Radiative-convective equilibrium in a grey atmosphere

Marco Casari

October 2, 2023

Abstract

Radiative-convective models provide an intermediate complexity approach to the simulation of climate. These models evaluate the atmospheric temperature profile averaged over all latitudes and longitudes, which is function of time and altitude. Physical processes which determine the energy exchange in the model are absorption, transmission, reflection of electromagnetic radiation and convection of fluid. In this work a radiative-convective model is used to derive the temperature of an atmosphere where the optical depth is constant with respect to radiation frequency. The resulting temperature profile is compared with the analytical solution provided under the condition of radiative equilibrium alone.

1 Introduction

Climate dynamics of a planet can be studied with models of varying complexity. One of the quantities analysed is the temporal and spatial distribution of temperature in the planetary atmosphere, which is the result of the heat exchange between different processes. Electromagnetic (EM) radiation is emitted, absorbed and scattered by the chemical species distributed in the atmosphere and by the planetary surface. Moreover, the atmosphere receives EM radiation by other celestial bodies, e.g. stars. Radiative processes are described by the Radiative Transfer Equation (RTE). Derivation of the RTE in a form useful for atmospheric studies is in [1, p. 25]. Local temperature differences generate motion of fluid parcels, hence convection, and a rough planetary surface can hinder horizontal heat transport. Fluid dynamics equations are needed to represent these processes.

Fluid parcels are treated as open systems and the temperature distribution is obtained by a suitable thermodynamic energy equation, coupled with the equations of the processes occuring in the atmosphere. Even by limiting the analysis to radiative processes and convection, which are the main drivers of temperature variations, the equations involved are not solvable analytically and are not tractable numerically without simplifications.

A first approximation is to consider the average over all latitudes and longitudes for quantities which depend on spatial position. The resulting atmosphere is represented by plane-parallel layers, identified by their altitude z ranging from z_g at ground level to z_TOA at Top Of the Atmosphere (TOA). With this hypothesis, the differential equation describing the average temperature $T(t,z)$ as function of time t and altitude z is

$$\frac{\partial T}{\partial t} = -\frac{1}{\varrho c_P}\frac{\partial q}{\partial z} \quad , \tag{1}$$

where c_P is the specific heat at constant pressure of the atmosphere, ϱ is the average volumetric mass density of the atmosphere and depends on atmospheric pressure P, q is the total energy flux due to heat transfer. In general all these quantities are functions of t and z, also through T. Details on the derivation are in [2, p. 466], where the equation is written initially in terms of volumetric power densities and Local Thermodynamic Equilibrium (LTE) is assumed.

A second hypothesis is the radiative-convective equilibrium of the atmosphere. This translates in the existence of a steady state $\frac{\partial}{\partial t}T(t,z) = 0$ where the planet is in radiative equilibrium, i.e. the total irradiance at TOA is null, and the atmosphere is in convective equilibrium, i.e. fluid parcels are stable with respect to vertical motion.

A model with the previous assumptions is called Radiative-Convective Model (RCM). To simplify further the RTE, in this work the dependence of quantities on the frequency of EM radiation is neglected. An atmosphere with this property is called grey atmosphere.

In the following sections the vertical temperature profile of a grey atmosphere in radiative-convective equilibrium is computed. First the hypothesis of radiative equilibrium is used alone to obtain an analytical solution and use it as validation for the respective numerical solution. Then convection is taken into account and a simple RCM is implemented starting from the numerical scheme for radiative equilibrium.

1.1 Hypotheses and conventions

Some additional hypotheses are assumed to simplify the study. Data on constants are listed in table 1 and where possible, values referred to Earth are used for a prompt comparison with reality. Dependencies of quantities are written explicitly when it helps to clarify the discussion.

Some assumptions are made on the planet. It is supposed to have a diurnal cycle, to receive a constant irradiance S_0 and to have a constant Bond albedo A. These conditions result in a costant irradiance S_t transmitted to the illuminated hemisphere of the planet at TOA from outer space. The surface of the planet is approximated as blackbody emitting in the upward direction with constant temperature T_g. Gravitational acceleration g is constant. The atmosphere is supposed to be in hydrostatic equilibrium,

$$\mathrm{d}P = -\varrho g \,\mathrm{d}z \tag{2}$$

with P atmospheric pressure. Specific heat at constant pressure c_P is assumed constant. Scattering is neglected, hence the attenutation coefficient is equal to the absorption coefficient and the symbol $\mu(z)$ is used for both. Moreover, the absorption coefficient is supposed to depend on z through

$$\mu(z) = \mu_\mathrm{m}\varrho(z) \quad , \tag{3}$$

where μ_m is the mass attenuation coefficient of the atmosphere, assumed constant.

For gases, specific gas constant R_m is used in thermodynamic relations, which is defined as the gas constant R divided by the molar mass of the gas. They obey the ideal gas law

$$P = \varrho R_\mathrm{m} T \quad . \tag{4}$$

The total heat flux q is determined by radiative transfer and atmospheric convection. Other means of vertical heat transfer are neglected, e.g. precipitation. The RTE is not solved directly by the RCM, instead a two-stream approximation is adopted for the radiation inside atmosphere: components of radiometric quantities in the upward and downward directions are treated separately. Neither the contribution to q due to atmospheric convection is obtained by solving the proper fluid dynamics equations, in its place a numerical correction is adopted. With these considerations, equation (1) can be rewritten as

$$\frac{\partial T}{\partial t} = -\frac{1}{\varrho c_P}\frac{\partial}{\partial z}\big(E_\mathrm{U} - E_\mathrm{D}\big) \quad , \tag{5}$$

where E_U and E_D are irradiances of upward and downward radiations, respectively.

1.2 Vertical coordinates

Altitude is an immediate choice as vertical coordinate used to describe the problem. However, calculations may simplify more if expressed with other coordinates.

An alternative choice is P and is convenient when used with equation (2) to remove the dependence on ϱ. A bijective relation relates P and z (cf. section B.1):

$$P(z) = P_\mathrm{g} \exp\left(-\frac{z - z_\mathrm{g}}{z_0}\right) \quad , \tag{6}$$

where ground level is chosen as reference and constant z_0 acts to remove the depence on temperature and thus sets the scale of z. Pressure decreases with altitude from standard value P_g at ground level to P_TOA at TOA.

To simplify radiative calculations optical depth δ is used as vertical coordinate, starting from 0 at TOA and increasing downward, up to value δ_g at ground level. From the hypotheses on μ, δ is a function of altitude through

$$\delta(z) = \mu_\mathrm{m}\int_z^{z_\mathrm{TOA}} \varrho(z')\,\mathrm{d}z' \quad , \tag{7}$$

but a simpler relation exists between δ and P using equation (2) to evaluate the integral in equation (7):

$$\delta(P) = \frac{\mu_\mathrm{m}}{g}(P - P_\mathrm{TOA}) \quad . \tag{8}$$

Relation (8) is used in conjunction with equation (6) to derive a more direct formula for $\delta(z)$:

$$\delta(z) = \frac{\mu_\mathrm{m}}{g}\left(P_\mathrm{g}\exp\left(-\frac{z-z_\mathrm{g}}{z_0}\right) - P_\mathrm{TOA}\right) \quad . \tag{9}$$

Value P_TOA can be calculated from a fixed z_TOA using equation (6), or vice versa.

Any of the previous relations for δ can be used to fix the value of μ_m if δ_g is known, or conversely μ_m can be used as parameter to derive δ_g.

2 Analytical solution in radiative equilibrium

Steady states of temperature profile and irradiances considering only radiative processes are derived analytically in this section.

With the hypotheses of LTE and non-scattering medium, the RTE becomes

$$\frac{1}{\mu}\frac{\partial L}{\partial z} = B_\nu - L \quad , \tag{10}$$

where $L(t, z, \theta, \nu)$ is the spectral radiance arriving at altitude z with angle θ with respect to direction $\hat{z}$ and $B_\nu(\nu, T(t, z))$ is Planck's function (cf. section B.2).

