A micro/macro parallel-in-time (parareal) algorithm applied to a climate model with discontinuous non-monotone coefficients and oscillatory forcing

Giovanni Samaey[1] and Thomas Slawig[2]

[1] Department of Computer Science, K.U. Leuven, Celestijnenlaan 200A, 3001 Leuven, Belgium,
giovanni.samaey@kuleuven.be
[2] Department of Computer Science, Kiel University, 24098 Kiel, Germany,
ts@informatik.uni-kiel.de

June 13, 2018

Abstract

We present the application of a micro/macro parareal algorithm for a 1-D energy balance climate model with discontinuous and non-monotone coefficients and forcing terms. The micro/macro parareal method uses a coarse propagator, based on a (macroscopic) 0-D approximation of the underlying (microscopic) 1-D model. We compare the performance of the method using different versions of the macro model, as well as different numerical schemes for the micro propagator, namely an explicit Euler method with constant stepsize and an adaptive library routine. We study convergence of the method and the theoretical gain in computational time in a realization on parallel processors. We show that, in this example and for all settings, the micro/macro parareal method converges in fewer iterations than the number of used parareal subintervals, and that a theoretical gain in performance of up to 10 is possible.

Keywords: Parallel-in-time algorithm; micro/macro parareal algorithm; energy balance climate model; nonlinear partial differential equation.

1 Introduction

Climate simulation is among the most challenging and time-consuming computational tasks, for a number of reasons. First, there is the complexity of the coupled climate system, with interactions between many different components and nonlinearity of many important processes, some of which are not completely understood by now (see e.g. [23]). Second, the need for high spatial resolution in global climate models results in a huge dimension of

the discretized systems that have to be solved. To reduce the needed computational time, spatial parallelization is a common strategy used in fully coupled high resolution climate models. Still, for long-time simulation runs (e.g., to compute full glacial cycles of hundreds of thousands of years), spatial parallelization alone is insufficient, since spatial and temporal resolutions are typically coupled via some kind of CFL-type condition. Thus, such long-time simulation runs are only feasible with lower complexity models (see, e.g., [11]. To proceed to long-time *and* high resolution models, an additional parallelization in time becomes very attractive.

In this paper, we present how we generalized and used a micro/macro version of a parallel-in-time algorithm for time integration, called *parareal* [18], as it was developed in [17]. As an exemplary problem for a climate model, we choose an energy balance model (EBM) that describes the evolution of global mean temperature by balancing incoming and outgoing radiation, see, e.g., [23, Section 3]). Such models are the simplest way to model the Earth's climate. Incoming radiation is determined by the energy received from the Sun, diminished by a fraction (called albedo) which is reflected by the Earth's atmosphere (e.g., clouds) or surface (e.g., ice). Outgoing radiation is usually determined by considering the Earth as a perfect black radiating body, for which the Stefan-Boltzmann law gives a relation between the outgoing radiation and the temperature of the body. Since – due to the greenhouse effect – the Earth is not perfectly radiating, an emissivity parameter is included in EBMs to take into account that part of the emitted heat is captured in the Earth's atmosphere. Here, we regard as the "original" or "micro" model an EBM in one space dimension, using latitude as the spatial coordinate.

The micro/macro parareal method will use an approximate "macro" model to obtain a numerically fast predictor, which is iteratively corrected by time-parallel simulations using the original 1-D model on different slices of the time interval. The macro model is obtained by averaging the micro 1-D model in space, which results in a 0-D model that takes the form of an ordinary differential equation (ODE). The resulting macro model is much cheaper to simulation numerically for two reasons. First, due to its low-dimensionality, fewer degrees of freedom need to be accounted for. Second, the macro model only contains the dominant slow time scale, such that larger time steps can be taken. We mainly use the 0-D model to accelerate simulations. Nevertheless, 0-D EBMs have their own justification, for educational purposes (see [23]), since they include most important features of the Earth's energy balance. They model the Earth as a point in space, and thus all parameters mentioned above (energy received from the Sun as only external forcing of the climate system as well as albedo and emissivity that enter the differential equations as coefficients) represent averaged values over the whole planet.

If only a few parareal iterations are required, the micro/macro parareal algorithm can achieve a significant reduction in the required wall-clock time, compared to a naive simulation using only the micro-simulator by performing the 1-D simulations on different slices

of the time interval *in parallel* in each parareal iteration. In that case, one can reach an accuracy that is much higher than that of the macroscopic simulation on the whole time interval of interest, with a wall-clock time that is of the order of the simulation time of a few parareal time slices. In this paper, we study to what extent this potential is realized for climate simulations based on EBMs. In particular, we investigate the ability of the method to deal with oscillatory and abrupt changes in forcing terms, and we study the dependency of the method on the accuracy of the macro model and the number of parareal time slices.

Since its introduction in [18], the parareal strategy has been applied to a wide range of problems, including fluid-structure interaction [8], Navier–Stokes equation simulation [9], and reservoir simulation [12]. We refer to [21, 22] for further analysis, and to [1, 28] for stability results. In [10], the method is reformulated in a more general setting that relates the parareal strategy to earlier time-parallel algorithms, such as multiple shooting (see e.g. [16, 25]) or multigrid waveform relaxation (see e.g. [19, 31]). The micro-macro parareal method in this paper is a direct generalization of the method in [17], in which the micro model was a high-dimensional stiff ODE and the macro model was an approximate, low-dimensional ODE for a limited set of slow degrees of freedom. Similar micro-macro parareal methods have been considered in the literature. The authors of [3, 20] consider a singularly perturbed system of ordinary differential equations (ODEs) at the microscopic level and the limiting differential-algebraic equation at the macroscopic level. In [6], a parareal algorithm for multiscale stochastic chemical kinetics is presented, in which the macroscopic level uses the mean-field limiting ODE. In [24], the parareal algorithm is used with kinetic Monte Carlo at the macroscopic level and molecular dynamics at the microscopic level.

This text is organized as follows. In Section 2, we describe the model problem in its 0-D and 1-D versions. We also describe how the 1-D coefficients are captured in the 0-D version. In Section 3, we describe the micro/macro parareal algorithm that is the focus of the present paper. In Section 4, we describe in detail the numerical experiments that we performed. We present the numerical results in Section 5, discussing both the effect of the choice of coefficients in the 0-D model and the choice of micro and macro time integration methods. We end the paper with a summary and conclusions in Section 6.

2 Model problem

In this Section, we discuss the energy balance models (EBMs) that will be used throughout the paper. In the 1-D model, we have a single spatial coordinate ϕ, which varies from $\phi = 0$ at the north pole to $\phi = \pi/2$ at the equator, thus named colatitude. Due to a symmetry assumption (following [2, 13]), only half of the sphere is modeled. The state variable, temperature, then is a function $T = T(\phi, t)$ of colatitude and time. In this model, spatial redistribution of heat is included via diffusion, with a potentially temperature-dependent diffusion coefficient. Incoming radiation, albedo and thermal capacity depend on the co-

latitude ϕ. Albedo and emissivity (which models the greenhouse effect) also depend on temperature, to take into account ice melting or increase of water vapor in the atmosphere, respectively, leading to additional nonlinearities in the model. By averaging over the colatitude ϕ, a 0-D model can be obtained, in which the Earth is seen as a point in space, and thus all parameters (energy received from the Sun, albedo, emissivity) are averaged over the whole planet. The state variable is again temperature T, but now as a global mean value, i.e., a scalar function T of time t.

For clarity of exposition, we first discuss the 0-D model in Section 2.1, after which we elaborate on the 1-D model in Section 2.2. The choice of the parameters in the 1-D model is discussed in Section 2.3, along with the connection between the 0-D and 1-D models. We conclude this Section with some comments on existence and uniqueness of solutions (Section 2.4).

2.1 0-D model

In a 0-D EBM [23, Section 3.2], we consider the instantaneous change of the time-dependent temperature $T(t)$ due to the difference between incoming and outgoing radiation. Any difference between ingoing and outgoing, energy – $R_{in}(t)$, resp., $R_{out}(t)$ – induces a temporal change of thermal energy,

$$\left(4\pi r^2 h\ c\ \rho\right)\ T'(t) = R_{in}(t) - R_{out}(t)$$

in which $4\pi r^2 h$ is the volume of the considered spherical shell, c is the specific heat of the fluid, and ρ the respective density.

The total amount of incoming radiant energy per time unit for the whole Earth is given as

$$R_{in} \quad = \quad (1-\alpha)\pi r^2 S \tag{2.1}$$

with $\alpha \in [0,1]$ the albedo, πr^2 the area of the Earth's two-dimensional projection (r being the Earth's radius), and $S \approx 1367\,\mathrm{Wm}^{-2}$ the amount of energy per second and area (often denoted as solar "constant", but it fact not constant due to temporal variations of solar activity). In general, the albedo α may be temperature-dependent. We will specify its value in Section 2.3.

