Forest Light Environmental Simulator (FLiES)

Version 2.4

(version 2.48)

User's Manual

By Hideki Kobayashi

January 11, 2018

Table of Contents

1. <u>Disclaimer</u>

The program code of FLiES was carefully debugged and verified under the author's computational environment. Please take note, however, that this program is provided as is without any guarantees or warranty. The author is not responsible for any damage, or losses of any kind caused by the use or misuse of the program.

This distribution includes the source code so users can modify the source code themselves. However, if you wish to redistribute this program, please consult the author in advance.

<u>If you publish results obtained using this model please cite the paper below.</u>

Kobayashi, H. and H. Iwabuchi (2008), A coupled 1-D atmosphere and 3-D canopy radiative transfer model for canopy reflectance, light environment, and photosynthesis simulation in a heterogeneous landscape, Remote Sensing of Environment, 112, 173-185.

If you find any bugs please contact the developer.

Hideki Kobayashi (developer)
E-mail: hkoba@jamstec.go.jp

2. Introduction

The Forest Light Environmental Simulator (FLiES) is a radiative transfer simulator for computing various kinds of light environments in vegetated surface (forests and grasslands). FLiES has an atmospheric radiative transfer module so that the users can run the model to understand the atmosphere-vegetation interaction and its impact on the light environment in plant canopies. FLiES simulates the bidirectional reflectance factor (BRF) at the top of a forest canopy and at the top of the atmosphere simultaneously. FLiES is also able to compute the three-dimensional (3D) distribution of photosynthetically active radiation (PAR) inside the canopy and the downward irradiance at the top of a forest canopy and at the forest floor. Furthermore, by inputting the results of the 3D distribution of absorbed PAR into a photosynthesis model (e.g. Farquhar model), it is possible to compute the canopy photosynthesis in given 3D landscapes.

FLiES is based on the Monte Carlo method. Thanks to the recent improvement of the computing power, the users can run the model in their PC in most cases. The model is also designed to run on the parallel computing system. Therefore, the uses can run the model on such computing system, enabling the radiative tranfer simulation in larger or more complex landscapes. The 3D models usually requires a lot of input parameters to characterize the spatial patterns in trees, forest floor etc. To avoid this, I carefully selected the parameters needed as less as possible for the determination of the landscape structure.

As is mentioned above, the simulator is also designed for paralell computing system. So larger-scale computations can be done with a slight change of the source code. This simulator can handle large-scale computations by parallel processing through a Message Passing Interface (MPI), with up to 1,024 CPUs. The source code was developed based on Fortran 77 and some Fortran 90 specifications, and it can be compiled in almost any Fortran 90 or higher compiler, including gfortran, Intel Fortran. **As of July 28th 2014, we do not support the g77 compiler environment any more.**

5

This manual summarizes the FLiES installation process, definitions of input files and input parameters, basic method of use, and provides several examples of analysis results.

3. Installation guide

The source code can be complied and installed according to the procedures below.

1. Input the Fortran command to be used after "FC=" on the first line of <u>Makefile</u> under the directory "src."

2. Execute the following make commands in the directory "src."

```
% make
% make install
% make clean
```

Caution: The executable program "FLiES" reads the atmosphere and vegetation data file (under "Data") and the forest landscape parameter file (crowndata.txt) and then executes the computation. For this reason, "FLiES" needs to be located in a directory higher than "Data."

4. Outline of the model

4.1 Model structure

In FLiES, the sampling of necessary radiation physical quantities is performed by lay-tracing using "model photons," which are the basic units of sampling, over the simulation landscape. In actual simulations, the simulation begins with the incidence of model photons from the top of the atmosphere or the top of the forest. (Fig. 1.)

When model photons are incident on the canopy objects of trees and forest floor vegetation, the direct travel distance of the model photons is determined by using random numbers. After the photons travel these distances, they collide with leaves and branches. The states of the model photons after collision—e.g., reflection, transmission, absorption—are determined for this point in time, and the simulation then continues by modifying the weight and travel direction of the model photons. This series of elementary processes continues until the model photons pass out of the simulation space. (Note that, since the computation in this model is executed under periodic
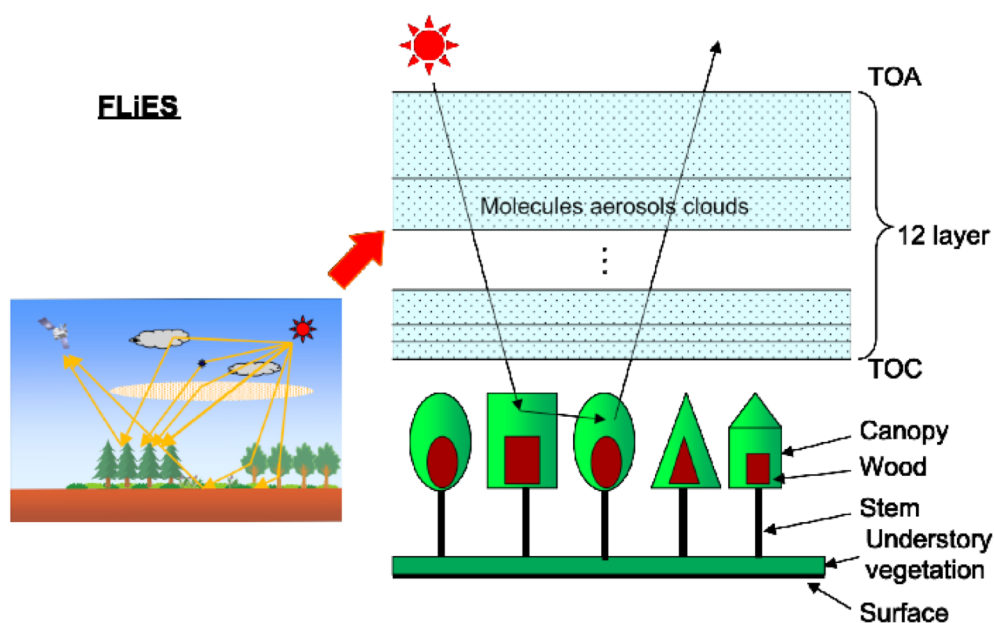

Fig. 1 Diagram of model photon simulations

boundary conditions, model photons that pass out in a horizontal direction are re-radiated into the landscape from reflective surfaces, thereby causing the simulation to continue.) Since FLiES uses random numbers to determine the direct distances and scatter directions of the model photons inside canopy objects, it employs a method (Monte Carlo method) that computes physical quantities by sampling large numbers of model photons.

FLiES conducts sampling of various kinds of radiation physical quantities associated with the atmosphere and vegetation. Some of the main sampled quantities include the illuminance incident at the top of the forest and on the forest floor vegetation, as well as the direct and scattered components of this illuminance, the downward irradiance at the forest top, the bidirectional reflectance coefficient at the forest top, the light absorption in the forest, and the albedo of the forest. Details about the sampling of these physical quantities are summarized in Kobayashi and Iwabuchi (2008).

