

# Radiative Transfer Model 3.0 integrated into the PALM model system 6.0

## Supplements

Pavel Krč<sup>1</sup>, Jaroslav Resler<sup>1</sup>, Matthias Sühling<sup>2</sup>, Sebastian Schubert<sup>3</sup>, Mohamed H. Salim<sup>3,4</sup>, and Vladimír Fuka<sup>5</sup>

<sup>1</sup>Institute of Computer Science, Czech Academy of Sciences, Prague, Czech Republic

<sup>2</sup>Institute of Meteorology and Climatology, Leibniz University Hannover, Hannover, Germany

<sup>3</sup>Geography Department, Humboldt-Universität zu Berlin, Berlin, Germany

<sup>4</sup>Faculty of Energy Engineering, Aswan University, Aswan, Egypt

<sup>5</sup>Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic

**Correspondence:** Pavel Krč (krc@cs.cas.cz)

### S1 Implementation of RTM

#### S1.1 Basic raytracing algorithm

The initialization step which includes raytracing between faces and establishing view and other required factors is organized in a way which optimizes their actual usage during the simulation time step phase. As the view factor  $\hat{F}_{i \rightarrow j}$  is used to determine the irradiance of face  $j$ , the view factors are computed in the MPI process which controls the subdomain where their *target* faces lie, so that the results of the raytracing do not need to be transferred among processes.

The raytracing process is called within a nested loop. The outer loop iterates over all target faces belonging to local process's subdomain. The inner loop iterates all faces from the whole modelled domain, treating them as sources for raytracing. During the initialization step, an index of all faces and their coordinates is generated and distributed among processes using the MPI `gather` operation. A typical urban scenario has the number of faces proportional to horizontal size of the domain and an index of all faces can fit in each process's memory.

The first test in the inner loop skips any source faces which are not oriented towards the processed target face or this target face is not oriented towards them (see Section 2.1.1). For each remaining source face, the potential view factor value is calculated using (5); this value is valid if the raytracing does not discover any opaque obstacles between the two faces. This value needs to be known in advance because it is used when creating CVF entries (see (19)). In case of legacy discretization with reducing of the number of view factors (Section 2.2.3), the raytracing is skipped for the face pair if the

potential view factor value is less than specified minimum value or if the separation distance is above the specified maximum. Otherwise, the actual raytracing is performed, starting from the source face towards the target face.

Tracing the ray which travels through a regular grid means generating a sorted list of grid boxes that the ray intersects. The raytracing algorithm takes advantage of the regularity, orthogonality and equidistance of the grid. A ray that travels from coordinates  $[x_1, y_1, z_1]$  to coordinates  $[x_2, y_2, z_2]$  necessarily crosses exactly those dividing planes in dimensions  $yz$  that are located between  $x_1 \dots x_2$ , the  $xz$  planes that are located between  $y_1 \dots y_2$  and the  $xy$  planes that are located between  $z_1 \dots z_2$ , only the order in which these crossings are intermixed is not yet known.

Due to computational efficiency, while tracing the ray the RTM uses a coordinate system in which the planes separating grid boxes have integer coordinates. Therefore each grid box centre has coordinates  $[x + \frac{1}{2}, y + \frac{1}{2}, z + \frac{1}{2}]$  for some  $x, y, z \in \mathbb{Z}$ . It means that in case of  $x_1 < x_2$ , the list of crossed  $yz$ -plane boundaries is  $\lceil x_1 \rceil \dots \lfloor x_2 \rfloor$ , and in case of  $x_1 > x_2$ , the list is  $\lfloor x_2 \rfloor \dots \lceil x_1 \rceil$ , and similarly for other dimensions.

The raytracing cycle follows this general pattern:

1. Set current position to the starting point.
2. Determine the Euclidean distance in raytracing coordinates between the starting point and the next separating plane which immediately follows the current position. This is done for each dimension  $x, y$  and  $z$  separately,
3. Select the dimension with the smallest distance to the next separating plane, mark this as the next grid crossing.

4. Identify the grid box between the current position and the next grid crossing.
5. Determine whether that grid box lies above or below the terrain in 2.5-D geometry. In the latter case, the ray is blocked by an opaque obstacle and the raytracing cycle ends prematurely with negative result for the face pair.
6. Perform all tasks scheduled for each visited grid box.
7. Advance the current position to the next grid crossing. For the selected dimension, advance one separating plane further, this will become the next separating plane for the selected dimension.
8. Repeat from step 3 until reaching the target point.