To apply equation (10) to irradiances, two integrations are needed: one over the whole EM spectrum and one over the solid angle corresponding to a hemisphere. The latter can be performed adopting the diffusion approximation (cf. [4, p. 55] for a summary and [1, p. 498] for a more general derivation), which have the effect to substitute δ with $\delta' = D\delta$, where D is the diffusion coefficient. The resulting equations for irradiances in terms of δ' are

$$-\frac{\partial}{\partial\delta'}E_\mathrm{U}(t,\delta') = \sigma T(t,\delta')^4 - E_\mathrm{U}(t,\delta') \quad , \tag{11}$$

$$\frac{\partial}{\partial\delta'}E_\mathrm{D}(t,\delta') = \sigma T(t,\delta')^4 - E_\mathrm{D}(t,\delta') \tag{12}$$

and they are coupled with equation (5) written in terms of δ':

$$\frac{\partial}{\partial t}T(t,\delta') = \frac{\mu_\mathrm{m} D}{c_P}\frac{\partial}{\partial\delta'}\big(E_\mathrm{U}(t,\delta') - E_\mathrm{D}(t,\delta')\big) \quad . \tag{13}$$

Equations (13), (11) and (12) form a system of non-linear Partial Differential Equations (PDEs) of first order in two variables.

When the steady state of T is searched, dependence on t is dropped and the PDEs become Ordinary Differential Equations (ODEs) of first order of an Initial Value Problem (IVP). Initial conditions for irradiances are

$$E_\mathrm{D}(0) = 0 \tag{14}$$

because energy released to atmosphere at TOA by the downward flux is negligible and $E_\mathrm{U}(0)$ is a constant called Outgoing Longwave Radiation (OLR). Radiative equilibrium provides the value of OLR:

$$E_\mathrm{U}(0) = S_\mathrm{t} \quad . \tag{15}$$

Moreover, at the steady state irradiances are related by

$$\frac{\mathrm{d}}{\mathrm{d}\delta'}\big(E_\mathrm{U}(\delta') - E_\mathrm{D}(\delta')\big) = 0 \quad , \tag{16}$$

which has constant solution determined by the condition of radiative equilibrium for the planet:

$$E_\mathrm{U}(\delta') - E_\mathrm{D}(\delta') = S_\mathrm{t} \quad . \tag{17}$$

Same relations are derived if the hypotheses of atmosphere in radiative equilibrium at all altitudes and atmopshere transparent to radiation coming from outside the planet are considered instead of atmopshere in radiative equilibrium at TOA and steady state.

An ODE for T can be written by adding and subtracting equations (11) and (12) and using relations (16) and (17):

$$2\sigma\frac{\mathrm{d}}{\mathrm{d}\delta'}T(\delta')^4 = S_\mathrm{t} \quad . \tag{18}$$

Initial condition for T is obtained similarly by summing equations (11) and (12) and applying relation (16) and initial conditions (15) and (14):

$$T(0) = \left(\frac{S_\mathrm{t}}{2\sigma}\right)^{\frac{1}{4}} \quad . \tag{19}$$

The solution of equation (18) in terms of δ is

$$T(\delta) = \left(\frac{S_\mathrm{t}}{2\sigma}(1 + D\delta)\right)^{\frac{1}{4}} \quad , \tag{20}$$

represented in figure 1.

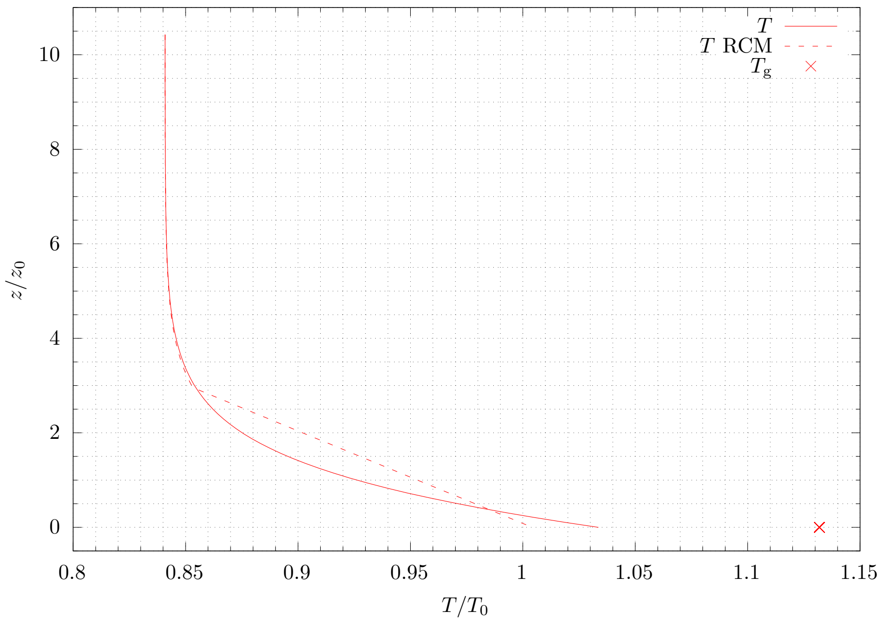

Figure 1: Vertical temperature profile of a grey atmosphere. Continuous line is the analytical solution in radiative equilibrium, dashed line is the numerical solution in radiative-convective equilibrium. Convective adjustment is visible at lower altitudes, but values are affected by the low precision of the numerical procedure. A discontinuity is present at ground level due to the lack of heat exchange between surface and the lowest atmospheric layer.

Once the temperature profile is known, $E_\mathrm{U}(\delta)$ and $E_\mathrm{D}(\delta)$ are evaluated with the same procedure used previously for T, resulting in:

$$E_\mathrm{U}(\delta) = \frac{S_\mathrm{t}}{2}(2 + D\delta) \quad , \tag{21}$$

$$E_\mathrm{D}(\delta) = \frac{S_\mathrm{t}}{2}D\delta \quad . \tag{22}$$

Irradiances are shown in figure 2. At every altitude, E_U and E_D are greater then their respective values at TOA. This is result of the greenhouse effect of the atmosphere.

Value δ_g can be fixed by using the irradiance emitted by the surface of the planet:

$$E_\mathrm{U}(\delta_\mathrm{g}) = \sigma T_\mathrm{g}^4 \quad . \tag{23}$$

With this condition, T presents a discontinuity at ground level, which is not physical. Other mechanisms of heat transport redistribute energy between the surface and the atmospheric layer directly above, removing the discontinuity. Their effect can be simulated by imposing radiative equilibrium at ground level.

3 Numerical solution in radiative equilibrium

To solve numerically the IVP defined in section 2, equations (18), (11) and (12) are rewritten in terms of variable δ and normalised,

$$Y_0 = \frac{T^4}{T_0^4} \quad , \quad Y_1 = \frac{E_\mathrm{U}}{S_\mathrm{t}} \quad , \quad Y_2 = \frac{E_\mathrm{U}}{S_\mathrm{t}} \tag{24}$$

with T_0 chosen arbitrarily, resulting in the system of ODEs

$$\begin{cases} \frac{\mathrm{d}Y_0}{\mathrm{d}\delta} = \frac{D}{2} \\ \frac{\mathrm{d}Y_1}{\mathrm{d}\delta} = D(Y_1 - Y_0) \\ \frac{\mathrm{d}Y_2}{\mathrm{d}\delta} = D(Y_0 - Y_2) \end{cases} \quad . \tag{25}$$

Initial conditions for the normalised functions are

$$Y_0 = \frac{1}{2} \quad , \quad Y_1 = 1 \quad , \quad Y_2 = 0 \quad , \tag{26}$$

from conditions (19), (15) and (14), respectively.