The outgoing radiant energy per unit time is given by the Stefan-Boltzmann law,

$$R_{out} \quad = \quad 4\pi r^2\ \epsilon\ \sigma T^4 \tag{2.2}$$

with $4\pi r^2$ the Earth's surface and $\sigma = 5.67 \times 10^{-8}\mathrm{Wm}^{-2}\,\mathrm{K}^{-4}$ the Stefan-Boltzmann constant, and including the emissivity ϵ, i.e., the fraction of outgoing radiation that is not captured by the atmosphere. Like the albedo α, the emissivity ϵ usually depends on the temperature, see Section 2.3.

Since πr^2 cancels out in the balance equation, we obtain as resulting ODE

$$T'(t) \;=\; \frac{1}{C}\left((1-\alpha)Q - \epsilon\sigma T(t)^4\right), \tag{2.3}$$

where we introduced the symbol $Q := S/4$ for notational convenience.

In the easiest case of constant solar insolation $Q := S/4$, heat capacity $C = hc\rho$, emissivity ϵ, and albedo α, a stationary solution can be computed from (2.3) as

$$T_{stat} \;=\; \sqrt[4]{\frac{(1-\alpha)Q}{\epsilon\sigma}}. \tag{2.4}$$

2.2 1-D model

The following 1-D model, based on [2, 13] and [14, Section 10], includes variation of the temperature in the latitudinal direction $\phi \in [0, \pi/2]$, i.e., we model a half sphere from north pole to equator. To this end, we add diffusive term of the form $\nabla \cdot (k\nabla T(\phi, t))$. Using spherical coordinates and taking into account that colatitude is the only spatial coordinate, this term reduces to

$$\nabla \cdot (k\nabla T(\phi, t)) \;=\; \frac{1}{\sin\phi}\frac{\partial}{\partial\phi}\left(k\sin\phi\frac{\partial T(\phi,t)}{\partial\phi}\right). \tag{2.5}$$

The coefficient k may, in principle, depend on space and temperature. We also include a dependence on the spatial coordinate for the solar radiation Q, the albedo α, and the heat capacity C a on the spatial coordinate ϕ. In the most general setting, the complete 1-D equation of energy balance gives:

$$C(\phi,T)\frac{\partial T}{\partial t} = \frac{1}{\sin\phi}\frac{\partial}{\partial\phi}\left(k(\phi,T)\sin\phi\frac{\partial T}{\partial\phi}\right) + \left[1-\alpha(\phi,T)\right]Q(\phi) - \epsilon(T)\sigma T^4. \tag{2.6}$$

We use homogeneous Neumann boundary conditions

$$\frac{\partial T}{\partial\phi}(\phi,t) \;=\; 0 \quad \text{for } \phi \in \left\{0, \frac{\pi}{2}\right\}, t \geq 0,$$

where the condition at $\phi = \pi/2$ (equator) comes from the assumed symmetry. The precise choice of the coefficients is the subject of the next Section 2.3.

In the numerical experiments, we discretize equation (2.6) in space. We introduce a spatial grid for $\phi \in [0, \pi/2]$ with stepsize $\Delta\phi = \pi/(2I), I \in \mathbb{N}$. The gridpoints then are $\phi_i = i\Delta\phi, i = 0, \ldots, I$. On these points, we denote the approximate solution as $\mathbf{T}_i(t) \approx T(\phi_i, t)$. We then use standard finite differences, see Appendix A, to obtain a semi-discretization,

$$\mathbf{T}'(t) \;=\; f(t, \mathbf{T}(t)), \quad t \geq 0, \tag{2.7}$$

with $\mathbf{T}(t), f(t, \mathbf{T}(t)) \in \mathbb{R}^{I-1}$ and some initial value $\mathbf{T}(0) \in \mathbb{R}^{I-1}$. Equation (2.7) is subsequently discretized in time, either by a forward Euler method with fixed time step or by a variable-step variable order linear multistep method, see Section 4.

2.3 Spatial and temperature-dependent modeling

In the 1-D model, the parameters solar radiation Q and albedo α can depend on colatitude ϕ. The heat capacity C, albedo α and emissivity ϵ can depend on temperature, leading to nonlinearity in the models. We first discuss the heat capacity, diffusion and emissivity (Section 2.3.1), after which we go into more depth into the albedo (Section 2.3.2) and the solar radiation (Section 2.3.3).

2.3.1 Heat capacity, diffusion and emissivity

Heat capacity. For the heat capacity $C(T)$, we use the temperature-dependent function from [2, Section 4.5]:

$$C(T) \quad = \quad (C_1 + C_2 \tanh\left(C_3(T - T_s)\right)) \times 10^8 \,\mathrm{Jm^{-2}K^{-1}}$$

with $C_1 = 3.14, C_2 = 1.15, C_3 = 0.08$, and $T_s = 263.15$. It is shown in Figure 1.

Diffusion coefficient. For the diffusion coefficient $k(\phi, T)$, Ghil [13, Eqn. (2e)] used the following nonlinear function of colatitude and temperature

$$k(\phi, T) \quad = \quad k_1(\phi) + k_2(\phi) g(T), \quad g(T) = \frac{c_4}{T^2} \exp\left(-\frac{c_5}{T}\right),$$

with given data for the coefficient functions k_1, k_2 at the grid points (see [13]) and parameters

$$\begin{aligned} c_4 &= 6.105 \times 0.75 \times \exp(19.6) \times 10^2 \mathrm{NKm^{-2}} \approx 1.4891 \times 10^{11} \mathrm{NKm^{-2}}, \\ c_5 &= 5.35 \times 10^3 \,\mathrm{K}. \end{aligned}$$

Because this choice (especially the coefficients suggested in [13]) does not guarantee positivity of k, we use a constant value, namely $k = 0.591 \,\mathrm{Wm^{-2}K^{-1}}$, taken from [2, Section 4.5]. Note that, also in the literature cited above, mostly only one source of nonlinearity (either C or k) has been chosen.

Emissivity. For the emissivity ϵ, we use

$$\epsilon(T) \quad = \quad 1 - \epsilon_1 \tanh\left(\left(\frac{T}{T_\epsilon}\right)^6\right)$$

with $\epsilon_1 = 0.5, T_\epsilon = 284.15 \,\mathrm{K}$, which is Ghil's [13] suggestion.

2.3.2 Albedo

The albedo $\alpha(\phi, T)$ describes the fraction of solar radiation that is absorbed by the earth. Since this fraction depends on features such as clouds or snow and ice on the surface,

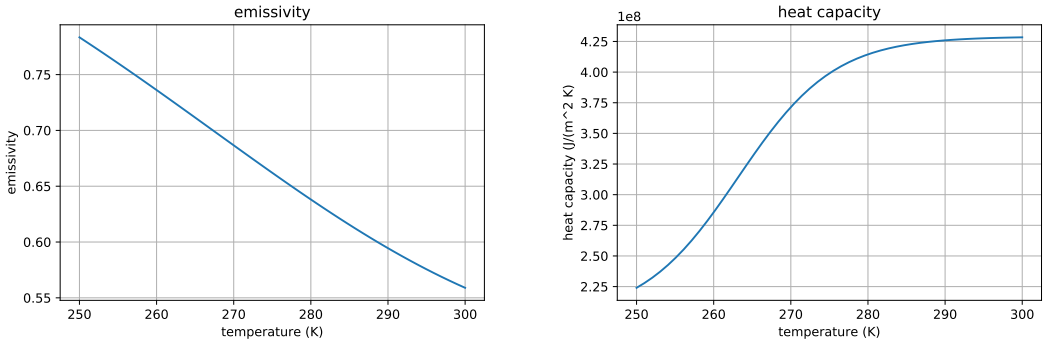

Figure 1: Left: Emissivity $\epsilon = \epsilon(T)$. Right: Heat capacity $C = C(T)$.

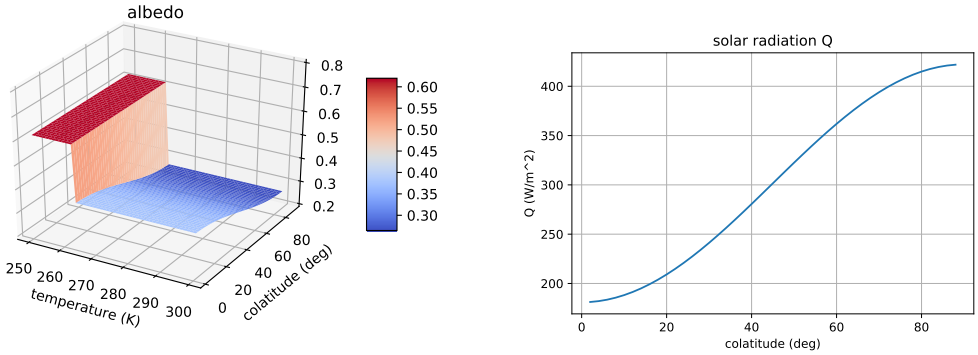

Figure 2: Left: Albedo $\alpha = \alpha(\phi, T)$. Right: Solar radiation $Q = Q(\phi)$.

the temperature-dependence is obvious. Piecewise constant and linear functions as well as smoothed counterparts are used in the literature. For the 1-D model, spatial dependency reflects the bigger area covered by ice or snow (with higher albedo) near the poles. We use the model of North [26], see also [2, Section 4.5]:

$$\alpha(\phi, T) \quad = \quad \begin{cases} \alpha_{\max}, & T \le T_s, \\ \alpha_1 + \alpha_2(\alpha_3 \cos^2(\phi) - \alpha_4), & T > T_s, \end{cases} \tag{2.8}$$

with $\alpha_{\max} = 0.62, \alpha_1 = 0.303, \alpha_2 = 0.0779, \alpha_3 = 1.5, \alpha_4 = 0.5$, and $T_s = 263.15$. Figure 2 shows the resulting non-discontinuous function.