Because the ray might cross multiple separating planes at once and in order to avoid numerical instability, steps 4–6 are skipped if the length of the ray's intersection with the grid box is less than 0.001 in raytracing coordinates. The only limitation of this algorithm is that the whole ray must not lie within one separating plane, but it can be easily demonstrated that this cannot happen when the raytracing is performed between centres of two faces that are oriented towards each other, nor may it happen for rays going to and from grid box centres. The map of terrain elevations for the whole domain, which is needed in the step 5, is represented by a 2-D array. It does fit in the memory of each MPI process and it is prepared in advance using the MPI `gather` operation, therefore the obstacle detection is a fast operation.

## S1.2 Radiation processing in time-stepping

The calculation of radiative fluxes by RTM within time-stepping is much less computationally demanding than calculations performed during the initialization phase. Memory usage is slightly lower than during initialization, because the VF and CVF entries have been fully aggregated and stored in size-optimized arrays.

The RTM time step begins with calculation of values that depend solely on time, like solar geometry and derived values (e.g. precalculated absorptivity values for a prototype semi-transparent box). If the forcing radiative scheme did not provide separate direct and diffuse horizontal irradiances, the global horizontal irradiance is split into these components using the statistically derived Boland model (Boland et al., 2008).

### S1.2.1 First atmospheric pass

The next step is updating radiosities for primary sources of radiation. For surfaces, the LW radiant exitance is calculated using surface temperature and emissivity according to the Stefan–Boltzmann law:

$$M_i = \varepsilon_i \sigma T_i^4,$$

where  $\varepsilon_i$  is the surface emissivity and  $T_i$  is the surface temperature. The assigned radiosity values are exchanged among processes using MPI `gather` operation. After that, the first propagation of radiation throughout the domain from sources to targets (before the first reflection) is simulated. The LW emitted radiant flux is multiplied by irradiance factor values to calculate the irradiance of target faces.

The VF entry cycle is followed by straightforward application of diffuse irradiance with the SVF. The direct solar irradiance is applied using lookup table from apparent solar position angles to precalculated direct solar transmittance entries. The value of direct solar irradiance for face  $i$  is calculated as

$$E_i^d = E^d \frac{\cos \gamma_i}{\cos \theta_s},$$

where  $E^d$  is the current value of horizontal direct solar irradiance,  $\theta_s$  is the solar zenith angle and  $\gamma_i$  is the angle between apparent solar position and the normal of face  $i$ . The fraction is precalculated for the five possible face orientations at the beginning of the radiative time step.

When the MRT calculation is enabled, the direct, diffuse and thermal MRT components are calculated using equivalent entries—MRT factors, SVFs and direct solar transmittances.

The last cycle in the first atmospheric pass before reflections iterates the CVF entries (if plant canopy is present). There are two types of CVF entries: those representing rays from the sky (one for each PCGB) and those representing rays from surfaces (per PCGB and source face). For the CVF entries from the sky, the respective fraction of current diffuse irradiance is absorbed and the fraction of direct solar irradiance is added using lookup table for solar angles.

The CVF entries from faces are used in this step only when plant canopy LW interaction is enabled. In that case, the respective fraction of transmitted thermal radiant flux is absorbed and at the same time, the emitted radiant flux from the PCGB towards the respective face is calculated according to (20), stored and subtracted from the absorbed radiant flux. In case the PCGB and the respective face have identical temperatures, these two fluxes are equal and the net absorbed flux is zero as expected. The sums of radiant fluxes emitted from plant canopy toward each face are then exchanged among processes using MPI `gather` and added to the rest of the incident LW irradiance.

### S1.2.2 Iterative reflections

If reflections are enabled, then the remaining steps are repeated  $n$ -times according to configuration. Each iteration starts with the absorption of the respective fraction (one minus albedo for SW, emissivity for LW) of irradiance from the previous step for each face. The remaining irradiance is reflected—assigned as outgoing radiosity for the current step and exchanged among processes using MPI `gather`. The

outgoing radiosity is then propagated once more through the domain using view factors, again reduced by partial shading according to plant canopy configuration.