4.2 Forest representation

The simulation landscape is defined as a space of 30 x 30 m, as shown in Fig. 2. The radiative transfer computation assumes that this space extends without a limit in the horizontal direction. The crowns and trunks of trees are defined as geometric shapes. Crowns are defined as a circular cone, cylinder, or spheroid. Trunks are defined as cylinders. As also shown in Fig. 2, the inside of the tree crown is divided into two regions: an outer region containing only leaves, and an inner region containing only non-photosynthetic elements, such as branches. (The inner region comprises approximately 50% of the tree in both the height and horizontal distances.)

The leaf area density (LAD) of the outer region and the non-photosynthetic bark area density (BAD) of the inner region are each assumed to be constant within each region. In addition, by default the angular distribution of leaf surfaces is considered uniform. The forest floor vegetation is modeled assuming that it is a parallel (to the ground) plane and that leaf surfaces are uniformly distributed. Furthermore, it is assumed that the bottom edge of the forest floor vegetation layer coincides with the soil surface (ground).

The leaf inclination distribution of the forest floor vegetation is also considered, by default, to be uniformly distributed.

Note that the size of the simulation space and the leaf surface angle distribution function can be varied by slightly modifying the program and recompiling it. The details for doing this are described in Chapter 6.
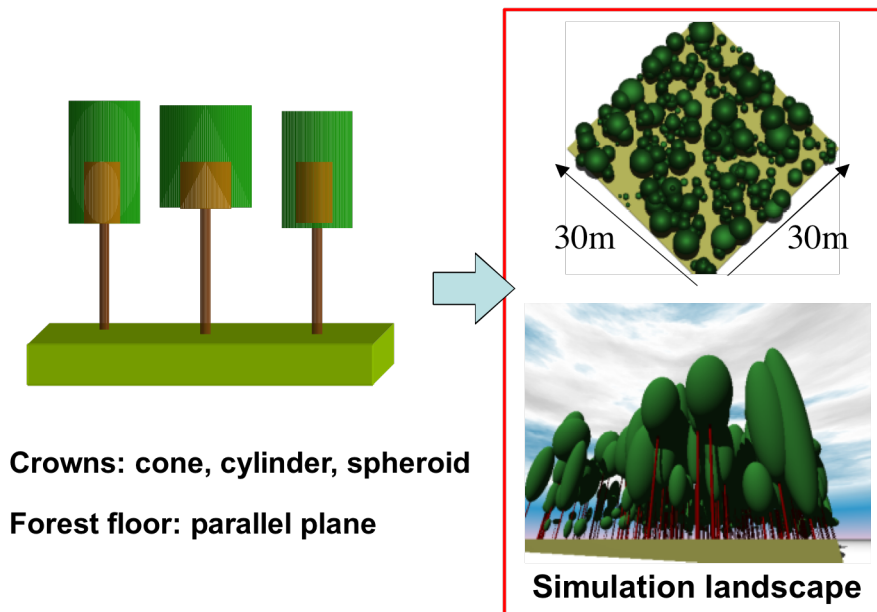

Crowns: cone, cylinder, spheroid

Forest floor: parallel plane

Simulation landscape

Fig. 2 Representations of tree shapes in FLiES (from left: spheroid, cylinder, cone) and the forest landscape for simulation execution

4.3　Creation of forest landscape data

The position and size of tree objects within the defined space need to be described in advance in the tree data file, "crowndata.txt." The standard coordinates of trees are created so that their (x, y) coordinates are each limited by 0 ≤ x ≤ 30 (for the case of a space size of 30 × 30 m). An example of a "crowndata.txt" file is shown below, along with the standard coordinates of each object.

```
000                          X emacs@koba.local
File  Edit  Options  Buffers  Tools  Help

 ◇    ◻    X    ◻    ◻    ↺    ✂    ◻    ◻    ◻    ◻    ◻    ?

   400
   3     0.19    0.34    7.23   14.14    3.09  1
   4     0.19    0.34    0.16    0.16    0.10  1
   3    25.20    3.25    6.84   13.32    2.92  1
   4    25.20    3.25    0.18    0.18    0.10  1
   3    19.00   22.14    6.93   13.52    2.96  1
   4    19.00   22.14    0.17    0.17    0.10  1
   3    12.39   14.97    7.09   13.86    3.03  1
   4    12.39   14.97    0.16    0.16    0.10  1
   3    14.62   16.67    6.52   12.65    2.77  1
   4    14.62   16.67    0.19    0.19    0.10  1
   3     1.80    7.77    6.55   12.73    2.79  1
   4     1.80    7.77    0.19    0.19    0.10  1
   3    20.14    1.25    6.40   12.39    2.72  1
   4    20.14    1.25    0.20    0.20    0.10  1
   3    16.72   20.31    6.52   12.65    2.77  1
   4    16.72   20.31    0.19    0.19    0.10  1
   3     3.89   29.04    5.88   11.32    2.49  1
   4     3.89   29.04    0.22    0.22    0.10  1
   3    11.97   11.73    6.16   11.90    2.61  1
   4    11.97   11.73    0.21    0.21    0.10  1
   3     7.35   15.56    5.96   11.48    2.53  1
   4     7.35   15.56    0.22    0.22    0.10  1
   3    17.14   24.98    5.79   11.12    2.45  1
   4    17.14   24.98    0.23    0.23    0.10  1
   3    14.06    1.46    6.05   11.67    2.57  1
   4    14.06    1.46    0.22    0.22    0.10  1
   3    23.03    1.40    6.20   11.97    2.63  1
   4    23.03    1.40    0.21    0.21    0.10  1
   3    11.31   16.68    5.30   10.08    2.23  1
   4    11.31   16.68    0.25    0.25    0.10  1
   3    10.90   19.41    5.72   10.97    2.42  1
   4    10.90   19.41    0.23    0.23    0.10  1
   3     2.60    1.67    5.41   10.32    2.28  1
   4     2.60    1.67    0.25    0.25    0.10  1
--:--   crowndata.txt      (Text)--L12--Top------------------------
■  (No changes need to be saved)
```

crowndata.txt definition file:

1st line: Total number (positive integer) of geometric objects (crown and trunk) used in the simulation

2nd line and beyond: Parameters of geometric objects

   1st string: Shape of geometric objects

   1= cone

   2= cylinder (for crown)

   3= spheroid

   4= cylinder (for trunk)

   5= semi-spheroid

2nd to 4th string: Reference coordinates of geometric objects (m) (x, y, z). The red points in Figure 3 indicate the reference coordinate of each object. Note that the positions of the reference coordinates depend on the object shapes. The reference coordinates of cone and cylinder shapes are the uppermost side. The reference coordinate of spheriod and semi-spheroid is the center and the bottom.