In the 0-D version, we restrict to a simple step function

$$\alpha(T) \quad = \quad \begin{cases} \alpha_{\max}, & T \le T_s, \\ \alpha_{\min}, & T > T_s, \end{cases} \tag{2.9}$$

which is also mentioned in [5], [14, Section 10.2], and [23, Section 3.2.2]. The parameter $\alpha_{\min}$ in the 0-D model is not present in the 1-D model. We chose $\alpha_{\min} = 0.275$, which gives a good approximation of the long-time steady state obtained by the micro model.

2.3.3 Solar radiation

Space dependence. Clearly, the incoming solar radiation $Q(\phi)$ depends on the colatitude of the considered position on the Earth's surface. We use a second order polynomial in the variable $x = \sin\phi$, see [29, Section 4.3] or [2, Section 4.2]),

$$Q(\phi) \;=\; \frac{S}{4}\left(Q_1 + Q_2 \sin^2\phi\right) \tag{2.10}$$

with $Q_1 = 0.5294$ and $Q_2 = 0.706$, see Figure 2. Ghil [13] took data of which the spatial distribution is similar to the one of the polynomial above.

For the 0-D model, we use the average of (2.10) over the half-sphere. The spatial mean of a quantity depending on colatitude ϕ is obtained by integrating it over $\phi \in [0, \pi/2]$ and dividing by the measure of the area. In spherical coordinates the arc length of the circle at constant colatitude ϕ is given by $\pi r \sin\phi$ (r being again the Earth's radius). This gives

$$\overline{Q} \;=\; \left(\int_0^{\frac{\pi}{2}} Q(\phi)\sin\phi\, d\phi\right)\left(\int_0^{\frac{\pi}{2}} \sin\phi\, d\phi\right)^{-1} = \int_0^{\frac{\pi}{2}} Q(\phi)\sin\phi\, d\phi, \tag{2.11}$$

since the integral in the denominator equals 1. The integral for Q given by the formula (2.10) can be evaluated exactly. Since we also made experiments with the date provided by Ghil, we use a numerical approximation by the trapezoidal quadrature rule on the numerical grid on which equation (2.6) is solved.

Temporal variation. Temperature data reconstructions over one or more glacial cycles typically show relatively sharp gradients which are superimposed by small-scale fluctuations [4]. Note that, in this context, "sharp" has to be interpreted in relation to the considered time ranges of several hundred thousands of years. Furthermore, a global warming trend can be seen in temperature observations in the last decades. Our motivation for the use of the parareal method is to capture such multi-scale behavior in climate models. Thus we additionally varied Q in the 1-D model by adding some jumps, a linear trend and small-scale random fluctuations, described by a time-dependent function ΔQ. We write

$$Q_\Delta(\phi, t) = Q(\phi)(1 + \Delta Q(t)), \tag{2.12}$$

in which the perturbation $\Delta Q(t)$ has the following form:

$$\Delta Q(t) = \sum_{i=1}^{2} q_i \chi_{[a_i,b_i]}(t) + q_3 \chi_{[a_3,b_3]}(t)(t - a_3) + q_4 R(t). \tag{2.13}$$

Here $\chi_{[a,b]}$ is the characteristic function on the interval $[a, b]$ and $\{R(t) : t \in \{1, \ldots, t_{\text{end}}\}\}$ is a set of uniformly distributed random numbers in $[-1, 1]$. These were computed once and then fixed for all experiments. For our tests, we use the values,

$$\begin{aligned} &q_1 = 0.05, \quad a_1 = 283, \quad b_1 = 335, \quad q_2 = -0.03, \quad a_2 = 487, \quad b_2 = 564, \\ &q_3 = 0.0001, \quad a_3 = 700, \quad b_3 = 1000, \quad q_4 = 0.05. \end{aligned} \tag{2.14}$$

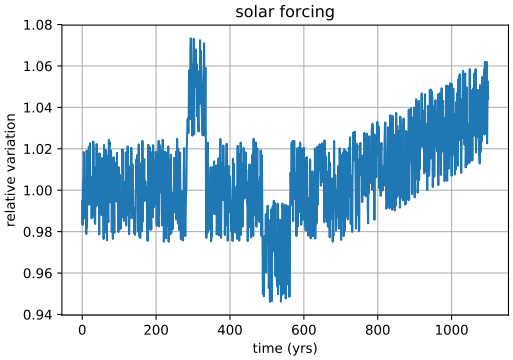

Figure 3: Introduced variation of solar forcing with four jumps, a linear trend after year 700, and a random variation in the whole time interval $[0, 1000]$, compare (2.12), (2.13) and (2.14).

In the time integration schemes, the values of $R(t)$ were interpolated linearly whenever needed at non-integer time instants. Figure 3 shows the relative deviation from the constant incoming solar radiation Q as function of time.

For the 0-D model, we tested two choices to handle the time-dependence of $Q_\Delta(t)$:

(1) a spatially averaged value that takes into account the time-dependence of $Q_\Delta(\phi, t)$ (ignoring the random fluctuations);

$$\overline{Q}_\Delta(t) = \int_0^{\frac{\pi}{2}} Q_\Delta(\phi, t) \sin\phi \, d\phi$$

(2) a constant value $\overline{Q}_\Delta(t) = \overline{Q}$ for all t.

2.4 Existence and uniqueness of solutions

In this section, we briefly summarize existence and uniqueness results of the two models. With the temperature-dependent modeling introduced above, the 0-D model takes the form

$$T' = \frac{(1 - \alpha(T))\overline{Q} - \epsilon(T)\sigma T^4}{C(T)}. \tag{2.15}$$

With the albedo being the step-function (2.9), the right-hand side does not fulfill the classical assumptions of the theorems of Peano or Picard-Lindelöf for existence and uniqueness. On the other hand, we have $\tanh(x) \in [-1, 1], x \in \mathbb{R}$, and thus $C(T) > 0$ for $T \in \mathbb{R}_{\geq 0}$. Hence, the sign of T' can be deduced from the numerator of the right-hand side of (2.15), compare Figure 4: For $T < T_s$ we have $T' < 0$ and for $T > T_s$ we have $T' > 0$. Thus any solution of the 0-D initial value problem for (2.15) with $T(0) = T_0$ will always remain in either $(0, T_s)$ or (T_s, ∞), depending in which interval T_0 lies. In each of these intervals, the right-hand side of the equation is continuously differentiable w.r.t. T and thus locally Lipschitz continuous.

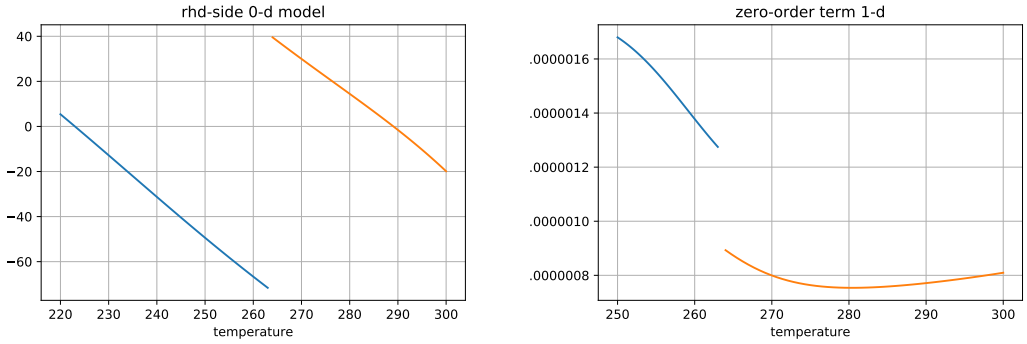

Figure 4: Left: Numerator of the right-hand side of the 0-D model. Right: Zero-order term of the 1-D model, here for spatially averaged values of α, Q.

Thus the Picard-Lindelöf Theorem is applicable and gives existence and uniqueness of the solution.