Runge-Kutta method of order 4 is used to integrate system (25), to maintain precision when T is derived from Y_0. Non-uniform step sizes are

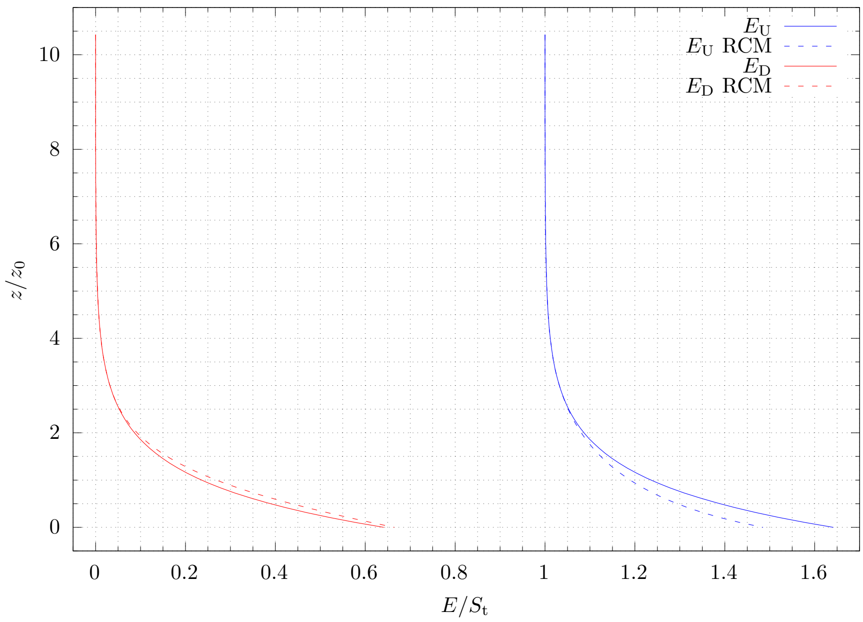

Figure 2: Upward and downward irradiances in a grey atmosphere. Continuous line is the solution in radiative equilibrium, dashed line is the solution in radiative-convective equilibrium. Greater values than TOA at lower altitudes are indicators of greenhouse effect. Irradiances of the RCM are not directly subject to convective adjustment, but they are affected due to the dependence on temperature.

adopted, because values δ are obtained from uniformly distributed values z through relation (9).

Accuracy of the numerical procedure is quantified through the errors of normalised T and irradiances with respect to analytical solutions. Errors are compatible with 0 based on precision of double-precision floating-point numbers, as shown in figure 3.

3.1 Stability analysis

Stability of the numerical method with respect to spatial grid size is studied varying N. Powers of 2 in the interval $[1, N_{\max}]$ are chosen as values of N and step size is kept constant, obtained by dividing interval $[\delta_{\mathrm{TOA}}, \delta_{\mathrm{g}}]$ in N subintervals. Errors are evaluated as absolute differences between numerical and analytical values for each of $Y_0(\delta_{\mathrm{g}})$, $Y_1(\delta_{\mathrm{g}})$ and $Y_2(\delta_{\mathrm{g}})$.

In figure 4 errors are plotted as function of N. For $N \leq 4096$, they are compatible with 0 within precision, while for greater N, they increase due to error propagation. In general, this behaviour does not hinder results of simulations because lower values of N are chosen for the model, otherwise averages approximating atmopsheric dynamics could become inaccurate and the computational demand of the equations involved could increase considerably.

3.2 Time integration

Numerical solutions for system (25) are obtained by using in advance the steady state condition (16). To preserve information on temporal depence, the more general system given by PDEs (13), (11) and (12) is solved numerically. More precisely, each variable is considered separately during the integration and an iterative procedure is adopted:

1. an arbitrary temperature profile is chosen;
2. equations (11) and (12) are solved with respect to δ';
3. the resulting E_{U} and E_{D} are used to step forward T with respect to t using equation (13) for each layer;

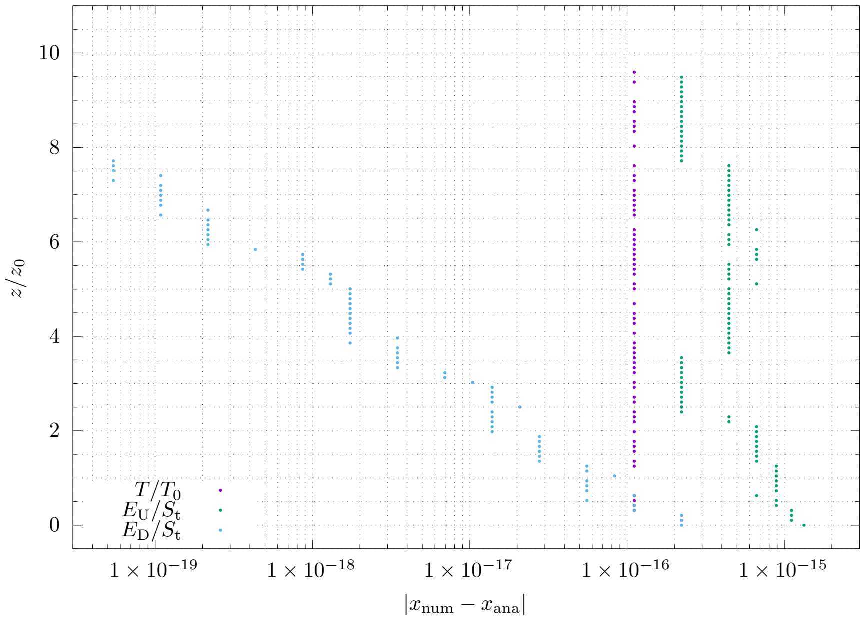

Figure 3: Errors between numerical and analytical solutions of a grey atmosphere in radiative equilibrium, assuming steady state. Points at some altitudes are not shown because their value is exactly 0.

4. the obtained temperature profile is used to restart the loop from point 2.

The iterations continue until a steady state for T is reached. This procedure describes an IVP with respect to δ' for T, E_U and E_D and an IVP with respect to t for T.

Normalisations (24) and

$$Y_3 = \frac{T}{T_0} \tag{27}$$

are used and δ is chosen as coordinate, hence the system of PDEs is rewritten as

$$\begin{cases} \frac{\partial}{\partial t} Y_3(t,\delta) = \frac{\mu_m S_t}{c_P T_0} \frac{\partial}{\partial \delta}(Y_1(t,\delta) - Y_2(t,\delta)) \\ \frac{\partial}{\partial \delta} Y_1(t,\delta) = D(Y_1(t,\delta) - Y_3(t,\delta)^4) \\ \frac{\partial}{\partial \delta} Y_2(t,\delta) = D(Y_3(t,\delta)^4 - Y_2(t,\delta)) \end{cases} . \tag{28}$$

At the start of the procedure, the initial condition for T is an arbitrary temperature profile, while at each temporal step, initial conditions (26) are reapplied. Point $Y_3(0,0) = \frac{1}{2}$ is fixed by radiative equilibrium at TOA but it is not used during the integration.

Integration of irradiances is performed as before using Runge-Kutta method of order 4. For the temporal integration of T Euler method is used with a costant time step Δt, chosen arbitrarily to reduce the errors of irradiances below the precision of numeric values outputs.

Figure 5 displays errors between numerical solutions of PDE system (28) and analytical solutions. Errors are propagated during the iterations following the non linearity of the equations, limiting the precision of the numerical procedure.

4 Radiative-convective equilibrium

Convective processes are responsible for heat transport in the vertical direction of the idealised atmosphere under study. The effect of convection is the motion of fluid parcels with T different than the surroundings, until the temperature gradient $-\frac{\partial T}{\partial z}$, called lapse rate, reaches a steady state. This state corresponds to convective equilibrium.