5th string: Height, h, of geometric object (m)

6th string: Maximum radius, r, of geometric object (m)

7th string: Tree type group number. (1-5). Trees of the same number have the same leaf surface reflectance and transmittance, trunk reflectance, and leaf area density.
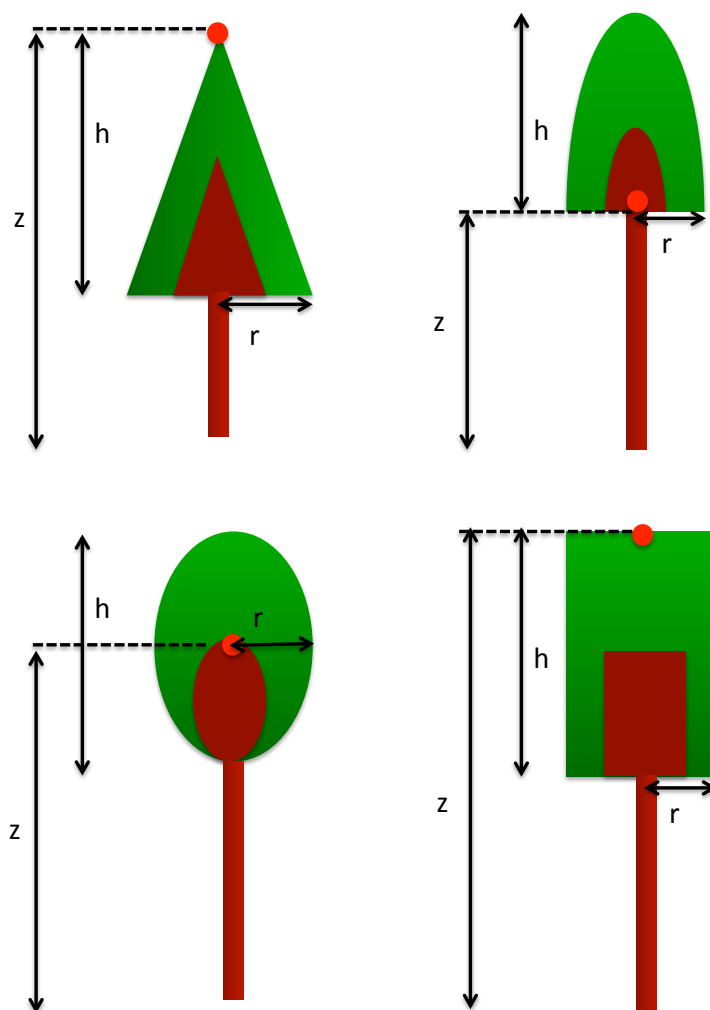
Fig. 3 Standard coordinate values are indicated in red; apex for cones, top center for cylinders, center for ellipsoids and bottom center of semi-ellipsoids for ellipsoids.

5. Examples of simulation

Here I provide an example of a simulation execution. FLiES is designed to execute the executable file "flies," and then to input successively the input parameters from the standard screen. It is possible to prepare the parameters necessary for executing the simulation in advance in an initial parameter file and then perform execution by inputting the parameter file directly into the program. For example, if the initial parameter set is described in the file "init.inp," the computation results can be obtained by executing as follows.

% ./flies < init.txt

5.1 Basic radiation parameters (e.g., BRF, albedo, transmission, absorption)

First, an example computation of a basic light environment is shown. The tree data file (crowndata.txt) and initial parameter file (init_51.txt) necessary for computation are stored in a directory below MANUAL/Samples/. These files are copied to the FLiES execution directory and the simulation is executed.

When the simulation is complete, the computation results are saved in two types of files: "flxsum.txt" and "brfsum.txt." The computation results for the basic radiation parameters are saved in "flxsum.txt"; the results of the BRF computation are saved in "brfsum.txt." Figure 4 shows the computation results for BRF. Since FLiES is an atmosphere-vegetation coupled model, the BRF of the top of the vegetation and the BRF of the top of the atmosphere can be computed simultaneously, enabling the influence of the atmosphere to be easily evaluated. Note that the total number of model photons (np) necessary to calculate the average radiation parameters for a forest landscape is different for each radiation parameter, but computation results of sufficient sampling precision can be obtained by using approximately 3,000 model photons for BRF and 30,000 photons for computing the fraction of absorbed radiation (FAR).

Tree data used for computation (crowndata.txt)
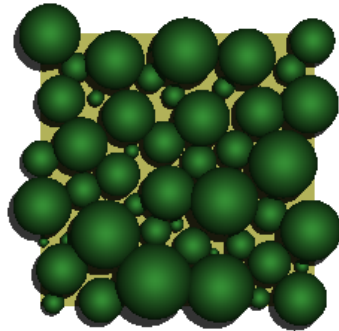
Initial parameter file used for computation (init_51.txt)

```
30000 /* np total photon number */
1 /* amode with atmosphere mode */
45.0 0.0 /* solar zenith anle and elevation (m) */
5 100. 120. 140. 150. 160. /* downward incident radiance (zenith) */
2 0. 180. /* downward incident radiance (zenith) */
2 /* imode integration mode */
2 /* rtype atmospheric profle (Mid latitude summer) */
2 /* atype aerosol type (Continental average) */
0.3 /* AOT at 550nm */
0 /* ctype cloud type (cloud free) */
2 /* surface type (3-D canopy) */
1 /* BRF only */
6 0. 10. 20. 30. 40. 50. /* nth, angt BRF view zenith angles */
2 0. 180.  /* nph angp BRF view azimuth angles */
1 /* nts groupe of tree species */
0.1 0.05 0.15 0.1 0.1 0.05  /* optical parameters of forest component */
0.5  /* LAD leaf area density of canopy */
0.5  /* gLAIforest floot LAI */
0.5  /* BAD bark area index */
0.25 /* sbar Spherical ave. shoot silhouette to totall needle area ratio */
```

Computation results for radiation parameters "flxsum.txt"

```
-- summary of radiative quantity --

 -- simulation mode --
     Atmospheric module:            Yes
        Spectral domain:           PAR
           Surface type:       3 D canopy


 -- Solar zenith angle, Elv(m), atm. transmittance --
     45.00        0.00000      0.81155  <-SZA, elevation, and atmospheric
transmittance


 -- Downward Irradiance at TOC --
 Energy unit (W/m2)
      Total       Beam    Diffuse  Diff.Ratio
   306.11737   198.37787   107.73946     0.35195        <-Irradiance at top of
crown (W/m2)


 Photon Unit (umol/m2/s)
      Total       Beam    Diffuse  Diff.Ratio
   1402.05176   926.07520   475.97339     0.33948        <-Irradiance at top of
crown (PPFD)


 Albedo (A)
     Actual      black      white
            (beam)   (diffuse)
    0.03231    0.03177     0.03329                 <-Albedo at top of crown


 -- fraction of downward flux (T) --
  Forest Floor  Surface floor
    0.27729830     0.20230557                      <-Transmittance (relative
to TOC)


 -- fraction of absorbed radiation (far) --
    total     leaf  non-leaf      floor
 0.7755866 0.6517738 0.0518212 0.0719917           <-Absorbed radiation
(FAR)
 Vertical profile of far
```

```
 H(m)    total    leaf  non-leaf
  21 0.0000000 0.0000000 0.0000000          <-Vertical profile of FAR
  20 0.0004312 0.0004312 0.0000000
  19 0.0058218 0.0058218 0.0000000
  18 0.0131304 0.0131304 0.0000000
  17 0.0347880 0.0347880 0.0000000
  16 0.0543855 0.0543855 0.0000000
  15 0.0692128 0.0689589 0.0002539
  14 0.0867288 0.0857571 0.0009716
  13 0.0968321 0.0920503 0.0047818
  12 0.0922089 0.0841115 0.0080974
  11 0.0805533 0.0698531 0.0107002
  10 0.0587493 0.0475848 0.0111645
   9 0.0403119 0.0317487 0.0085632
   8 0.0258449 0.0197077 0.0061372
   7 0.0165323 0.0126383 0.0038941
   6 0.0103250 0.0070066 0.0033184
   5 0.0058327 0.0030703 0.0027624
   4 0.0036527 0.0014710 0.0021817
   3 0.0035315 0.0010414 0.0024902
   2 0.0027770 0.0000635 0.0027135
   1 0.0027585 0.0000001 0.0027584
```

```
-- spctrl fraction. energy & photon flux(%)         <-Wavelength spectra of
radiation incident on crown
```