The 1-D model is a semilinear parabolic PDE of the general form

$$C(T)\,T_t + (kT_\phi)_\phi + g(\phi, T) = f,$$

where subscripts denote partial derivatives. Since C is positive it can be re-written as

$$T_t + \frac{1}{C(T)}\,(kT_\phi)_\phi + \tilde{g}(\phi, T) = 0.$$

It is not the aim of this paper to perform an analysis of the equation. We just note that the zero-order term

$$\tilde{g}(\phi, T) = \frac{(\alpha(\phi, T) - 1)Q(\phi) + \epsilon(T)\sigma T^4}{C(T)}$$

shows – besides the discontinuity at $T = T_s$ coming from the albedo function – a non-monotonicity w.r.t. T for $T \geq T_s$, compare Figure 4. Classical existence theorems (see e.g. [7, Section 9.2, Theorem 2], [30, Section 7.3, Lemma 5.3]) require either Lipschitz continuity or monotonicity of $\tilde{g}$. This lack of theoretical results shows the typical analytical properties of nonlinear climate models, even if they are low-dimensional as in this case.

3 Micro/macro parareal algorithm

In this section, we propose the micro/macro parareal algorithm that will be studied further on. It is a straightforward generalization of the parareal algorithm proposed in [18]; see also [17], where a similar algorithm was proposed in the context of singularly perturbed ordinary differential equations. We first introduce the necessary notation (Section 3.1), after which we outline the algorithm (Section 3.2).

3.1 Notations

We introduce a time discretization $(t_n)_{n=0}^N$, with $t_n = n\Delta t$, as well as the numerical approximations $\mathbf{T}^n \approx \mathbf{T}(\phi, t_n)$ of the 1-D model (2.6) (or, more precisely, its spatial discretization (2.7)), and $T^n \approx T(t_n)$ of the macroscopic model (2.3), respectively.

Fine-scale and macro propagators. The micro/macro parareal algorithm makes use of a *micro-scale propagator*, that advances the microscopic model (2.7) over a time-step Δt,

$$\mathbf{T}^{n+1} = F_{\Delta t}(\mathbf{T}^n), \tag{3.1}$$

and, similarly, a *macro propagator* for the macroscopic model (2.3),

$$T^{n+1} = G_{\Delta t}(T^n). \tag{3.2}$$

For now, we consider a forward Euler time discretization with time step Δt for the macro propagator and with time step $\delta t \ll \Delta t$ for the micro-scale propagator.

Lifting, restriction and projection. Furthermore, we introduce operators that connect the microscopic and macroscopic levels of description. The *restriction* operator

$$\mathcal{R} : T \mapsto T = \mathcal{R}(\mathbf{T}), \tag{3.3}$$

maps a microscopic state to the corresponding macroscopic state. For the model problem (2.7)-(2.3), this restriction operator simply becomes the averaging formula (2.11). For notational convenience, we also introduce the complement of the restriction operator,

$$\mathcal{R}^\perp(\mathbf{T}) := \frac{\mathbf{T}}{\mathcal{R}(\mathbf{T})},$$

such that we can write $T(\phi, t) = \mathcal{R}(T(\phi, t)) \cdot \mathcal{R}^\perp(T(\phi, t))$.

Conversely, to reconstruct a microscopic state from a given macroscopic state, we distinguish between a *lifting* operator, and a *projection* operator. A lifting operator $\mathcal{L}$ initializes a microscopic temperature profile $T(\phi, t)$ that is uniquely determined by the given spatially averaged temperature $T(t)$,

$$\mathcal{L} : T \mapsto \mathbf{T} = \mathcal{L}(T). \tag{3.4}$$

For the model problem (2.7)-(2.3), for instance, we could choose

$$T(\phi, t) = \mathcal{L}(T(t)) := T(t)\Psi(\phi), \tag{3.5}$$

with $\Psi(\phi)$ an arbitrary function such that $\mathcal{R}(\Psi(\phi)) = 1$. Clearly, one requires $\mathcal{R} \circ \mathcal{L} = \mathrm{Id}$.

In contrast, one may also *match* a "nearby" (*prior*) temperature profile $\mathbf{T}^*(\phi)$ with a desired macroscopic spatially averaged temperature T. Then, the result is not uniquely

determined by the macroscopic state T, but depends also on the prior $\mathbf{T}^*(\phi)$. We call the resulting reconstruction operator a matching operator,

$$\mathcal{P} : T, \mathbf{T}^*(\phi) \mapsto \mathbf{T} = \mathcal{P}(T, \mathbf{T}^*(\phi)). \tag{3.6}$$

This operator projects a microscopic temperature profile $\mathbf{T}^*(\phi)$ onto the manifold of microscopic temperature profiles consistent with the macroscopic spatially averaged temperature T. Here, we require $T = (\mathcal{R} \circ \mathcal{P})(T, \mathbf{T}^*(\phi))$, for any $\mathbf{T}^*(\phi)$. Additionally, a so-called *self-consistency* property is of particular importance.

Definition 3.1 (Self-consistency)**.** *A projection operator* $\mathcal{P} : T, \mathbf{T}^*(\phi) \mapsto \mathbf{T}(\phi) = \mathcal{P}(T, \mathbf{T}^*(\phi))$*, is called self-consistent if, and only if,*

$$\forall \mathbf{T}(\phi) : \mathcal{R}(\mathbf{T}(\phi)) = T \;\; \Rightarrow \;\; \mathbf{T}(\phi) = \mathcal{P}(T, \mathbf{T}(\phi)). \tag{3.7}$$

When this property holds, a microscopic temperature profile is not altered if it is projected onto a macroscopic spatially averaged temperature with which it is already consistent. As a guideline, the matching should be such that $\mathcal{R}(\mathbf{T}(\phi)) = T$, while requiring $\mathbf{T}(\phi)$ to be as close to $\mathbf{T}^*(\phi)$ as possible, in a sense to be made precise for the problem at hand. For the model problem (2.7)-(2.3), we choose the matching operator as

$$\mathbf{T}(\phi) = \mathcal{P}(T, \mathbf{T}^*(\phi)) := T \frac{\mathbf{T}^*(\phi)}{\mathcal{R}(\mathbf{T}^*(\phi))} = T \cdot \mathcal{R}^{\perp}(\mathbf{T}^*(\phi)). \tag{3.8}$$

3.2 Algorithm

The parareal algorithm iteratively constructs approximations on the whole time domain. We denote by T_k^n the approximate solution at time t_n, obtained during the k-th parareal iteration. We start from an initial condition $\mathbf{T}(\phi, 0) = T_0$, and create an initial approximation on the whole time interval by using the macro propagator, i.e. we restrict $T_0 = \mathcal{R}(\mathbf{T}_0)$, and compute

$$T_0^{n+1} = G_{\Delta t}(T_0^n), \qquad T_0^0 = T_0. \tag{3.9}$$

We then lift this macro approximation to the micro scale,

$$\mathbf{T}_0^n = \mathcal{L}(T_0^n). \tag{3.10}$$

We now have an initial approximation of the microscopic solution at each of the time instants t_n, $1 \leq n \leq N$. The parareal iterations then proceed as follows.

a) Compute (in parallel) the time propagation at each time instance, using the macro and micro-scale propagators,

$$\bar{T}_k^{n+1} = G_{\Delta t}(T_k^n), \tag{3.11}$$

$$\bar{\mathbf{T}}_k^{n+1} = F_{\Delta t}(\mathbf{T}_k^n). \tag{3.12}$$

b) Compute the jumps (the difference between the two propagated values) *at the macro level*,

$$J_k^{n+1} = \mathcal{R}(\bar{\mathbf{T}}_k^{n+1}) - \bar{T}_k^{n+1}. \tag{3.13}$$

c) Propagate (serially) the macro jumps over the whole time domain using the macro propagator,

$$T_{k+1}^{n+1} = G_{\Delta t}(T_{k+1}^n) + J_k^{n+1}, \tag{3.14}$$

and reconstruct the micro-scale state from the corrected macro state,

$$\mathbf{T}_{k+1}^{n+1} = \mathcal{P}(T_{k+1}^{n+1}, \bar{\mathbf{T}}_k^{n+1}). \tag{3.15}$$

Remark that the micro-scale state is reconstructed by projecting the intermediate value $\bar{\mathbf{T}}_k^{n+1}$ onto the corrected macro value T_{k+1}^{n+1}, using the matching operator (3.8).

4 Setup of numerical experiments

In this section, we describe the numerical experiments that we performed. In Section 4.1, we discuss the two options for the time integration of the micro and macro models. In Section 4.2, we discuss the two versions of the macro model that will be considered. The source code is available at `https://doi.org/10.5281/zenodo.1287561`.