When convective processes are coupled with radiative processes, the former redistributes the energy accumulated because of the latter to higher layers, cooling the lower layers of the atmosphere. The two processes work on different time scales,

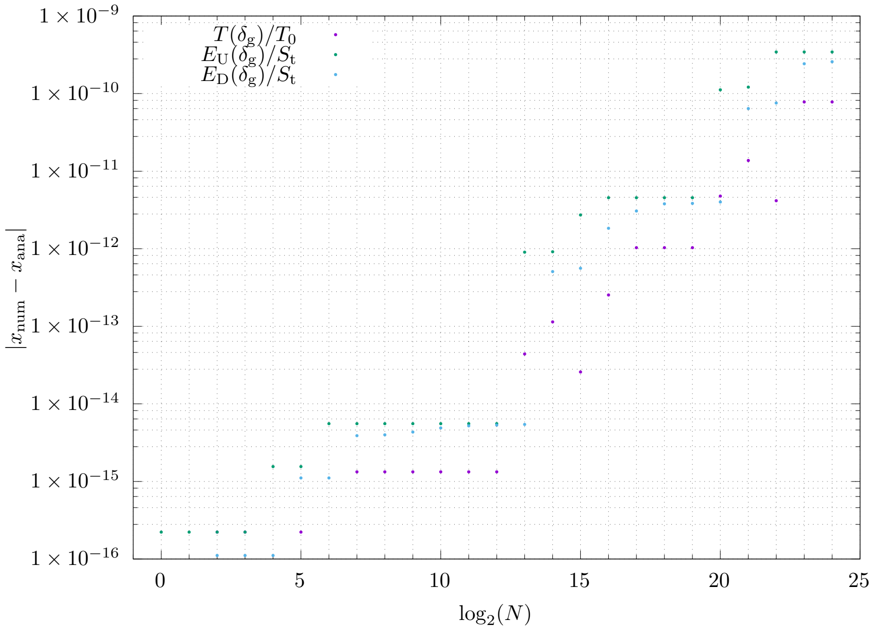

Figure 4: Stability of numerical solution for ODEs system in radiative equilibrium with respect to spatial grid size. Errors are negligible up to 4096 layers, then error propagation dominates reducing the precision of the method. Missing points have value 0.

convection being faster than radiative processes. The steady state reached by T when the two processes compensate is called radiative-convective equilibrium.

Convection is introduced in the model by imposing values of T such that the lapse rate is always greater or equal than some critical lapse rate. The substitution of T with prescribed values is justified by the difference in time scales of the processes: convective equilibrium is reached before radiative processes are able to change T. This procedure is called convective adjustment.

In particular for this RCM, a constant value Γ_0 is used as critical lapse rate and T for layers in interval $(z_{\mathrm{g}}, z_{\mathrm{TOA}})$ is set through an iterative procedure starting from ground level. The lapse rate of each atmopsheric layer is calculated using values of the actual and previous layers, then if condition $-\frac{\partial T}{\partial z} > \Gamma_0$ is met, T of the actual layer is set to satisfy $-\frac{\partial T}{\partial z} = \Gamma_0$. Convective adjustment is applied at each temporal step of the numerical procedure presented in section 3.2, after the evaluation of the temperature profile in radiative equilibrium.

Steady states for T and irradiances after convective adjustment are shown in figures 1 and 2, respectively. Although values are affected by errors derived from the numerical method, two regions can be isolated in the temperature profile: one in radiative-convective equilibrium at low altitudes, called troposhere, the other at higher altitudes where radiative processes dominate the heat transfer, called stratosphere. The separation between these regions goes under the name tropopause. Convective adjustment acts on irradiances indirectly, because they are evaluated at each time step using a modified temperature profile.

5 Conclusion

In this work the vertical temperature profile of a grey atmosphere is studied under hypotheses which simplify convective and radiative processes. The problem is defined in general by a system of PDEs and at the steady state by a system of ODEs. Numerical solution is evaluated for the planet in radiative equilibrium and is compared with analytical values. Errors are negligible, except for the ones in temporal integration of the PDEs. Convection is recovered using convective adjustment to set

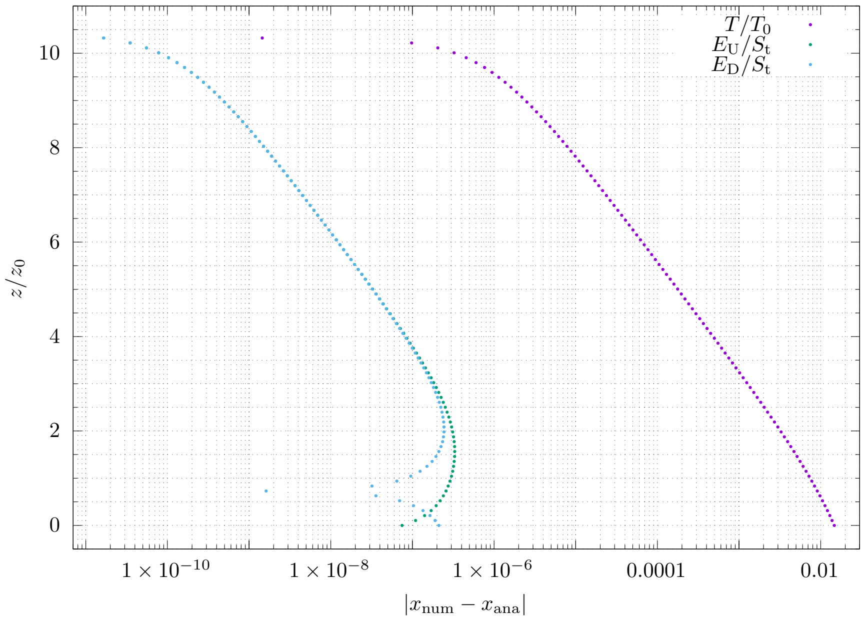

Figure 5: Errors between numerical and analytical solutions of a grey atmosphere in radiative equilibrium. Steady state is reached throught iterative temporal and spatial integrations. Precision is reduced by propagation of errors during successive iterations. Points at TOA are omitted being compatible with 0 within precision of double-precision floating-point numbers.

temperature values according to a prescribed lapse rate.

The RCM can be improved in various ways. First of all, differences in composition and properties of the atmosphere can be considered, e.g. different chemical species absorbing radiation, non-constant values of lapse rate and albedo. Thus, interesting effects which characterise planetary atmospheres can be studied, e.g. response of the atmosphere to changes in composition, more realistic stratosphere and greenhouse effect. In addition, overall energy conservation is not checked during time integration, hence more advanced procedures of convective adjustment can be adopted to account for conservation of quantities. Precision can be improved by using different integration methods, for instance implicit methods where PDEs are discretised simultaneously on the spatial and temporal dimensions.

A Source code

Calculations are performed with code in file `main.cpp` (cf. listing 1). Header `constants.h` provides values of constants listed in table 1, it is shown in listing 2 for completeness. Functions for ODEs integration are stored in file `mclib.cpp` which is not shown, since the functions are available in any library for numerical solution of ODEs.

Table 1: Data on constants used in the present work. The middle rule separates standard values on top from arbitrary values chosen for the present work on bottom.