wl(um)	Etot	Ebeam	Ediffuse	Ptot	Pbeam	Pdiffuse
0.410	5.705	4.090	8.679	4.268	3.002	4.268
0.430	5.866	4.564	8.264	4.602	3.513	4.605
0.450	7.255	6.192	9.214	5.958	4.988	5.957
0.470	7.386	6.583	8.865	6.333	5.538	6.335
0.490	7.281	6.798	8.170	6.508	5.962	6.507
0.510	7.068	6.918	7.346	6.574	6.312	6.573
0.530	7.246	7.246	7.245	7.007	6.875	7.009
0.550	7.141	7.360	6.739	7.165	7.245	7.164
0.570	6.952	7.410	6.108	7.229	7.561	7.228
0.590	6.707	7.271	5.669	7.217	7.677	7.216
0.610	6.701	7.372	5.466	7.457	8.048	7.459
0.630	6.574	7.360	5.127	7.557	8.300	7.559
0.650	6.171	6.943	4.748	7.317	8.078	7.316

```
0.670    6.327    7.202    4.715    7.733    8.636    7.731
0.690    5.619    6.691    3.647    7.074    8.264    7.073
```

-- Downward radiance at the top of canopy --   <-Radiance at top of crown
idrc theta phi radiance_F stdev_F radiance_Q stdev_Q

```
 1 100.000   0.000    0.05707   0.20664    0.05434    0.19805
 2 120.000   0.000    0.03566   0.11097    0.03327    0.10326
 3 140.000   0.000    0.02809   0.07781    0.02625    0.07197
 4 150.000   0.000    0.02882   0.07580    0.02708    0.07085
 5 160.000   0.000    0.03253   0.08145    0.03078    0.07697
 6 100.000 180.000    0.23732   0.68386    0.23578    0.69447
 7 120.000 180.000    0.32879   0.81696    0.32242    0.82239
 8 140.000 180.000    0.39860   1.01477    0.39054    1.01991
 9 150.000 180.000    0.21701   0.53050    0.21152    0.52969
10 160.000 180.000    0.12965   0.30702    0.12608    0.30634
```

BRF computation results "brfsum.txt"

BRF at top of atmosphere (TOA), top of crown (TOC), and proportional contribution from crown foliage, non-photosynthetic elements, and forest floor

| Theta	Phi	BRF(TOC)	BRF(TOA)	rover	rbark	rfloor
0.00	0.00	0.029634	0.089122	0.815367	0.023232	0.161404
10.00	0.00	0.032178	0.095847	0.822253	0.022793	0.154958
20.00	0.00	0.035212	0.105783	0.831359	0.025773	0.142871
30.00	0.00	0.038952	0.118291	0.845470	0.028333	0.126198
40.00	0.00	0.046257	0.135705	0.834029	0.030818	0.135154
50.00	0.00	0.051004	0.155096	0.850170	0.028411	0.121421
0.00	180.00	0.029634	0.089122	0.815367	0.023232	0.161404
10.00	180.00	0.027380	0.084336	0.808380	0.017802	0.173816
20.00	180.00	0.025931	0.083929	0.793133	0.017645	0.189223
30.00	180.00	0.024693	0.088926	0.797164	0.018147	0.184687
40.00	180.00	0.023959	0.100360	0.815220	0.017672	0.167107
50.00	180.00	0.023989	0.123967	0.838198	0.015823	0.145983
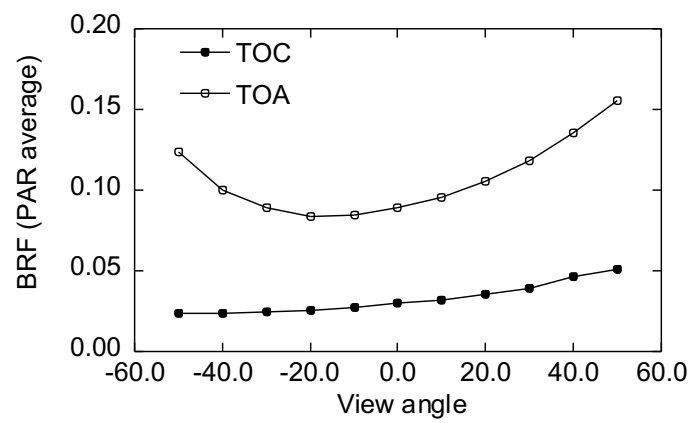
Fig. 4 BRF computation results. TOA: top of atmosphere, TOC: top of canopy.

5.2 Reflectance image and absorbed photosynthetically active radiation (APAR)

FLiES enables computation not only of the average radiation of a forest landscape, but also the spatial distribution of direct line-of-sight radiation (BRF) and the three-dimensional distribution of the absorbed light quantity. These computations may take considerable time, depending on the computer used. (For reference: using a PowerPC 1.33 GHz CPU running Mac OS X under X11, the BRF computation took approximately 15 min. and the APAR computation took approximately 150 min.) Here, an example of calculating a near-infrared reflectance distribution (Samples/init_52ref.txt), and an example of calculating the 3-dimensional distribution of absorbed photosynthetically active radiation (APAR) (Samples/init_52apar.txt) are shown. In these computation examples, the reflectance image was computed in atmosphere-off mode, and the APAR calculation was performed in atmosphere-coupled mode.

The results below were obtained by copying the tree data file, "crowndata.txt," and "init_52ref.txt" and "init_52apar.txt" to the program execution directory and then executing the simulation, as in the case shown in Section 5.1. Normally, at least 10^7 model photons are needed to compute the horizontal distribution of BRF and at least 10^8 photons are needed to compute APAR distribution.