4.1 Time integration

Two time discretization methods. We simulate up to time $t_{\text{end}} = 1000$, using an initial value of 285 K for all computations, in 0-D as well as uniform in space for the 1-D model. For the micro-scale propagator $F_{\Delta t}$, we use the spatial discretization (2.7) for the 1-D model, using $I = 45$. For the time discretization in the micro 1-D model, we use the built-in `odeint` routine in Python's `scipy` library, which is the routine `lsoda` [27] from the `odepack` library [15]. This method automatically switches between an Adams-Bashforth method (for non-stiff problems) or a backward Differentiation Formula (BDF, for stiff ones). The 1-D model is stiff, thus nearly in all cases and at all time instants the implicit version was chosen by the algorithm. The `odeint` routine also uses a variable time step, based on a user-prescribed error tolerance. We use $\varepsilon_{\text{rel}} = \varepsilon_{\text{abs}} = 10^{-6}$ for both relative and absolute tolerances. This means that a time-step is accepted if the predicted error e_n in time instant t_n satisfies

$$\|e_n\|_\infty \leq \varepsilon_{\text{rel}} \|\mathbf{T}_n\|_\infty + \varepsilon_{\text{abs}}.$$

Note that, because of the time-step adaptation strategy, we have no control over the time steps taken during time integration. In particular, it is very unlikely that the same

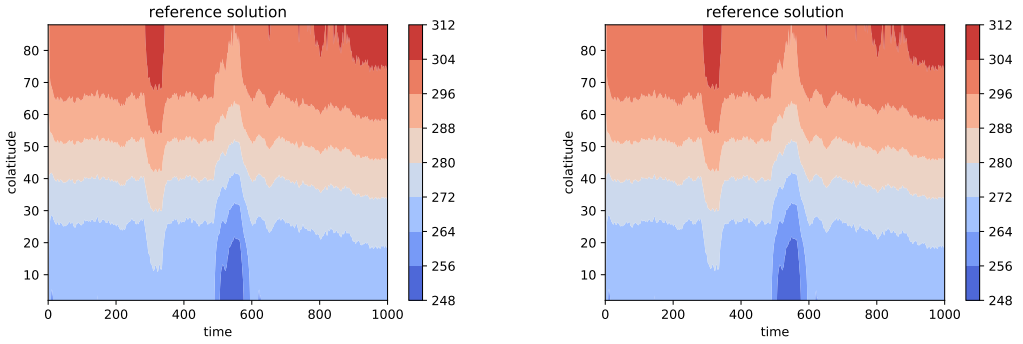

Figure 5: 1-D macro reference solutions obtained by the Euler method with constant stepsize 0.005 (left) and the adaptive `lsoda` library routine.

time step will be taken in subsequent parareal iterations (or in a serial micro simulation over the full time interval). As a consequence, the exactness property of the parareal method becomes very hard to check numerically. We therefore also perform numerical experiments in which the micro model is integrated using the explicit Euler method with fixed stepsize, chosen small enough (here 0.005) to give reasonable results for all experiments. Specifically the trajectory of the solution after the second jump ($t > a_2$, compare (2.14)), is difficult to obtain with larger stepsizes.

Reference solutions. We compute wo 1-D micro solutions sequentially using each of the above-described time discretization methods, and use them as reference solutions to compute the differences to the respective parareal solutions. These reference solutions are are denoted by $\mathbf{T}^*(t)$. and shown in figure 5.

Macro propagator. As the macro-scale propagator $G_{\Delta t}$ for the 0-D model, we again used the explicit Euler method, but now with the constant stepsize 10.0.

4.2 Two versions of macro model

Besides varying the microscopic time discretization, we also use two versions of the 0-D macro model. One has a similar temporally varying forcing as the micro model, but without the small-scale random fluctuations. This means we use ΔQ as in (2.13) and (2.14), but with $q_4 = 0$. The other version of the macro model has a constant solar forcing, i.e., $\Delta Q = 0$. The 1-D micro reference solution (with varying forcing) obtained with the Euler method (the same as in Figure 5 on the left, now averaged in space) is compared in Figure 6 with these two macro solutions. As can be seen and is obvious by construction, the macro model with constant solar forcing is an even more coarse and inaccurate approximation of the 1-D micro model. It does not follow neither the jumps nor the linear trend of the forcing

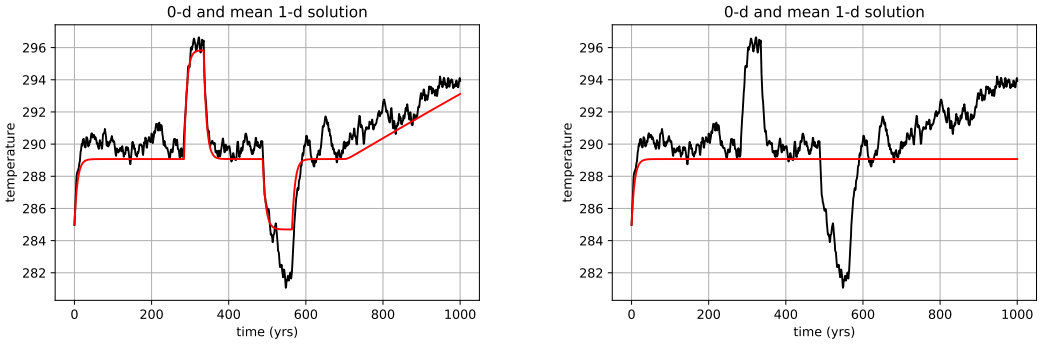

Figure 6: Spatially averaged 1-D micro solution $\overline{\mathbf{T}}$ with introduced perturbation compared to macro solution T with variable (left) and constant forcing (right).

(which is catched by the micro model), whereas the macro model with variable forcing does, at least to some extend. Our aim was to see how much this fact influences the parareal convergence.

4.3 Summary of numerical setup

We thus end up with four configurations, namely using either Euler or an adaptive library routine as micro solver and using either the macro model with or without added perturbation Q_Δ in solar forcing. For all these four settings, we varied the parareal time-step $\Delta t \in \{10, 20, 25, 40, 50, 100\}$ (corresponding to $N \in \{100, 50, 40, 25, 20, 10\}$) and compared the results w.r.t. difference of the parareal solution $\mathbf{T}_k$ in iteration k and the serially computed micro reference solution $\mathbf{T}^*$. The differences were always evaluated at all integer time instants $t \in \{1, \ldots, t_{\text{end}}\}$.

With the tolerances set as in Section 4.1, the library routine `odeint` and the forward Euler method require approximately the same time for a computation. In this setting, the 0-D macro model with constant forcing requires (approximately and on average (over all performed runs) only a fraction 1/700 of the micro computation time. The macro model with temporally varying forcing takes about 2-3 times as much time as macro model with constant forcing. Recall that the 1-D micro model always uses the varying forcing.

5 Numerical results

5.1 Results using Euler method as micro propagator

Figures 7 and 8 show the convergence of the parareal method to the micro reference solution using the Euler method for the micro model. It can be seen that the differences reach machine precision, see also Table 1. A reasonable difference level of 10^{-2} (talking about temperature which is the range of 200-300 K) is reached even faster.

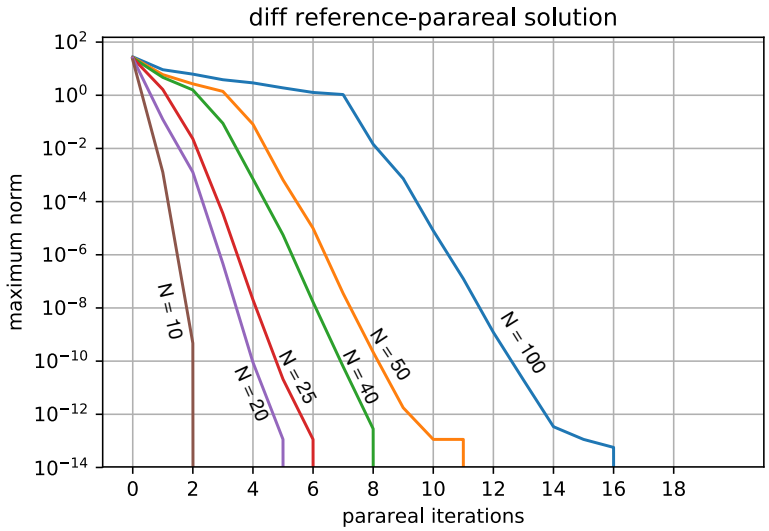

Figure 7: Convergence of parareal method using Euler method for the micro model and the macro model with temporally varying solar forcing (depicted in left plot of Figure 6) for different numbers N of parareal subintervals. Plotted are maximum norms over space and time.

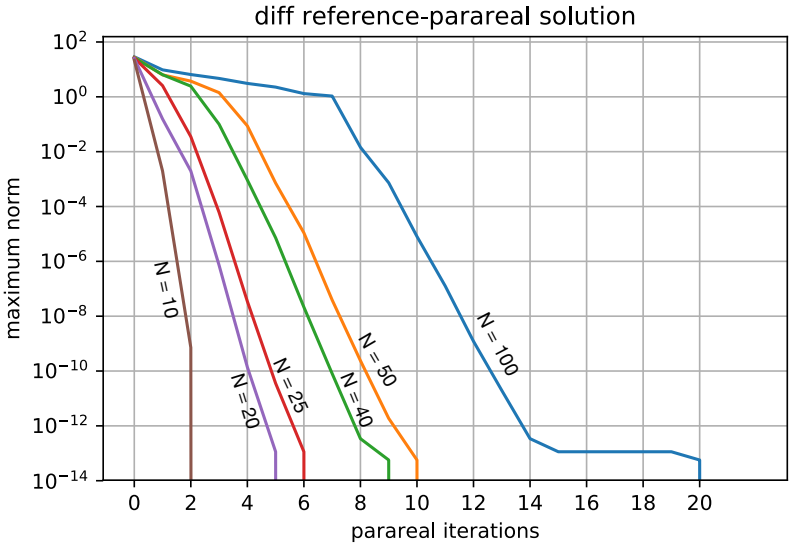

Figure 8: Same as Figure 7, but using the macro model with constant solar forcing (depicted in right plot of Figure 6).