Symbol	Value	Unit	Notes
A	0.3		Bond albedo value of Earth compatible with various observations, cf. [3, p. 1281]
c	2.99792458×10^8	m/s	Speed of light in vacuum
c_P	1.004×10^3	J/(K kg)	Specific heat at constant pressure of air, from [4, p. 16]
δ_{TOA}	0		Optical depth at TOA, by definition
D	1.66		Diffusion coefficient, commonly used value from [4, p. 55]
h	$6.62607015 \times 10^{-34}$	J s	Planck constant
g	9.80665	m/s^2	Standard gravitational acceleration of Earth
Γ_0	6.5×10^{-3}	K/m	Environmental lapse rate of Earth's troposphere, from [5, p. 3]
k_{B}	1.380649×10^{-23}	J/K	Boltzmann constant
P_{g}	1.013250×10^5	Pa	Standard pressure at ground level of Earth, from [5, p. 2]
R_{m}	2.8705287×10^2	J/(K kg)	Specific gas constant of dry air
σ	$5.670374419 \times 10^{-8}$	W/(m^2 K^4)	Stefan-Boltzmann constant
S_0	1361.0	W/m^2	Nominal total solar irradiance, from [6]
T_{g}	288.15	K	Earth's surface temperature based on [5, p. 2]
z_{g}	0	m	Nominal ground level
δ_{g}	$\frac{1}{D}\left(\frac{2\sigma T_{\mathrm{g}}^4}{S_{\mathrm{t}}} - 2\right)$		Optical depth at ground level
Δt	864 000	s	Time step for temporal integration
μ_{m}	$\frac{\delta_{\mathrm{g}} g}{P_{\mathrm{g}} - P_{\mathrm{TOA}}}$	m^2/kg	Mass attenuation coefficient of the atmosphere
N	100		Number of atmospheric layers
N_{max}	2^{24}		Maximum number of atmospheric layers used for stability analysis
P_0	1×10^5	Pa	Arbitrary reference value for pressure
P_{TOA}	3	Pa	Arbitrary TOA pressure
S_{t}	$(1 - A)\frac{S_0}{4}$	W/m^2	Irradiance transmitted from outer space to TOA
T_0	$\left(\frac{S_{\mathrm{t}}}{\sigma}\right)^{\frac{1}{4}}$	K	Arbitrary reference value for temperature
z_0	2000	m	Arbitrary constant for normalisation of altitude
z_{TOA}	$z_{\mathrm{g}} - z_0 \ln\left(\frac{P}{P_{\mathrm{g}}}\right)$	m	Arbitrary TOA altitude

```cpp
#include <cmath>
#include <fstream>
#include <iomanip>
#include <iostream>

#include "constants.h"
#include "../../mclib/mclib.h"

#ifdef N_PRECISION
#undef N_PRECISION
#endif /* N_PRECISION */

#define DIR_DATA "../data"
#define N_PRECISION 6
#define TOLERANCE 1e-6
#define N_STABILITY 25

int const global_N = 100;
double const global_P_0 = 1e5; // Pa
double const global_P_TOA = 3.0; // Pa
double const global_z_0 = 2000.0; // m

double const global_S_t = 0.25 * (1.0 - const_A) * const_S_0; // / (W / m^2)
double const global_delta_g = 2.0 * (const_sigma / global_S_t * const_T_g*const_T_g*const_T_g*const_T_g - 1.0) / const_D;
double const global_mu_m = global_delta_g * const_g / (const_P_g - global_P_TOA); // / (m^2 / kg)
double * global_delta, * global_sigma;
double * global_P; // / Pa
double global_T_0; // / K
double * global_z; // / m

double get_altitude(double P);
double get_pressure(double z);
double get_optical_depth_z(double z, double P_TOA);
double get_sigma(double P, double P_TOA);
double get_theta(double T, double P);
double temperature_norm(double delta);
double irradiance_upward_norm(double delta);
double irradiance_downward_norm(double delta);
void rhs_delta(double t, double const * Y_0, double * R);
void rhs_t(double t, double const * Y_0, double * R);

int main(int argc, char * argv[]) {
  using namespace std;
  cout << fixed << setprecision(N_PRECISION);

  // Configure vertical coordinates.
  double z_TOA, dz; // / m
  z_TOA = get_altitude(global_P_TOA);
  dz = (z_TOA - const_z_g) / global_N;
  global_z = new double[global_N + 1];
  global_delta = new double[global_N + 1];
  global_P = new double[global_N + 1];
  global_sigma = new double[global_N + 1];
  for (int i = 0; i <= global_N; i++) {
    global_z[i] = z_TOA - i * dz;
    global_delta[i] = get_optical_depth_z(global_z[i], global_P_TOA);
    global_P[i] = get_pressure(global_z[i]);
    global_sigma[i] = get_sigma(global_P[i], global_P_TOA);
  }