Reflectance (BRF) distribution: TOCIMG.txt

        ASCII file 300 x 300 pixels. In the case of a 30 × 30 m space the pixel size is 0.1 m
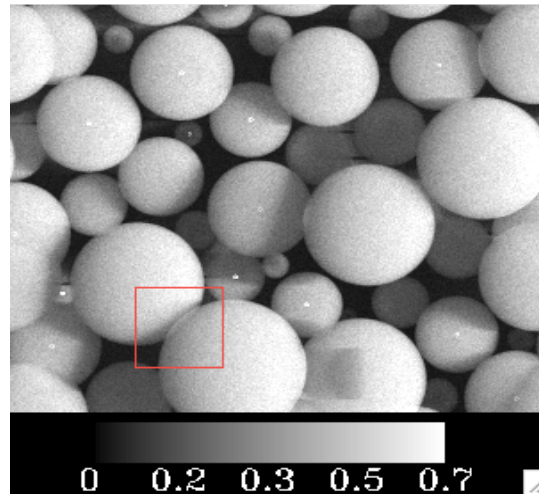
Fig. 5 Computation results for reflectance (BRF) spatial distribution

3D APAR distribution: apar.txt (total APAR), aparb.txt (APAR direct component), apard.txt (APAR scattered component)

>ASCII file horizontal direction 300 x 300 pixels, pixel size 0.1 m. Average values for each 1 m layer in the vertical direction are saved in order as Zmax (m), Zmax-1 (m), ... 0 (m) (Fig. 6). Since the value of Zmax depends on the height of the tree, the total number of layers also depends on the maximum tree height within the landscape. The output results are expressed in units of μ mol/m^3/s.
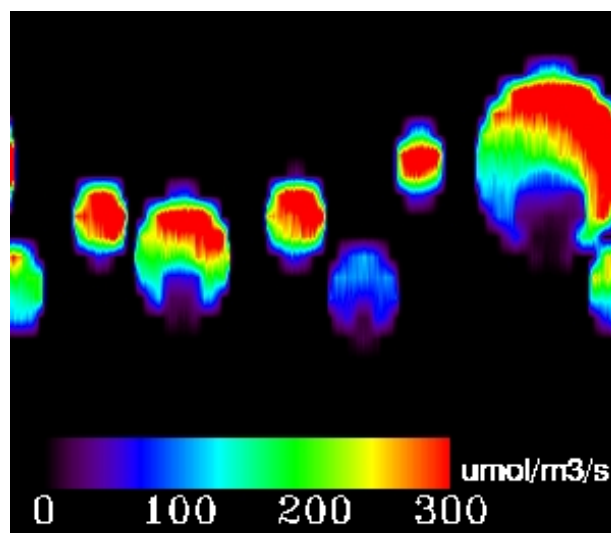

Fig. 6 Vertical slice of the APAR distribution computation results

Forest floor APAR distribution: aparf.txt, aparfd.txt

Figure 7 shows the output results for the total and scattered component of the APAR horizontal distribution due to forest floor vegetation, represented as parallel planar layers. In the case of the forest floor, the APAR output is a cumulative value in the vertical direction, so the output results are expressed in units of μ mol/m^2/s.
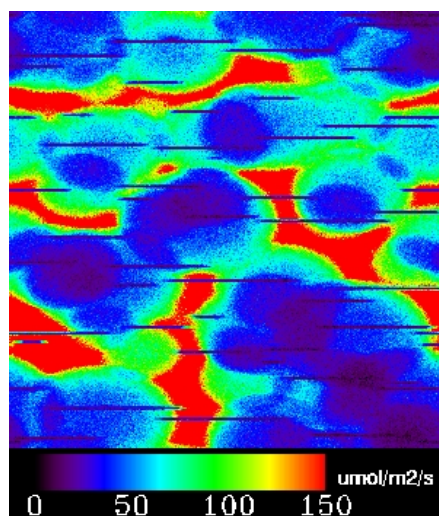

Fig. 7 APAR horizontal distribution at the forest floor

5.3 Calculation of leaf area index (LAI) and crown projection area

FLiES uses leaf area density (LAD) rather than leaf area index (LAI) to compute the leaf volume of tree crowns. When there is no overlap between crowns, LAI can be calculated by multiplying the volume of crown objects by LAD. However, when there is overlap, it is not easy to accurately calculate LAI corresponding to the LAD provided as the input value. In addition, for the same reason, it is not easy to calculate the vertical profile of LAI and crown projection area. Therefore, FLiES provides a computation mode specifically for the crown LAI corresponding to the given LAD. In this case, after

FLiES is executed, "-5" is input when the total number of model photons (np) is input. The initial parameter file for computing the crown LAI is /MANUAL/Samples/init_53.inp. When LAD and BAD on the third and fourth lines of this file are changed and execution is performed, the following are computed: the crown LAI corresponding to the input LAD, its vertical profile in 1 m units, and the crown projection area ratio of the forest landscape.

<u>Initial parameter file for LAI computation: initi_53.inp</u>

```
-5 /* np total photon (-5=LAI calculation mode) */
1 /* nts number of tree species */
0.5 /* u: leaf area density */


0.5 /* branch area density */
0.1 /* sampling grid unit (m) */
```

<u>LAI computation results (standard output)</u>

```
LAI =  1.81402731; Crown cover (%) =  86.5577774
 H(m)  LAI_profile
   21  0.000000
   20  0.001293
   19  0.010799
   18  0.025413
   17  0.069311
   16  0.109497
   15  0.149534
   14  0.197963
   13  0.234416
   12  0.243859
   11  0.229197
   10  0.181468
    9  0.134958
```

```
8  0.093091

7  0.063336

6  0.037258

5  0.019830

4  0.008787

3  0.003608

2  0.000389

1  0.000019
```

5.4 Forest gap probability distribution (fish-eye image)

FLiES is equipped to compute the gap probability of the entire sky as seen from a designated place in the given tree landscape. In this case, after FLiES is executed, "-4" is input when the total number of model photons (np) is input. The initial parameter file used to calculate the gap probability is /MANUAL/Samples/init_54.inp. When LAD and BAD on the third and fourth lines of this file are modified and execution is performed, the gap probability corresponding to the input LAD is computed. The LAI (effective LAI) is also calculated, using the same algorithm employed by LAI-2000 (LiCOR).

Initial parameter file for LAI computation: init_54.inp

```
-4 /* np total photon (-4= Gap Probability mode) */

1 /* nts # of group of tree species */

0.5 /* u leaf area density */

0.5 /* BAD branch area density */

0.25 /*sbar normally 0.25) */

15.0 15.0 /* x and y position */
```

Effective LAI calculated from gap probability distribution (standard output)
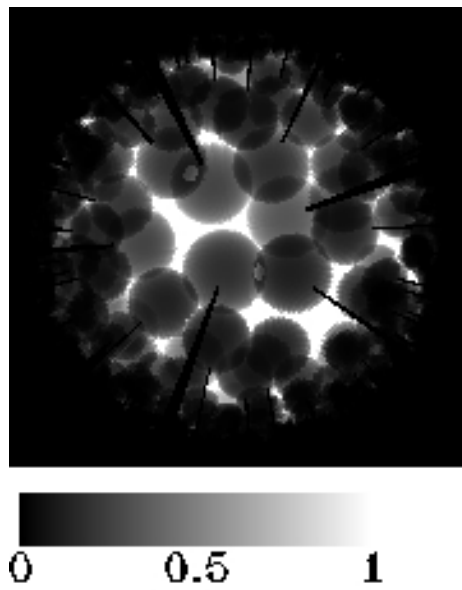
```
Effective LAI =  1.79352045
```

Fig. 8 Forest gap probability at the center (x= 15.0, y = 15.0) of an example landscape (crowndata.txt)

6. Advanced uses

FLiES can be adapted to perform customized computations by slightly modifying and recompiling the program.