Micro	N (number of parareal subintervals)	100	50	40	25	20	10
propagator	Δt (length of parareal subintervals)	10	20	25	40	50	100
explicit	macro model: temporally varying forcing						
Euler	$\min\{k : \|\mathbf{T}_k - \mathbf{T}^*\|_\infty < eps\}$	17	12	9	7	6	3
	$\min\{k : \|\mathbf{T}_k - \mathbf{T}^*\|_\infty < 10^{-2}\}$	9	5	4	3	2	1
	macro model: constant forcing						
	$\min\{k : \|\mathbf{T}_k - \mathbf{T}^*\|_\infty < eps\}$	21	11	10	7	6	3
	$\min\{k : \|\mathbf{T}_k - \mathbf{T}^*\|_\infty < 10^{-2}\}$	9	5	4	3	2	1
`lsoda`	macro model: temporally varying forcing						
library	$\min\{k : \|\mathbf{T}_k - \mathbf{T}^*\|_\infty < 10^{-2}\}$	12	5	4	3	3	1
routine	macro model: constant forcing						
	$\min\{k : \|\mathbf{T}_k - \mathbf{T}^*\|_\infty < 10^{-2}\}$	10	5	4	3	3	1

Table 1: Convergence of the micro/macro parareal method to a reasonable tolerance of 10^{-2}, using the Euler method and the adaptive library method in the micro model. For the Euler method also the convergence to machine precision (in double precision IEEE arithmetic, $eps \approx 2.2 \times 10^{-16}$) is shown. $\mathbf{T}_k$ denotes the parareal solution in the k-th iteration, $\mathbf{T}^*$ the reference solution.

To investigate the theoretical gain in computational effort, we assume that N processors are used and discard the effort for communication and macro propagator. It can be seen from the values in the table that a reduction by a factor of approximately 10 is reached. This refers to a computation up to a difference of the given reasonable tolerance of 10^{-2}. For computations up to machine precision the factor is approximately 3 to 5.

Figure 9 shows as example the solution obtained by the parareal method for the setting with $N = 100$ subintervals (Euler method in the micro, constant forcing in the macro model). This choice of subintervals showed the worst convergence behavior w.r.t. the needed number of parareal iterations, compare Figure 8 and Table 1. It can be seen that even in this case already after 4 iterations there is no visible difference between parareal and reference solution in the spatial mean, and also only a small one in the spatial state. The main difficulty for the method is to capture the steep gradient at the time instant close to $t = b_2$ (compare (2.14)), which is the second upward jump in solar forcing. This can be seen in Figure 10. It shows the difference between reference and parareal solution over time. Between 6th and 8th iteration, a significant reduction (by a factor ≈ 100) can be seen.

No big difference can be seen in the convergence behavior between the use of both macro model versions. Both versions of the macro model give the same precision after nearly the same number of parareal ierations.

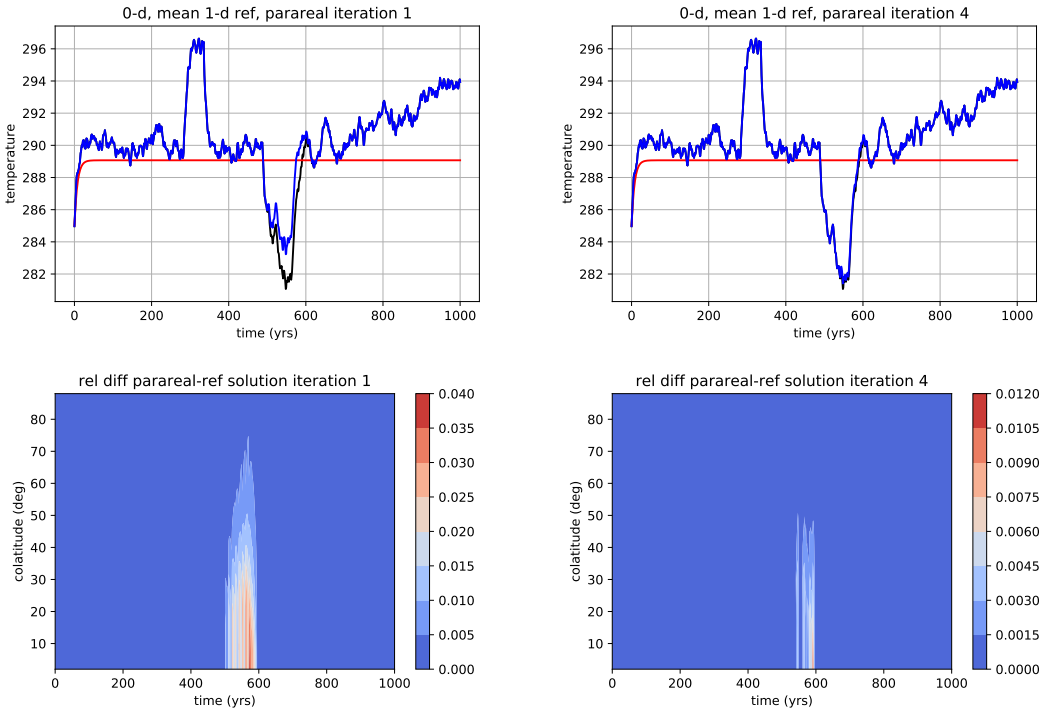

Figure 9: Top: Spatial mean of parareal and reference and macro solution after first (left) and fourth (right) iteration for $N = 100$ subintervals, using Euler method in the micro model and constant forcing in the macro model. Bottom: pointwise relative difference between parareal and reference solution.

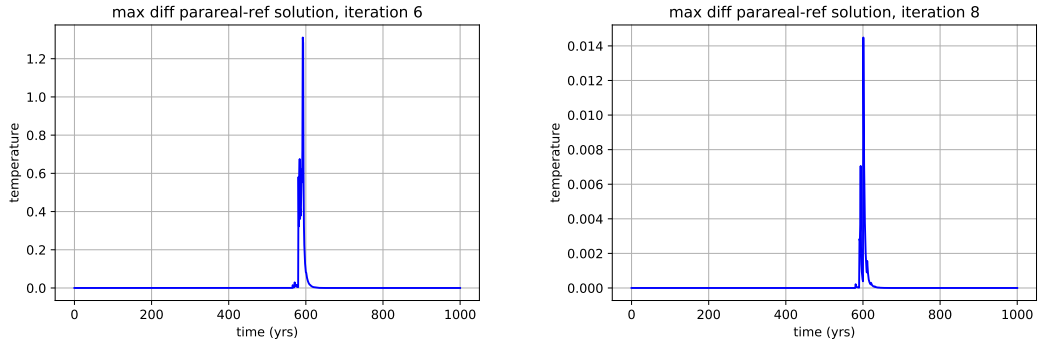

Figure 10: Temporal variation of maximum difference (in space) between parareal and reference solution after 6th (left) and 8th (right) iteration for $N = 100$ subintervals, using Euler method in the micro model and constant forcing in the macro model.

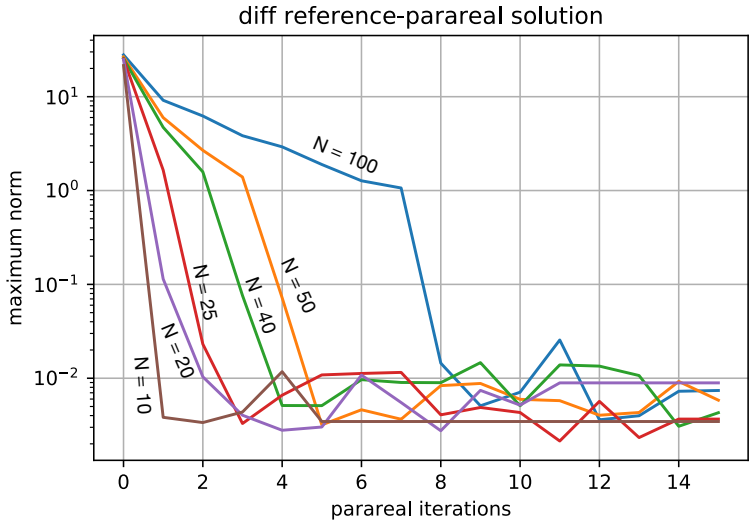

Figure 11: Same as Figure 7, but using `lsoda` library routine for the micro model, and the macro model with varying forcing.