```

```cpp


/* Analytical solution in radiative equilibrium */

cout << "Analytical solution in radiative equilibrium" << endl;

// Prepare variables.
double * Y_3_ana, * theta_norm, * Y_1_ana, * Y_2_ana;
Y_3_ana = new double[global_N + 1];
theta_norm = new double[global_N + 1];
Y_1_ana = new double[global_N + 1];
Y_2_ana = new double[global_N + 1];
global_T_0 = pow(global_S_t / const_sigma, 0.25);

// Prepare output files.
ofstream file_temperature, file_irradiance;
char fn_temperature_analytical[] = DIR_DATA "/temperature_analytical.dat";
char fn_irradiance_analytical[] = DIR_DATA "/irradiance_analytical.dat";
file_temperature << fixed << setprecision(N_PRECISION);
file_temperature.open(fn_temperature_analytical);
file_temperature << "#z/z_0 T/T_0 delta P sigma theta/T_0" << endl;
file_temperature << "#'1' '1' '1' 'Pa' '1' '1'" << endl;
file_irradiance << fixed << setprecision(N_PRECISION);
file_irradiance.open(fn_irradiance_analytical);
file_irradiance << "#z/z_0 E_U/S_t E_D/S_t delta P sigma" << endl;
file_irradiance << "#'1' '1' '1' '1' 'Pa' '1'" << endl;

// Plot analytical solutions.
for (int i = 0; i <= global_N; i++) {
  Y_3_ana[i] = temperature_norm(global_delta[i]);
  theta_norm[i] = get_theta(Y_3_ana[i], global_P[i]);
  file_temperature << global_z[i] / global_z_0 << ' '
    << Y_3_ana[i] << ' '
    << global_delta[i] << ' '
    << global_P[i] << ' '
    << global_sigma[i] << ' '
    << theta_norm[i] << '\n';
  Y_1_ana[i] = irradiance_upward_norm(global_delta[i]);
  Y_2_ana[i] = irradiance_downward_norm(global_delta[i]);
  file_irradiance << global_z[i] / global_z_0 << ' '
    << Y_1_ana[i] << ' '
    << Y_2_ana[i] << ' '
    << global_delta[i] << ' '
    << global_P[i] << ' '
    << global_sigma[i] << '\n';
}
file_temperature.close();
cout << "- Temperature stored in file " << fn_temperature_analytical << endl;
file_irradiance.close();
cout << "- Irradiances stored in file " << fn_irradiance_analytical << endl;



/* Numerical solution in radiative equilibrium */

cout << "Numerical solution in radiative equilibrium" << endl;

// Prepare variables.
int i_t, is_steady;
double * Y_3, * Y_3_prev, * Y_1, * Y_2;
double t; // / s
double Y[3];
```

```
double Y_3_tmp, theta_tmp, theta_PDE_tmp;
double Delta_t; // s
Y_3 = new double[3 * (global_N + 1)]; // Use to store irradiances.
Y_3_prev = new double[global_N + 1];
Y_1 = Y_3 + global_N + 1;
Y_2 = Y_1 + global_N + 1;
Delta_t = 10 * 24 * 3600.0;

// Set initial values PDEs.
i_t = 0;
Y_3[0] = temperature_norm(const_delta_TOA);
for (int i = 1; i <= global_N; i++) {
  Y_3[i] = const_T_g / global_T_0;
}
Y_1[0] = 1.0;
Y_2[0] = 0.0;

// Integrate PDEs.
do {
  Y_3_prev[0] = Y_3[0];
  i_t++;
  t = i_t * Delta_t;
  Y[1] = 1.0;
  Y[2] = 0.0;
  for (int i = 1; i <= global_N; i++) {
    Y_3_prev[i] = Y_3[i];
    Y[0] = Y_3[i]*Y_3[i]*Y_3[i]*Y_3[i];
    rungekutta4(global_delta[i], global_delta[i] - global_delta[i-1], Y, rhs_delta, 3);
    Y_1[i] = Y[1];
    Y_2[i] = Y[2];
  }
  eulerstep(t, Delta_t, Y_3, rhs_t, global_N + 1);
  // Temperature profile steady state condition.
  is_steady = 1;
  for (int i = 0; i <= global_N; i++) {
    if (fabs(Y_3[i] - Y_3_prev[i]) > TOLERANCE) {
      is_steady = 0;
      break;
    }
  }
} while (! is_steady);
cout << "- Steady state reached in " << i_t << " iterations" << endl;

// Prepare output files.
char fn_temperature_numerical[] = DIR_DATA "/temperature_numerical.dat";
char fn_irradiance_numerical[] = DIR_DATA "/irradiance_numerical.dat";
file_temperature << fixed << setprecision(N_PRECISION);
file_temperature.open(fn_temperature_numerical);
file_temperature << "#z/z_0 T/T_0 T_PDE/t_0 T_err/T_0 T_PDE_err/T_0 delta P sigma theta/T_0 theta_PDE/T_0 theta_err/T_0 theta_PDE_err/T_0" << endl;
file_temperature << "#'1' '1' '1' '1' '1' '1' 'Pa' '1' '1' '1' '1' '1'" << endl;
file_irradiance << fixed << setprecision(N_PRECISION);
file_irradiance.open(fn_irradiance_numerical);
file_irradiance << "#z/z_0 E_U/S_t E_U_PDE/S_t E_D/S_t E_D_PDE/S_t E_U_err/S_t E_U_PDE_err/S_t E_D_err/S_t E_D_PDE_err/S_t delta P sigma" << endl;
file_irradiance << "#'1' '1' '1' '1' '1' '1' '1' '1' '1' '1' 'Pa' '1'" << endl;

// Set initial values ODEs.
Y[0] = 0.5;
Y[1] = 1.0;
Y[2] = 0.0;
```

```
Y_3_tmp = pow(Y[0], 0.25);
theta_tmp = get_theta(Y_3_tmp, global_P[0]);
theta_PDE_tmp = get_theta(Y_3[0], global_P[0]);
file_temperature << global_z[0] / global_z_0 << ' '
  << Y_3_tmp << ' '
  << Y_3[0] << ' '
  << scientific << fabs(Y_3_tmp - Y_3_ana[0]) << ' '
  << fabs(Y_3[0] - Y_3_ana[0]) << fixed << ' '
  << global_delta[0] << ' '
  << global_P[0] << ' '
  << global_sigma[0] << ' '
  << theta_tmp << ' '
  << theta_PDE_tmp << ' '
  << scientific << fabs(theta_tmp - theta_norm[0]) << ' '
  << fabs(theta_PDE_tmp - theta_norm[0]) << fixed << '\n';
file_irradiance << global_z[0] / global_z_0 << ' '
  << Y[1] << ' '
  << Y_1[0] << ' '
  << Y[2] << ' '
  << Y_2[0] << ' '
  << scientific << fabs(Y[1] - Y_1_ana[0]) << ' '
  << fabs(Y_1[0] - Y_1_ana[0]) << ' '
  << fabs(Y[2] - Y_2_ana[0]) << ' '
  << fabs(Y_2[0] - Y_2_ana[0]) << fixed << ' '
  << global_delta[0] << ' '
  << global_P[0] << ' '
  << global_sigma[0] << '\n';

// Integrate ODEs.
for (int i = 1; i <= global_N; i++) {
  rungekutta4(global_delta[i], global_delta[i] - global_delta[i-1], Y, rhs_delta, 3);
  Y_3_tmp = pow(Y[0], 0.25);
  theta_tmp = get_theta(Y_3_tmp, global_P[i]);
  theta_PDE_tmp = get_theta(Y_3[i], global_P[i]);
  file_temperature << global_z[i] / global_z_0 << ' '
    << Y_3_tmp << ' '
    << Y_3[i] << ' '
    << scientific << fabs(Y_3_tmp - Y_3_ana[i]) << ' '
    << fabs(Y_3[i] - Y_3_ana[i]) << fixed << ' '
    << global_delta[i] << ' '
    << global_P[i] << ' '
    << global_sigma[i] << ' '
    << theta_tmp << ' '
    << theta_PDE_tmp << ' '
    << scientific << fabs(theta_tmp - theta_norm[i]) << ' '
    << fabs(theta_PDE_tmp - theta_norm[i]) << fixed << '\n';
  file_irradiance << global_z[i] / global_z_0 << ' '
    << Y[1] << ' '
    << Y_1[i] << ' '
    << Y[2] << ' '
    << Y_2[i] << ' '
    << scientific << fabs(Y[1] - Y_1_ana[i]) << ' '
    << fabs(Y_1[i] - Y_1_ana[i]) << ' '
    << fabs(Y[2] - Y_2_ana[i]) << ' '
    << fabs(Y_2[i] - Y_2_ana[i]) << fixed << ' '
    << global_delta[i] << ' '
    << global_P[i] << ' '
    << global_sigma[i] << '\n';
}
file_temperature.close();
cout << "- Temperature stored in file " << fn_temperature_numerical << endl;
file_irradiance.close();
```

```
cout << "- Irradiances stored in file " << fn_irradiance_numerical << endl;



/* Stability analysis of numerical solution in radiative equilibrium */

cout << "Stability analysis of numerical solution in radiative equilibrium" << endl;

// Prepare output file.
ofstream file_stability;
char fn_stability[] = DIR_DATA "/stability.dat";
file_stability << scientific << setprecision(N_PRECISION);
file_stability.open(fn_stability);
file_stability << "#N T_err(delta_g)/T_0 E_U_err(delta_g)/S_t E_D_err(delta_g)/S_t" <<
   endl;
file_stability << "#'1' '1' '1' '1'" << endl;

// Integrate up to ground level.
int n;
for (int i = 0; i < N_STABILITY; i++) {
  n = 1 << i;
  Y[0] = 0.5;
  Y[1] = 1.0;
  Y[2] = 0.0;
  integrate_IVP(n, (global_delta_g - const_delta_TOA) / n, Y, rhs_delta, 3,
  rungekutta4);
  Y_3_tmp = pow(Y[0], 0.25);
  file_stability << n << ' '
    << fabs(Y_3_tmp - Y_3_ana[global_N]) << ' '
    << fabs(Y[1] - Y_1_ana[global_N]) << ' '
    << fabs(Y[2] - Y_2_ana[global_N]) << '\n';
}
file_stability.close();
cout << "- Errors stored in file " << fn_stability << endl;



/* Radiative-convective equilibrium */

cout << "Radiative-convective equilibrium" << endl;

// Set initial values.
i_t = 0;
Y_3[0] = temperature_norm(const_delta_TOA);
for (int i = 1; i <= global_N; i++) {
  Y_3[i] = const_T_g / global_T_0;
}
Y_1[0] = 1.0;
Y_2[0] = 0.0;

// Run model.
do {
  Y_3_prev[0] = Y_3[0];
  i_t++;
  t = i_t * Delta_t;
  Y[1] = 1.0;
  Y[2] = 0.0;
  for (int i = 1; i <= global_N; i++) {
    Y_3_prev[i] = Y_3[i];
    Y[0] = Y_3[i]*Y_3[i]*Y_3[i]*Y_3[i];
    rungekutta4(global_delta[i], global_delta[i] - global_delta[i-1], Y, rhs_delta, 3)
  ;
```

```
      Y_1[i] = Y[1];
      Y_2[i] = Y[2];
    }
    eulerstep(t, Delta_t, Y_3, rhs_t, global_N + 1);
    // Convective adjustment.
    for (int i = global_N - 1; i > 0; i--) {
      if (- global_T_0 * (Y_3[i] - Y_3[i + 1]) / (global_z[i] - global_z[i + 1]) > const_Gamma_0) {
        Y_3[i] = Y_3[i + 1] - (global_z[i] - global_z[i + 1]) * const_Gamma_0 / global_T_0;
      }
    }
    // Temperature profile steady state condition.
    is_steady = 1;
    for (int i = 0; i <= global_N; i++) {
      if (fabs(Y_3[i] - Y_3_prev[i]) > TOLERANCE) {
        is_steady = 0;
        break;
      }
    }
  } while (! is_steady);
  cout << "- Steady state reached in " << i_t << " iterations" << endl;

  // Prepare output files.
  char fn_temperature_RCM[] = DIR_DATA "/temperature_RCM.dat";
  char fn_irradiance_RCM[] = DIR_DATA "/irradiance_RCM.dat";
  file_temperature << fixed << setprecision(N_PRECISION);
  file_temperature.open(fn_temperature_RCM);
  file_temperature << "#z/z_0 T/T_0 T_err/T_0 delta P sigma theta/T_0 theta_err/T_0" << endl;
  file_temperature << "#'1' '1' '1' '1' 'Pa' '1' '1' '1'" << endl;
  file_irradiance << fixed << setprecision(N_PRECISION);
  file_irradiance.open(fn_irradiance_RCM);
  file_irradiance << "#z/z_0 E_U/S_t E_D/S_t E_U_err/S_t E_D_err/S_t delta P sigma" << endl;
  file_irradiance << "#'1' '1' '1' '1' '1' '1' 'Pa' '1'" << endl;

  // Print output values.
  for (int i = 0; i <= global_N; i++) {
    theta_tmp = get_theta(Y_3[i], global_P[i]);
    file_temperature << global_z[i] / global_z_0 << ' '
      << Y_3[i] << ' '
      << scientific << fabs(Y_3[i] - Y_3_ana[i]) << fixed << ' '
      << global_delta[i] << ' '
      << global_P[i] << ' '
      << global_sigma[i] << ' '
      << theta_tmp << ' '
      << scientific << fabs(theta_tmp - theta_norm[i]) << fixed << '\n';
    file_irradiance << global_z[i] / global_z_0 << ' '
      << Y_1[i] << ' '
      << Y_2[i] << ' '
      << scientific << fabs(Y_1[i] - Y_1_ana[i]) << ' '
      << fabs(Y_2[i] - Y_2_ana[i]) << fixed << ' '
      << global_delta[i] << ' '
      << global_P[i] << ' '
      << global_sigma[i] << '\n';
  }
  file_temperature.close();
  cout << "- Temperature stored in file " << fn_temperature_RCM << endl;
  file_irradiance.close();
  cout << "- Irradiances stored in file " << fn_irradiance_RCM << endl;
```

```
  // Tear down.
  delete[] global_z;
  delete[] global_P;
  delete[] global_delta;
  delete[] global_sigma;
  delete[] Y_3_ana;
  delete[] theta_norm;
  delete[] Y_1_ana;
  delete[] Y_2_ana;
  delete[] Y_3;
  delete[] Y_3_prev;

  return 0;
}

double get_altitude(double P) {
  return const_z_g - global_z_0 * log(P / const_P_g);
}

double get_pressure(double z) {
  return const_P_g * exp(- (z - const_z_g) / global_z_0);
}

double get_optical_depth_z(double z, double P_TOA) {
  return global_mu_m / const_g * (get_pressure(z) - P_TOA);
}

double get_sigma(double P, double P_TOA) {
  return (P - P_TOA) / (const_P_g - P_TOA);
}

double get_theta(double T, double P) {
  return T * pow(global_P_0 / P, const_R_m / const_c_P);
}

double temperature_norm(double delta) {
  return pow(0.5 * (1.0 + const_D * delta), 0.25);
}

double irradiance_upward_norm(double delta) {
  return 0.5 * (2.0 + const_D * delta);
}

double irradiance_downward_norm(double delta) {
  return 0.5 * const_D * delta;
}

void rhs_delta(double t, double const * Y_0, double * R) {
  R[0] = 0.5 * const_D;
  R[1] = const_D * (Y_0[1] - Y_0[0]);
  R[2] = const_D * (Y_0[0] - Y_0[2]);
}

void rhs_t(double t, double const * Y_0, double * R) {
  double const * Y_1, * Y_2;
  Y_1 = Y_0 + global_N + 1;
  Y_2 = Y_1 + global_N + 1;
  R[0] = 0.0;
  for (int i = 1; i <= global_N; i++) {
    R[i] = global_S_t * global_mu_m / (const_c_P * global_T_0) * (Y_1[i] - Y_1[i-1] -
    Y_2[i] + Y_2[i-1]) / (global_delta[i] - global_delta[i-1]);
  }
```

```
}
```