6.1 Modifying the simulation space size

Although the default simulation space size is 30 x 30 m, the size can be modified. However, in the current version of FLiES, simulation landscapes can only be square in shape. The procedure for modification is as follows.

  i) Open the source code file "iparam.f" for initial settings in the src directory.

  ii) Change the value of xmax (default = 30) on the 218th line of the source code to the value you wish to use. Units are meters (m). The y (vertical size) value (ymax) must be equal to xmax.

  iii) Recompile the program following the installation procedure described in Ch. 2.

Note that the reflectance image and APAR distribution output size in FLiES is always 300 x 300, regardless of whether the space size is modified. Accordingly, pixel size is 0.1 m/pixel for 30 x 30 m, but 0.2 m/pixel in the case of a landscape size of 60 x 60 m.

6.2 Modifying the leaf surface angle distribution function

By default a uniform distribution is assumed for the leaf surface angle distribution. FLiES also supports two other distribution functions: planophile and erectrophile. Any of these functions can be set independently for the crown, forest floor, and branches. The procedure for modification is outlined below.

  i) Open the source code file "iparam.f" for initial settings in the src directory.

  ii) Change the values (default values = 1) of mc (crown), mb (branch), and mf (forest floor) on lines 230 to 232 of the source code to the index values corresponding to the leaf surface angle distribution function you wish to use. 1 = uniform distribution (uniform) (Default), 2 = planophile, 3 = erectrophile.

iii) Recompile the program following the installation procedure described in Ch. 2.

6.3 Parallel processing (MPI)

FLiES can make use of parallel computer processing that uses up to 1,024 CPUs through MPI (Message Passing Interface). All the functions needed for parallel processing are defined in the file "main.f." However, by default these functions are commented out using "c." In order to execute using MPI, modify and recompile the program according to the procedure below.

 i) Execute the file "mpimain.csh" in the src directory. This script deletes all the commented out symbols "c" from the mpi function lines within "main.f." If this script cannot be used, all the instances of "c" can be deleted manually.

 ii) Recompile the program using makefile (Makefile_MPI). Set options when recompiling as shown below.

   % make –f Makefile_MPI

 iii) After installation is successfully completed, the executable file "pflies" is generated.

For example, in the case of parallel processing using 256 CPUs, execution is as follows.
% mpirun –n 256 ./pflies
% mpijob –n 256 ./pflies
For further details refer to the manuals for the computer used for parallel processing. Note that for versions of the program set up for parallel processing, parameters are read from all the "init.txt" files; that is, parameters are not input in the standard way. To execute computations, an input parameter file needs to be created and saved with the name "init.txt," as explained in Ch. 5.

Appendix 1. Parameter list

Parameter	Definition
np	Total number of model protons used for a simulation
	np = -4: crown gap computation mode
	np = -5: LAI computation mode
amode	Selection of atmosphere radiation process computation mode
	1: Computation from top of atmosphere
	2: Computation from top of crown
dif	Fraction of diffuse radiation to total from atmosphere (0-1)
stype	Selection of ground surface model
	1: Lambert surface
	2: 3D vegetation
The0	Solar zenithal angle (°)
ntha, th	Total number of zenithal angles and angle values for computing solar irradiance at the canopy
	Ex.: For calculation at 0, 15, 30, and 45
	4 0.0 15.0 30.0 45.0
npha, ph	Total number of azimuthal angles and angle values for computing solar irradiance at the canopy
	Ex.: For calculation at 0 and 180
	2 0.0 180.0
imode	Wavelength integral mode
	1: Single-wavelength computation
	2: Photosynthetically active radiation (400-700 nm)
	3: Short wavelengths (300-4000 nm)
wl0	Wavelength value for single-wavelength computation
rtype	Selection of atmosphere profile
	1 Tropical
	2 Mid latitude summer
	3 Mid latitude winter
	4 High latitude summer
	5 High latitude winter
	6 US standard atm.
atype	Selection of atmosphere aerosol type
	1 Continental clean
	2 Continental average
	3 Continental polluted
	4 Urban
	5 Desert
	6 Maritime clean

	7	Maritime polluted
	8	Maritime tropical
	9	Arctic
	10	Antarctic
	11	Smoke
tauref	Optical thickness of aerosol at 550 nm	
ctype	Selection of cloud type	
	0	Cloud-free
	1	Stratus continental
	2	Stratus maritime
	3	Cumulus continental clean
	4	Cumulus continental polluted
	5	Cumulus maritime
	6	Fog
	7	Cirrus 1 (-25 degC)
	8	Cirrus 2 (-50 degC)
	9	Cirrus 3 (-50degC + small particle)

For detail, see: Hess et al (1998), "Optical properties of aerosols and clouds: The software package OPAC." Bulletin of the American Meteorological Society **79**(5): 831-844.

ctauref	Optical thickness of cloud at 550 nm
ctop, cbot	Max. and min. altitude of cloud layer (m)
cmode	Selection of physical quantity computed inside 3D canopy
	1: BRF
	2: Zenithal reflectance image computation
	3: Photosynthetically active radiation absorption 3D distribution computation
nth, angt	Total number of zenithal angles and angle values for BRF computation
	Ex.: For calculation at 0, 15, 30, and 45
	4 0.0 15.0 30.0 45.0
nph, angp	Total number of azimuthal angles and angle values for BRF computation
	Ex.: For calculation at 0 and 180
	2 0.0 180.0
nts	Number of tree type groups (up to 5 groups can be defined, which have different crown foliage and trunk optical characteristics)
lr, lt, ult, ult, str, sor	Respectively, reflectance and transmittance of crown foliage, reflectance and transmittance of forest floor vegetation, trunk reflectance, and soil reflectance
	Ex.: In the case of 2 tree type groups: for first group (leaf surface reflectance, leaf surface transmittance, trunk reflectance) = (0.45, 0.51. 0.2); for second group = (0.34, 0.42. 0.25); leaf surface reflectance and transmittance at forest floor = (0.44, 0.43); soil reflectance = 0.10:
	0.45 0.34 0.51 0.42 0.44 0.43 0.2 0.25 0.10
u	Leaf area density. Input for each group in the case of multiple tree type groups.
	Ex.: Leaf area density of 0.25 and 0.1, respectively, for tree type group 1 and 2:

	0.25 0.1
gLAI	LAI of forest floor vegetation
BAD	Branch area index. Input for each group in the case of multiple tree type groups.
	Ex.: Leaf area density of 0.25 and 0.1, respectively, for tree type group 1 and 2:
	0.25 0.1
sbar	The ratio of the spherical average shoot silhouette area to the total needle area:
	Set to 0.25 in the case when the influence of shoot clumping is not considered.
	Input for each group in the case of multiple tree type groups.
	Ex.: sbar is 0.25 and 0.15, respectively, for tree type group 1 and 2:
	0.25 0.15

Appendix 2. Source codes

main.f

  This is a main program. The simulation start with this code and the following subroutines and functions are called from this main program hierarchially.

pmain.f

  This is a main program for parallel computation mode (MPI). This code is automatically generated from main.f with a shell script "mpimain.csh".

