouzel 2007]

Thank you for your attention!

If you are interested in details:

`https://github.com/BenniSchmiedel/Low-dimensional-EBMs`

or `https://lowebms.readthedocs.io/en/latest/`

Referenzen

- M. I. Budyko, G. Observatory, and M. Spasskaja: The effect of solar radiation variations on the climate of the Earth. Tellus XXI (1969), 7, 1968.
- W. D. Sellers: A Global Climatic Model Based on the Energy Balance of the Earth-Atmosphere Sys- tem. Journal of Applied Meteorology, 8(3):392–400, 1969.
- IPCC: Climate Change 2013 - The Physical Science Basis. Cambridge University Press, Cambridge, 2013.
- M. Toohey, B. Stevens, H. Schmidt and C. Timmreck: Easy Volcanic Aerosol (EVA v1.0): an idealized forcing generator for climate simulations. Geosci. Model Dev., 2016.
- T. J. Crowley et al.: Volcanism and the Little Ice Age. PAGES News, April 2008.
- F. Steinhilber, J. Beer, and C. Fröhlich: Total solar irradiance during the Holocene. Geophysical Research Letters, 2009.
- G. R. North and K. Kwang-Yul: Energy Balance Climate Models. Wiley-VCH Verlag GmbH & Co. KGaA, 2017.
- T. Stocker: Introduction to Climate Modelling. Springer Berlin Heidelberg, 2011.
- K. McGuffie and A. Henderson-Sellers: The Climate Modelling Primer. Wiley-Blackwell, 2014.
- J. Jouzel et al.: Orbital and Millennial Antarctic Climate Variability over the Past 800,000 Years. Science, 2007.

Parameter	A_0	B_0	A_1	B_1	n	β
Value	$222.74\ \frac{W}{m^2}$	$2.23\ \frac{W}{m^2K}$	$47.73\ \frac{W}{m^2}$	$1.59\ \frac{W}{m^2K}$	0.5	$3.74\ \frac{W}{m^2K}$

$$R(t)_{in,i} = Q_i \cdot (1 - \alpha_i(T))$$

$$R(t)_{out,i} = A_0 + B_0 T(t)_i - n \cdot (A_1 + B_1 T(t)_i)$$

$$F(t)_{transfer,i} = \beta \cdot (T_i(t) - T_p(t))$$

$$R(t)_{in,i} = Q_i \cdot (1 - \alpha_i)$$
$$R(t)_{out,i} = \sigma T^4(t)_i (1 - m \cdot \tanh(\gamma T^6(t)_i))$$
$$F(t)_{transfer,i} = (P(t)_i l_i - P(t)_{i+1} l_{i+1})/A_i$$
$$P_i = L\, c(t)_i + C(t)_i + F(t)_{ocean,i}$$
$$c = \left(vq - K_w \frac{\Delta q}{\Delta y}\right) \frac{\Delta p}{g}$$
$$C = \left(vT - K_h \frac{\Delta T}{\Delta y}\right) \frac{c_p}{g} \Delta p$$
$$F_{ocean,i} = -K_o \Delta z \frac{l'}{l} \frac{\Delta T}{\Delta y} c_{p,w} \rho_w$$

$$v = -a(\Delta T \pm |\overline{\Delta T}|)$$

$$q = \frac{\epsilon e}{p}$$

$$\Delta q = \frac{\epsilon^2 L e \Delta T}{p R_d T_0^2}$$

$$e = e_0 \left(1 - 0.5 \frac{\epsilon L \Delta T}{R_d T_0^2}\right)$$

Description	Value
R_{out}	
m	0.5
σ	$5.67 \cdot 10^{-8}\ W/(m^2K^4)$
γ	$1.9 \cdot 10^{-15}\ 1/T^6$
$F_{transfer}$	
g	$9.81\ m/s^2$
ϵ	0.622
p	$1000\ mb$
e_0	$17\ mb$
L	$2.5 \cdot 10^3\ J/g$
R_d	$0.287\ J/(gK)$
Δy	$1.11 \cdot 10^6 m$
c_p	$1.004\ J/(gK)$
l'	0.5
$c_{p,w}$	$4.182\ J/(kgK)$
ρ_w	$997\ kg/m^3$