The next step is calculation of radiative flux absorbed by plant canopy using CVF entries. In the reflected passes, only the entries for flux originating from surfaces are processed. At the end of each reflective iteration, the MRT factors are processed and the reflected radiation is added to the radiative background for MRT points.

After  $n$  reflective passes, the remaining incident radiant flux is added completely to the absorbed flux of the respective surface and final calculations are done for the timestep, including converting SW and LW radiances to MRT temperature in Kelvins, converting absorbed radiant flux in plant canopy to volumetric heating rates and calculating plant canopy evapotranspiration rates (see Section 4.3).

## S2 Additional examples

### S2.1 Approximation of plant canopy absorption of direct irradiance

As specified in Section 2.4.3, the fraction of absorbed direct solar radiative flux in each PCGB is approximated using a value calculated for a reference leaf area density  $a_r = 0.9 \max\{a_m\}$  for all PCGBs  $m$  in the domain. The coefficient 0.9 was chosen empirically, as described below.

At the beginning of each time step where the sun is above horizon, the apparent solar position is taken from the radiation model and the absorbance  $A_r$  of the prototype PCGB with LAD  $a_r$  is calculated using the sub-grid discretization model described in Section 2.4.3. The value represents an average over the projected area of the PCGB and it is calculated from the average transmittance of the  $n$  rays as  $A_r = 1 - \frac{1}{n} \sum_{l=1}^n T_l$ . Then for each PCGB  $k$  with LAD  $a_k$ , the approximate absorbance  $\tilde{A}_k$  is calculated as

$$\tilde{A}_k = 1 - e^{\frac{a_k}{a_r} \log(1-A_r)}.$$

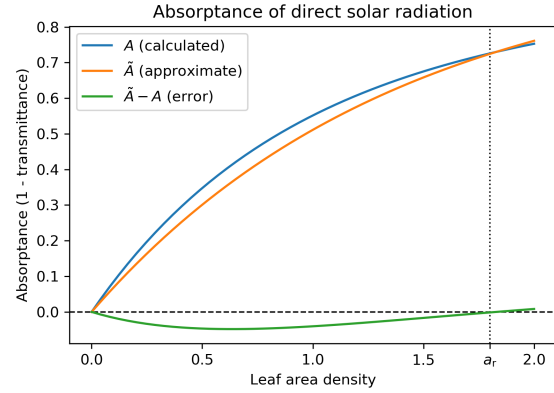
When the rays are parallel to one of the grid axes, all rays intersect the grid cell over identical distance and the approximate value is identical to the true value for any LAD.

The absorbed radiative flux  $\Phi_m^{a,d}$  at PCGB  $k$  from direct solar radiation is then calculated as

$$\Phi_k^{a,d} = E^d T_{D_j \rightarrow k}^r a^d \tilde{A}_k,$$

where  $E^d$  is the current direct normal irradiance,  $T_{D_j \rightarrow k}^r$  is the total ray transmittance for PCGB  $k$  and discretized solar direction  $j$  and  $a^d$  is the current projected area of (any) PCGB in the orthographic projection in the solar direction, which is also calculated by the sub-grid discretization model.

Figure S1 compares absorbance values for all leaf area densities for a reference scenario with maximum LAD  $\max\{a_m\} = 2.0 \text{ m}^{-1}$ , resolution 2 m (i.e. PCGB dimensions



**Figure S1.** Calculated and approximate values of absorbance for a reference urban scenario.

$2 \times 2 \times 2 \text{ m}$ ) and solar zenith angle  $\theta = 25$ . The values  $A$  have been calculated explicitly by the sub-grid model while the values  $\tilde{A}$  have been approximated as described. The approximation error is in the range  $(-0.048, 0.008)$ .

In this example, the chosen reference LAD  $a_r = 0.9 \max\{a_m\} = 1.8 \text{ m}^{-1}$  provides reasonable approximation errors. Because the reference value is based on the maximum LAD value in the scenario, the errors are limited, and because the reference value is less than the maximum value, the errors are both positive and negative. Several scenarios like this were examined and based on the distribution of LAD values, the factor 0.9 was selected as a rather conservative choice which did not cause high errors in any of the examined scenarios.

## References

Boland, J., Ridley, B., and Brown, B.: Models of diffuse solar radiation, *Renew. Energy*, 33, 575–584, 2008.