5.2 Results using adaptive method as micro propagator

Figures 11 and 12 show the convergence of the parareal method using the `lsoda` library routine for the micro model. Here, the choice of the macro model leads to no relevant differences in any case. Due to the reasons already mentioned at the beginning of this section, a reduction of the difference between reference and parareal solution up to machine precision could not be reached in this setting. We have chosen the tolerances of the method such that a difference of less than 10^{-2} was reached. This allows us to compare the results, shown in Table 1, to those of the Euler method. No big difference in the number of iterations needed to give a maximal error less then 10^{-2} in both methods can be seen.

Figure 13 shows again as example the solution obtained for the setting with $N = 100$ subintervals with constant forcing in the macro model. Also here, this choice of subintervals showed the worst convergence w.r.t. the needed number of parareal iterations, compare Figure 12. The behavior is comparable to the results obtained using the Euler method: After four iterations no big difference is visible.

6 Summary and conclusions

We applied the micro/macro parareal method to a 1-D climate model with temporally multi-scale forcing. As macro model, we use two 0-D versions of the model with spatially averaged coefficients.

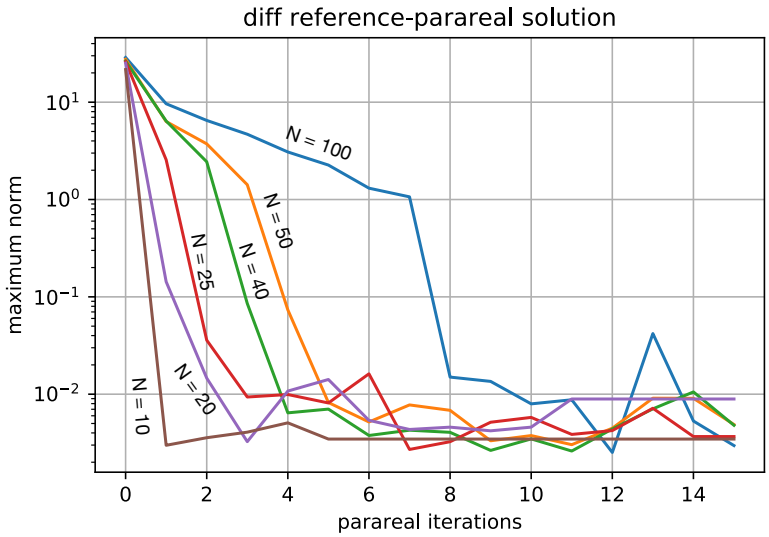

Figure 12: Same as Figure 11, but using the macro model with constant forcing.

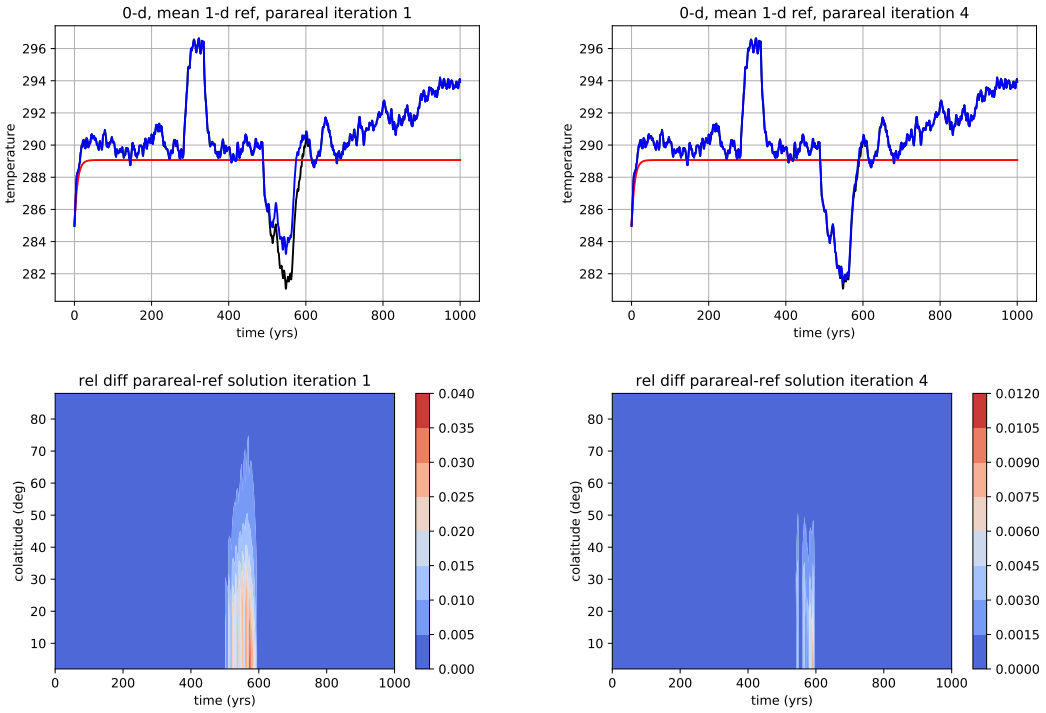

Figure 13: Same as Figure 9, but now for the `lsoda` library routine as micro model propagator after the first (left) and the fourth parareal iteration.

We investigated four configurations (micro model with explicit Euler method with constant stepsize or adaptive library routine on one hand and macro model with varying or constant forcing on the other) and a variety of number of parareal subintervals.

In all configurations, a reasonable difference to the respective reference solution (obtained serially with the same time integrator) was obtained after quite a low number of parareal iterations. The computational gain (discarding macro solver and communication), was about a factor of 10 when using any of the two micro propagators. Using the Euler method for the micro propagator, machine precision was reached in less iterations than numbers of used subintervals. For the adaptive library routine, this was not the case due to the different internal time-grids used on the whole time interval and when restarted on each subinterval.

There was no relevant difference between the two versions of the macro model: The more inaccurate version with constant forcing was able to predict the solution in all cases but one as well as the one with time-varying forcing.

We conclude that the applied micro/macro parareal algorithm is appropriate for this kind of problem, even if the macro model does not include the temporal multi-scale features. The computational gain could be even increased when lower (but still reasonable) accuracy requirements are used. Our results motivate the application of the micro/macro parareal method to more realistic climate models with higher spatial dimension and resolution.

Acknowledgements

GS acknowledges the support of the Research Council of the University of Leuven through grant 'PDEOPT', and of the Research Foundation Flanders (FWO - Vlaanderen) under grant G.A003.13. TS acknowledges the support of the German Federal Ministry of Education and Research (BMBF) as part of the Research for Sustainability Initiative (FONA) in the Palmod project, under grant 01LP1514A.

References

[1] G. Bal. On the convergence and the stability of the parareal algorithm to solve partial differential equations. In R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Widlund, and J. Xu, editors, *Domain decomposition methods in science and engineering*, volume 40 of *Lecture Notes in Computational Science and Engineering*, pages 425–432. Springer Berlin Heidelberg, 2005.

[2] D. Bayer. *Einfache mathematische Modelle zur Beschreibung globaler Klimaänderungen*. Verlag Dr. Kovac Hamburg, 1991.

[3] A. Blouza, L. Boudin, and S.-M. Kaber. Parallel in time algorithms with reduction methods for solving chemical kinetics. *Communications in Applied Mathematics and Computational Science*, 5(2):241–263, 2010.

[4] E. Brook. Windows on the greenhouse. *Nature*, 453:291–292, 2008.

[5] M. I. Budyko. The effect of solar radiation variations on the climate of the Earth. *Tellus*, 21:611–619, 1969.

[6] S. Engblom. Parallel in time simulation of multiscale stochastic chemical kinetics. *Multiscale Modeling and Simulation*, 8:46–68, 2009.

[7] L.C. Evans. *Partial Differential Equations*. American Math. Society, Providence, Rhode Island, 1998.

[8] C. Farhat and M. Chandesris. Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid–structure applications. *International Journal for Numerical Methods in Engineering*, 58(9):1397–1434, 2003.

[9] P. Fischer, F. Hecht, and Y. Maday. A parareal in time semi-implicit approximation of the Navier-Stokes equations. In R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Widlund, and J. Xu, editors, *Domain decomposition methods in science and engineering*, volume 40 of *Lecture Notes in Computational Science and Engineering*, pages 433–440. Springer Berlin Heidelberg, 2005.

[10] M.J. Gander and S. Vandewalle. Analysis of the parareal time-parallel time-integration method. *SIAM Journal on Scientific Computing*, 29:556–578, 2007.

[11] A. Ganopolski and V. Brovkin. Simulation of climate, ice sheets and co_2 evolution during the last four glacial cycles with an earth system model of intermediate complexity. *Climate of the Past*, 13(12):1695–1716, 2017.