Subroutines and functions:

alpha.f

  subroutine alpha(ti, pi, to, po, cosa, a)

  This subroutine returns angle (a) and its cosine (cosa) in between incoming vector (ti, pi) and outgoing vector (to, po).

arthshift.f

  subroutine arthshift(x, ux, path, xmax)

  A function to shift photon horizontally. When the photon exits from the defined area. The photon re-enters from the opposite side. This function is only used in the atmospheric radiative transfer module.

artphsfintp.f

  subroutine artphsfintp(rawang, rawphs, wrkc1, wrkc2, wrkc3, wrkang, wrkphs, nraw, nwrk)

  Interpolation of phase function by natural cubic spline. This function is only used in the atmospheric radiative transfer module.

artrroulette.f

  subroutine artrroulette(w, wrr)

  Russian roulette: survive or killed? If killed, the photon tracing is stopped and if survive photon gets weight, not to make a statistical bias owing to suddnly killed photons. This function is only used in the atmospheric radiative transfer module.

artsanglut.f

  subroutine artsanglut(wrkang, wrkphs, wrkcum, wrkpdf, nwrk, nlut, wrksca)

Make LUTs of scattering angle for uniform distributions and others. This function is only used in the atmospheric radiative transfer module. This function is only used in the atmospheric radiative transfer module.

artscat.f

subroutine artscat(ux, uy, uz, sinq, cosq, sinf, cosf)

Scattering (renewal of the direction of photons). This function is only used in the atmospheric radiative transfer module.

canort.f

subroutine canort(x, y, ux, uy, uz, w, wq, cmode, nscat, lr, lt, ulr, ult, str, sor, ichi, ikd)

This is the main subroutine to do a photon tracing in plant canopies. Once the photon enter the canopy space, this function is called. The function search the first intersection geometry: canopy geometric volume, understory plane parallel layer or vertical stem?

clai.f

subroutine clai(lai, plai, spn, cv)

This function calculates the canopy total LAI and its vertical profile. Canopy coverage is also estimated.

cones.f

subroutine cones(s, x, y, z, ux, uy, uz, tobj, face, io)

This subroutine calculates the distance between the (x,y,x) and cone external boudary.

cyls.f

subroutine cyls(s, x, y, z, ux, uy, uz, tobj, face, io)

This routine calculates the distance between the (x,y,x) and cylinder external boudary

elpss.f

subroutine elpss(s ,x ,y ,z ,ux ,uy ,uz ,tobj, io)

This routine calculates the distance between the (x,y,x) and rotational elliptical sphere external boudary

frnd.f

real*8 function frnd()

Random number generator based on the Tausworthe sequence method. Original code and algorithm: Shinomoto (1992), Statistical mechanics: Parity physics cource., Maruzen press, P27 (in Japanese)

frndi.f

integer function frndi(ix)

Initialized function of rondom number sequence. Random number function based on the Tausworthe sequence method. Original code and algorithm: Shinomoto (1992), Statistical mechanics: Parity physics cource., Maruzen press, P27. (in Japanese)

getext.f

subroutine getext(nmix,cflg, ext, Qext, Qext_ref, d, cbnz, ctnz, taur, ctaur, rat)

This subroutine calculates the spectral extinction coefficient of the atmopshere (including molecular, aerosol and clouds). The extinction coefficents are computed every layer of the atmosphere.

real function fzpf(z,d)

Aerosol vertical distribution function. Assumed exponential function with regard to the height.

getphs.f

subroutine getatm(wl0, nkd, wkd, ext, rfname)

This fucntion is to obtain the atmospheric parameters related to extinction coefficients and phase functions.

subroutine getphs(wl0, span, nmix, re, Qext_ref, Qext, omg, G, Qabs, ang, phs, fname, imode)

This fucntion is to obtain the atmospheric parameters related to phase functions.

getrancircle.f

subroutine getrancircle(r2min, r2, r, sinf, cosf)

Get a random point withing a unit circle. This is an efficient computing technique to obtain random numbers.

helpss.f

subroutine helpss(s, x, y, z, ux, uy, uz, tobj, face, io)

This routine calculates the distance between the (x,y,x) and rotational half of elliptical sphere external boudary.

i_rvctrssrch1.f

    integer function i_rvctrssrch1(xgrd, x, ix, ixmin, ixmax)

    Find a grid number i, where

     xgrd(i) <= x   < xgrd(i+1) when xgrd is increasing vector, or

     xgrd(i)   > x >= xgrd(i+1) when xgrd is decreasing vector,

    by sequential search method with a given initial estimate, ix.

    Output i will be in the range [ixmin, ixmax]. x should follow

     xgrd(ixmin) <= x   < xgrd(ixmax+1), or

     xgrd(ixmin)   > x >= xgrd(ixmax+1).

    This function is only used in the atmospheric radiative transfer module.

idivspace.f

    subroutine idivspace

    This function is prepared for the efficient computing. It creates big cubics (voxels) with a size of a 50-pixel-equivalent length (50   pixel by 50 pixel by 50 pixel). Then tree geometry information (geometry ID which will be given in the program automatically) within a big cubic is collected and stored in the array named "divs(i, n)". The first subscript i indicates the number of cubic. e.g. the cubic with i=1 is (xmin=0,ymin=0,zmin=0) to (xmax=5,ymax=5,zmax=5), and i=2 (xmin=5,ymin=0,zmin=0) to (xmax=10, ymax=5, zmax=5). The second subscriupt n is a number of geometry.All tree geometry is sorted into this big voxel in order to reduce the computation time in ray tracing. If a photon lies in the i-th cubic, we only need to seach the potential intersection point from geometries listed in the array of "divs"

ifunc_vctrbinsrch.f

    integer function ifunc_vctrbinsrch(grd, n, irvrs, dat)

    Search grid number by binary search method.

    grd(i) <= dat < grd(i+1) or grd(i) > dat >= grd(i+1).