Listing 1: File `main.cpp`.

```
#ifndef CONSTANTS_H
#define CONSTANTS_H

double const const_A = 0.3;
double const const_c = 2.99792458e8; // m / s
double const const_c_P = 1.004e3; // J / (K kg)
double const const_delta_TOA = 0.0;
double const const_D = 1.66;
double const const_h = 6.62607015e-34; // J s
double const const_g = 9.80665; // m / s^2
double const const_Gamma_0 = 6.5e-3; // K / m
double const const_k_B = 1.380649e-23; // J / K
double const const_P_g = 1.013250e5; // Pa
double const const_R_m = 2.8705287e2; // J / (K kg)
double const const_sigma = 5.670374419e-8; // W / (m^2 K^4)
double const const_S_0 = 1361.0; // W / m^2
double const const_T_g = 288.15; // K
double const const_z_g = 0.0; // m

#endif /* CONSTANTS_H */
```

Listing 2: File `constants.h`.

B Mathematical derivations

In this appendix mathematical derivations of some ancillary results and formulae used in the main text are explicitly shown.

B.1 Relation between pressure and altitude

A general result regarding planetary atmospheres is that atmospheric pressure decreases with increasing altitude. Theoretical relations which approximate this behaviour can be obtained. Hypotheses considered in section 1 are valid.

If density is assumed constant, equation (2) can be solved easily resulting in a linear dependence of pressure P on altitude z,

$$P(z) = P_0 - \varrho g(z - z_0) \quad , \tag{29}$$

where (z_0, P_0) is a reference point inside the atmosphere.

If density is not constant its expression is given by the ideal gas law (cf. equation (4)) and, assuming constant temperature T, equation (2) becomes

$$\begin{aligned} \mathrm{d}P = -\frac{Pg}{R_\mathrm{m}T}\,\mathrm{d}z \iff \\ \iff \frac{\mathrm{d}P}{P} = -\frac{g}{R_\mathrm{m}T}\,\mathrm{d}z \end{aligned} \tag{30}$$

with solution

$$\begin{aligned} \ln(P')\Big|_{P_0}^{P(z)} = -\frac{g}{R_\mathrm{m}T}z'\Big|_{z_0}^{z} \iff \\ \iff P(z) = P_0 \exp\left(-\frac{g}{R_\mathrm{m}T}(z - z_0)\right) \quad . \end{aligned} \tag{31}$$

This relation is not meaningful if used at every z, since the aim of the work is to derive the non-constant temperature profile of the atmosphere. However, it can be used inside atmospheric layers where the temperature is considered constant (e.g. stratosphere).

A better approximation assumes non-constant density and constant lapse rate Γ, hence temperature depends linearly on altitude,

$$\Gamma = -\frac{\mathrm{d}T}{\mathrm{d}z} \impliedby T(z) = T_0 - \Gamma(z - z_0) \quad , \tag{32}$$

with T_0 temperature corresponding to reference altitude z_0. Using these assumptions and the density rewritten through the ideal gas law (4), equation (2) becomes

$$\begin{aligned} \mathrm{d}P = -\frac{Pg}{R_\mathrm{m}T}\left(-\frac{\mathrm{d}T}{\Gamma}\right) \iff \\ \iff \frac{\mathrm{d}P}{P} = \frac{g}{R_\mathrm{m}\Gamma}\frac{\mathrm{d}T}{T} \quad , \end{aligned} \tag{33}$$

which has solution

$$\begin{aligned} \ln(P')\Big|_{P_0}^{P(z)} = \frac{g}{R_\mathrm{m}\Gamma}\ln(T')\Big|_{T_0}^{T(z)} \iff \\ \iff P(z) = P_0\left(\frac{T_0 - \Gamma(z - z_0)}{T_0}\right)^{\frac{g}{R_\mathrm{m}\Gamma}} \quad . \end{aligned} \tag{34}$$

Equation (34) can be used also with a piecewise constant lapse rate in altitude intervals where it is not null. Otherwise, in altitude intervals where lapse rate is null, equation (31) is valid with appropriate boundary conditions to ensure continuity between layers.

B.2 Radiometric quantities

Refer to [7] and [1] for more details on quantities reviewed in this section.