[12] I. Garrido, M. Espedal, and G. Fladmark. A convergent algorithm for time parallelization applied to reservoir simulation. In R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Widlund, and J. Xu, editors, *Domain Decomposition Methods in Science and Engineering*, volume 40 of *Lecture Notes in Computational Science and Engineering*, pages 469–476. Springer Berlin Heidelberg, 2005.

[13] M. Ghil. Steady-State Solutions of a Diffusive Energy-Balance Climate Model and Their Stability. Technical Report IMM 410, New York Unversity, Courant Institute of Mathematical Sciences, May 1975.

[14] M. Ghil and S. Childress. *Topics in Geophysical Fluid Dynamics: Atmospheric Dynamics, Dynamo Theory, and Climate Dynamics*, volume 60 of *Applied Math. Sci.* Springer, 1987.

[15] A. C. Hindmarsh. ODEPACK, A Systematized Collection of ode solvers. In R. S. Stepleman et al., editor, *Scientific Computing*, volume 1 of *IMACS Transactions on Scientific Computation*, pages 55–64. North-Holland, 1983.

[16] H.B. Keller. *Numerical methods for two-point boundary-value problems.* Blaisdell (Waltham, MA), 1968.

[17] F. Legoll, T. Lelièvre, and G. Samaey. A micro-macro parareal algorithm: Application to singularly perturbed ordinary differential equations. *SIAM J. Sci. Comput.*, 35(4):1951–1986, 2013.

[18] J.-L. Lions, Y. Maday, and G. Turinici. Résolution d'EDP par un schéma en temps "pararéel". *C. R. Acad. Sci., Paris, Sér. I, Math.*, 332(7):661–668, 2001.

[19] C. Lubich and A. Ostermann. Multi-grid dynamic iteration for parabolic equations. *BIT Numerical Mathematics*, 27(2):216–234, 1987.

[20] Y. Maday. Parareal in time algorithm for kinetic systems based on model reduction. In A. Bandrauk, M.C. Delfour, and C. Le Bris, editors, *High-dimensional partial differential equations in science and engineering*, volume 41 of *CRM Proceedings and Lecture Notes*, pages 183–194. American Mathematical Society, 2007.

[21] Y. Maday and G. Turinici. A parareal in time procedure for the control of partial differential equations. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, 335(4):387–392, 2002.

[22] Y. Maday and G. Turinici. The parareal in time iterative solver: a further direction to parallel implementation. In R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Widlund, and J. Xu, editors, *Domain decomposition methods in science and engineering*, volume 40 of *Lecture Notes in Computational Science and Engineering*, pages 441–448. Springer Berlin Heidelberg, 2005.

[23] K. McGuffie and A. Henderson-Sellers. *The Climate Modelling Primer.* Wiley, Chichester, 4th edition, 2014.

[24] S. Mitran. Time parallel kinetic-molecular interaction algorithm for CPU/GPU computers. *Procedia Computer Science*, 1:745–752, 2010.

[25] J. Nievergelt. Parallel methods for integrating ordinary differential equations. *Communications of the ACM*, 7(12):731–733, 1964.

[26] G. R. North. Theory of Energy-Balance Clmeate Models. *J. Atmos. Sci.*, 32:2033–2043, 1975.

[27] L. R. Petzold. Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations. *SIAM J. Sci. Comput.*, 4:136–148, 1983.

[28] G. Staff and E. Rønquist. Stability of the parareal algorithm. In R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Widlund, and J. Xu, editors, *Domain decomposition methods in science and engineering*, volume 40 of *Lecture Notes in Computational Science and Engineering*, pages 449–456. Springer, 2005.

[29] T. Stocker. *Introduction to Climate Modeling (Lecture Notes)*. Physics Institute, University of Bern, 2014.

[30] F. Tröltzsch. *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*. Graduate Studies in Mathematics. American Mathematical Society, 2010.

[31] S. Vandewalle and R. Piessens. Efficient parallel algorithms for solving initial-boundary value and time-periodic parabolic partial differential equations. *SIAM Journal on Scientific and Statistical Computing*, 13:1330–1346, 1992.

A Discretization of equation (2.6)

A.1 Spatial discretization

In this appendix, we treat the spatial discretization of the right-hand side of (2.6), which basically means the discretization of the diffusion term (2.5). Here, we omit the temporal dependency of T in the notation. All indices referring to spatial discretization are denoted as subscripts, whereas indices coming from the temporal discretization will (later on) be denoted as superscripts.

We introduce a spatial grid for $\phi \in [0, \pi/2]$ with stepsize $\Delta\phi = \pi/(2I), I \in \mathbb{N}$. The gridpoints then are $\phi_i = i\Delta\phi, i = 0, \ldots, I$. On these points we compute the approximate solution denoted by $\mathbf{T}_i \approx T(\phi_i)$ for every discrete time-step. We also need intermediate points $\phi_{i+\frac{1}{2}} = (i + \frac{1}{2})\Delta\phi, i = 0, \ldots, I-1$.

We discretize the diffusion term (2.5) by applying central finite differences with stepsize $(\Delta\phi)/2$ twice: We approximate the inner derivative on the intermediate grid points by

$$\frac{dT}{d\phi}(\phi_{i+\frac{1}{2}}) \approx \frac{\mathbf{T}_{i+1} - \mathbf{T}_i}{\Delta\phi} =: \Theta_{i+\frac{1}{2}}, \quad i = 0, \ldots, I-1. \tag{A.1}$$

Then, the derivative of

$$F(\phi) := k(\phi, T(\phi)) \sin(\phi) \frac{\partial T}{\partial \phi}(\phi)$$

at the points ϕ_i is computed, again by central finite differences:

$$\nabla \cdot (k \nabla T(\phi))|_{\phi=\phi_i} = \frac{dF}{d\phi}(\phi_i) \approx \frac{F(\phi_{i+\frac{1}{2}}) - F(\phi_{i-\frac{1}{2}})}{\Delta\phi}, \quad i = 1, \ldots, I-1. \tag{A.2}$$

We need the values of the function $k = k(\phi, T(\phi))$ on the intermediate points:

$$\kappa_{i+\frac{1}{2}} \;:=\; k(\phi_{i+\frac{1}{2}}, T(\phi_{i+\frac{1}{2}})) \;\approx\; k\left(\phi_{i+\frac{1}{2}}, \frac{\mathbf{T}_i + \mathbf{T}_{i+1}}{2}\right), \quad i = 0, \ldots, I-1,$$

where the unknown temperature values at the intermediate points are interpolated linearly. For the outmost points $\phi_{\frac{1}{2}}, \phi_{I-\frac{1}{2}}$ this gives, using the homogenous Neumann boundary conditions:

$$\kappa_{\frac{1}{2}} \approx k(\phi_{\frac{1}{2}}, \mathbf{T}_1), \qquad \kappa_{I-\frac{1}{2}} \approx k(\phi_{I-\frac{1}{2}}, \mathbf{T}_{I-1}).$$

Eventually, needed values of the coefficient k (if not constant) at the intermediate points $\phi_{i+\frac{1}{2}}$ have to be interpolated accordingly. We then obtain for the terms on the right-hand side of (A.2):

$$F(\phi_{i+\frac{1}{2}}) \approx \kappa_{i+\frac{1}{2}} \sin(\phi_{i+\frac{1}{2}})\Theta_{i+\frac{1}{2}}, \qquad F(\phi_{i-\frac{1}{2}}) \approx \kappa_{i-\frac{1}{2}} \sin(\phi_{i-\frac{1}{2}})\Theta_{i-\frac{1}{2}}. \tag{A.3}$$

A.2 Time discretization

We now obtain a nonlinear nonautonomous system of ODEs

$$\mathbf{T}'(t) \;=\; f(t, \mathbf{T}(t)), \quad t \geq 0 \tag{A.4}$$

with $\mathbf{T}(t), f(t, \mathbf{T}(t)) \in \mathbb{R}^{I-1}$ and some initial value $\mathbf{T}(0) \in \mathbb{R}^{I-1}$. The i-th component of the function f is given by (2.6) using (A.1), (A.2), and (A.3). For both models, we re-scaled the time from seconds to years, i.e. we set

$$\tilde{t} \;:=\; \frac{t}{s_{year}} \quad \text{with } s_{year} = 60 \times 60 \times 24 \times 365 = 3.1536 \times 10^7.$$

The time derivative on the left-hand side of both models can be transformed using the definition $\tilde{T}(\tilde{t}) := T(s_{year}\tilde{t})$ and the formula

$$\frac{dT}{dt}(t) = \frac{dT}{d(s_{year}\tilde{t})}(s_{year}\tilde{t}) = \frac{1}{s_{year}} \frac{dT}{d\tilde{t}}(s_{year}\tilde{t}) = \frac{1}{s_{year}} \frac{d\tilde{T}}{d\tilde{t}}(\tilde{t}).$$

The dependency on ϕ in the 1-D model was suppressed here for simplicity. Then both models, (2.3) and (2.6), retain their formulation (omitting the tildes) when the respective right-hand sides are multiplied by s_{year}.