Consider electromagnetic radiation emitted by a point source. The total emitted power is called *radiant flux*, with unit W. The density of radiant flux with respect to a solid angle in the direction of emission is called *radiant intensity*, expressed in W/sr. When radiation interacts with a surface, i.e. it gets absorbed, transmitted or reflected, its radiant intensity distributed over the surface is measured through *radiance* in $\mathrm{W/(m^2\,sr)}$. If the area on which the radiation is incident is expressed through the solid angle it subtends, the integral of radiance over this solid angle is called *irradiance*, expressed in $\mathrm{W/m^2}$. Note that the coordinate system where the solid angles of radiant intensity and irradiance are defined may not be the same. Radiant flux emitted by a body normalised over the surface of emission is measured by *radiant exitance* in $\mathrm{W/m^2}$.

All previous quantities can be expressed as densities with respect to the wavelength or the wavenumber and the adjective *spectral* is prefixed to their names. Their units are divided by the respective spectral quantity (e.g. spectral radiance with wavenumber in 1/cm has units $\mathrm{W\,cm/(m^2\,sr)}$).

Spectral radiance of a blackbody is given by Planck's law

$$B_\nu(\nu, T) = 2hc^2\nu^3 \frac{1}{e^{\frac{hc\nu}{k_B T}} - 1} \quad , \tag{35}$$

where ν is the wavenumber in unit 1/m, T in unit K is the temperature of the emitting body and the other quantities are constants (cf. table 1). Note that Planck's law has different form when it is expressed in terms of wavelength, due to its definition as density and the resulting change of variables:

$$B_\lambda(\lambda, T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda k_B T}} - 1} \quad . \tag{36}$$

If radiance is isotropic, i.e. it is not dependent on the direction of the radiation, the corresponding irradiance is proportional. For instance, if the radiation is absorbed by a hemispheric surface approximated by a blackbody, the spectral irradiance of the surface is

$$\begin{aligned}
&\int B_\nu(\nu, T)\,\mathrm{d}\phi \sin(\theta)\,\mathrm{d}\theta \cos(\theta) = \\
&= B_\nu(\nu, T) \int_0^{2\pi} \mathrm{d}\phi \int_0^{\frac{\pi}{2}} \sin(\theta)\cos(\theta)\,\mathrm{d}\theta = \\
&= 2\pi B_\nu(\nu, T) \int_0^1 \sin(\theta)\,\mathrm{d}(\sin(\theta)) = \\
&= 2\pi B_\nu(\nu, T) \frac{1}{2} = \\
&= \pi B_\nu(\nu, T) \quad ,
\end{aligned} \tag{37}$$

where spherical coordinates are used to describe the surface and the term $\cos(\theta)$ considers the component of radiation along the normal of the infinitesimal solid angle.

B.3 Radiation attenuation

Details on quantities present in this section can be found in [1, p. 285]. Radiation crossing a medium loses energy due to absorption and scattering. The effect of chemical species on these processes is quantified through the *attenuation coefficient* (commonly called *extinction coefficient* in atmospheric sciences), which has different definitions based on the way it is derived (cf. [4, p. 44]). The attenuation coefficient is the sum of *absorption coefficient* and *scattering coefficient* which contain information on the attenuation due to the respective physical processes. The *optical depth* (also called *optical thickness*) takes in consideration the amount of substance involved in the absorption.

Ratios between radiant fluxes are related by the conservation of energy: the sum of *internal transmittance* and *internal absorptance* is 1, as well as the sum of *reflectance*, *absorptance* and *transmittance*.

In general these coefficients are functions of wavelength or wavenumber, in which case the prefix *spectral* is adopted. If the medium is a fluid, they depends on temperature and pressure of the medium.

The spectral internal transmittance is defined as

$$\tau_\mathrm{i}(\nu, s, s_0) = e^{-\delta(\nu, s, s_0)} \quad , \tag{38}$$

where $\delta(\nu, s, s_0)$ is the spectral optical depth, which depends only on the spectral attenuation coefficient $\mu(\nu, s')$ of the medium traversed by the radiation from s_0 to s on the optical path. In the RCM optical paths are straight and form an angle θ with the direction normal to the layers, hence the definition of the spectral optical depth becomes:

$$\delta(\nu, s, s_0) = \frac{1}{\cos\theta} \int_{s_0}^{s} \mu(\nu, s')\,\mathrm{d}s' \quad . \tag{39}$$

If the absorbing species do not interact, $\mu(\nu, s')$ is simply the sum of the spectral attenuation coefficients of the individual components of the medium.

Moreover, if the medium is homogeneous, in the sense that quantities affecting radiative calculations are not dependent on spatial position (e.g. attenuation coefficients μ are constant inside the medium), the spectral attenuation coefficient depends only on the concentration of the absorbing species, hence the spectral attenuation coefficient can be rewritten as

$$\mu(\nu, s') = \mu_\mathrm{m}(\nu)\rho(s') \tag{40}$$

where $\mu_\mathrm{m}(\nu)$ is the spectral mass attenuation coefficient and $\rho(s')$ is the volumetric mass density of the absorber.

Names of radiative properties ending with suffix *-ance* are generally used for rough surfaces, while suffix *-ivity* indicates smooth surfaces. In this work the former is adopted. Refer to [1, p. 59] for more information and to the definition of spectral absorptivity in [7] for an example of the difference.

B.4 Quantities commonly used in atmospheric sciences

Earth's surface horizontal profile is not uniform, hence altitude and pressure near ground level could present sudden variations. In models where this is taken into consideration, the sigma coordinate system is commonly used instead, defined by

$$\sigma = \frac{P - P_{\mathrm{TOA}}}{P_{\mathrm{g}} - P_{\mathrm{TOA}}} \quad . \tag{41}$$

To avoid confusion, in this work symbol σ is used for the Stefan-Boltzmann constant (cf. table 1), except for equation (41).

An alternative quantity evaluated in place of $T(t, z)$ for a given parcel of fluid is the potential temperature

$$\theta(t, z) = T(t, z)\left(\frac{P_0}{P(z)}\right)^{\frac{R_{\mathrm{m}}}{c_P}} \quad , \tag{42}$$

where P_0 is a reference pressure and quantities $P(z)$, R_{m} and c_P refer to the fluid.

References

[1] M. F. Modest and S. Mazumder, *Radiative Heat Transfer*, 4th ed. Academic Press, 2021.

[2] V. Ramanathan and J. A. Coakley Jr., "Climate modeling through radiative-convective models," *Reviews of Geophysics*, vol. 16, no. 4, pp. 465–489, 1978. DOI: `https://doi.org/10.1029/RG016i004p00465`. eprint: `https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/RG016i004p00465`. [Online]. Available: `https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/RG016i004p00465`.

[3] J. E. Harries and C. Belotti, "On the variability of the global net radiative energy balance of the nonequilibrium earth," *Journal of Climate*, vol. 23, no. 6, pp. 1277–1290, 2010. DOI: `https://doi.org/10.1175/2009JCLI2797.1`. [Online]. Available: `https://journals.ametsoc.org/view/journals/clim/23/6/2009jcli2797.1.xml`.

[4] D. C. Catling and J. F. Kasting, *Atmospheric evolution on inhabited and lifeless worlds.* Cambridge University Press, 2017, ISBN: 9781316825488.

[5] N. Oceanic and A. Administration, "Us standard atmosphere, 1976," *Technical Report*, 1976.

[6] A. Prša, P. Harmanec, G. Torres, *et al.*, "Nominal values for selected solar and planetary quantities: Iau 2015 resolution b3," *The Astronomical Journal*, vol. 152, no. 2, p. 41, Aug. 2016. DOI: `10.3847/0004-6256/152/2/41`. [Online]. Available: `https://dx.doi.org/10.3847/0004-6256/152/2/41`.

[7] International Commission on Illumination, "CIE S 017:2020 ILV: International Lighting Vocabulary, 2nd edition," International Commission on Illumination, Standard, 2020.