    Note :

     irvrs = 0   : grd(i) is increasing with i

     1   : grd(i) is decreasing with i

    This function is only used in the atmospheric radiative transfer module.

igtbl.f

subroutine igtbl

This subroutine pre-computes the so called "G-function", a projected leaf area in a unit leaf area and create a look-up-table (LUT). This LUT is used while photon tracing. Calculation of G-function requires doule integral so the use of pre-computed LUT significantly reduces the computation time in a photon tracing.

real function fpsi(th, thl)

This function compute the factor called "psi" defined in Shultis and Myneni (1988), J. Quant. Spectrosc. Radiat. Transfer, 39, 2, 115-129 (See Appendix). This is basiacally a absolute value of the inner product between two unit vectors.

real function fgl(thl, i)

Leaf angle distribution function (gl x sin(th)) using Bunnik (1978) definition. Be sure that "sine" is already multiplied in this function.

imath.f

subroutine imath

Some frequently used mathematical functions (sin, cos, acos, exp, delta-function) are pre-computed and the creates look-up-table.

iparam.f

subroutine iparam (

    &   **flg,cflg,rtype,atype,nkd,**

    &   **wl0,wls,nwl,span,**

    &   **taur,ctaur,d,**

    &   **th0,ph0,tm,hfov,**

    &   **tflx,bflx,dflx,obflx,odflx,**

    &   **tpfd,bpfd,dpfd,obpfd,odpfd,**

    &   **rflx,rbflx,rdflx,scmpf,scmpp,**

    &   **cbnz,ctnz,RF,RQ,**

    &   **fpc,fpf,fpv,ext,Nid**

    &)

Variable initialization function. All variables used in the program are first initialized by this subroutine.

ipf.f

subroutine ipf

This function pre-computes the kernel of the leaf phase function in bi-lamertian leaf surface case. The equation is defined in Eq 43 in Shultis and Myneni (1988), J. Quant. Spectrosc. Radiat. Transfer, 39, 2, 115-129. Phi+ and Phi- are the kernel computed in this subroutine. This fucntion is double-integral so pre-computation and LUT significantly reduces the computation time. There are two options in line 33 (op=1 or op=2). op=1 will compute the kernal and op=2 reads the pre-computed LUT from the text file in the directory "Data". Three kernels corresponding to the leaf angle cases are pre-computed (spherical, planophile and erectrophile). In default, op=2 is chosen.

math.f

real function fsin(x)

real function fcos(x)

real function facos(x)

real function fexp(x)

These functions are used for sine, cosine, arccosine, and exponential calculation from pre-computed LUT. The use of these functions reduce the computation time significantly.

math_test.f

This program is used for the validation of the math functions in math.f & imath.f. This source code is not used in Make for FLiES compilation.

mc1descape.f

subroutine mc1descape(x, y, z, uxr, uyr, uzr, iz, ichi, ikd, xr, yr, tau)

This subroutine traces a photon trajectory from the current position to a given point in plane-parallel vertically-inhomogeneous atmosphere and computes the optical thickness (tau) along the path. In default in current program, "a given point" is set at the bottom of the

mc1doptics.f

subroutine mc1doptics(ext, omg, phs, ang, knmix, nmix,iwl)

Make LUTs for phase function & scattering angle of atmopsheric medium (molecular, aerosol and clouds).

mc1dtrace.f

subroutine mc1dtrace(w, wq,x, y, z, ux, uy, uz, ftau, chi, iz, ikd, nscat, ichi,iwl)

This subroutine traces a photon trajectory in plane-parallel vertically-inhomogeneous atmosphere.

mccnp.f

subroutine mccnp(w, wq, x, y, z, ux, uy, uz, cmode, nscat, tobj, inobj, io, face, lr, lt, str, ichi, ikd)

This subroutine calculates photon trajectry wihtin a canopy geometry by using Monte Calro technique.

mccnp_lrch.f

subroutine mccnp(w, wq, x, y, z, ux, uy, uz, cmode, nscat, tobj, inobj, io, face, lr, lt, str, ichi, ikd)

This function is specifically prepared for larch forest. In default, this program is not included in make (Makefile).

mcflr.f

subroutine mcflr(w, wq, x, y, z, ux, uy, uz, nscat, cmode, ulr, ult, sor, ichi, ikd)

This subroutine calculates photon trajectry in the plane parallel forest floor.

mcstm.f

subroutine mcstm (w, wq, x, y, z, ux, uy, uz, nscat, cmode, tobj, face, str, ichi, ikd)

This subroutine calculates photon reflection on the stem surface.

mpisum.f

subroutine mpisum(tflx, bflx, dflx, tpfd, bpfd, dpfd, scmpf, scmpp, rflx, rbflx, rdflx)

This subrotine summarizes the values. This is only used for mpi.

nrmlzphsf.f

subroutine nrmlzphsf(nang, ang, pf, sumnorm)

Normalize phase function to sumnorm.

nrmlzuvctr.f

subroutine nrmlzuvctr(ux, uy, uz)

Renormalize a unit vector by using Newton method. A given unit vector must satisfy that

$$ux*ux + uy*uy + uz*uz$$

is nearly equal to 1.

pf.f

subroutine fpf(gm, thi, tho, phr, lr, lt, cb)

This function calculates the phase function of plant canopy from LUT.

phsfbiasym2.f

subroutine phsfbiasym2(ang, phs, nang, angs, iangs, g0fwd, g0bwd, g1, g1fwd, g1bwd, g2, g2fwd, g2bwd)

Get asymmetry factors of scattering phase function for each

   forward and backward regions.

Note: ang() should be an increasing function.

phsftrunc5.f

subroutine phsftrunc5(cosa, cump, pdf, ang, nang, g1, g2, fd, ftf, ftb, g1t, g2t, angtf, angtb, iangtf, iangtb)

Dual-end truncation approximation (DTA):

Truncation of forward peak of phase function with a correction

for conservativation of 1st and 2nd moments of cosQ.

Note:

1, cump() is in the reverse order (1 to 0)

2, The renormalized phase function is

P = phsf(ang) / (1 - ftf - ftb) if angtf < ang < angtb

0   else

planes.f

subroutine planes(s, x, y, z, ux, uy, uz, x1, y1, z1, face, intv)

This routine calculates the distance (an intersection point) between the (x,y,z) and cube (voxel).

prepatm.f

subroutine prepatm(imode, wq, RQ, span, spcq, np, npproc,

      &   wl0, wls, iwl, npl, nkd, wkd, ext, rfname, fname,

      &   nmix, cflg, re, G, Qabs, Qext, Qext_ref, d,

    & cbnz, ctnz, taur, ctaur)

This subroutine prepares the atmopsheric variables for the condition defined in a simulation.

r_acos.f

  real function r_acos(x)

  Get single-precision acos(x) using LUT, where x should be in the range from -1 to 1.

rparam.f

  subroutine rparam(

    & np,amode,cmode,imode,nwl,npl,nmix,

    & atype,rtype,stype,cbnz,ctnz,cflg,

    & dif,th0,ph0,phi,th,ph,

    & sinf0,cosf0,cosq0,sinq0,

    & wq,span,wl0,wls,RF,RQ,

    & d,spcf,spcq,taur,

    & ctaur,rfname,fname,alb,

    & lr,lt,ulr,ult,str,sor,

    & npproc, Nprod, num)

  Reading input parameters.

scatvec.f

  subroutine scatvec(lr, lt, uxi, uyi, uzi, uxo, uyo, uzo, m)

  This functiom determins photon scattering direction when it hits in the plant canopy media. The multiple rejection method is used as described in Frontier Technical Report No7 p90-91.

srfref.f

  subroutine srfref(ux, uy, uz, sor, nscat, mode)

  Scattering in the surface boudary (soil surface). In default, lambertian scattering is assumed. The users can modify this subroutine so as to consider the bi-directional reflectance.

test_intersec.f

  This program is used to check the calculaiton of the intersection points by using subroutines: "planes", "elpss", "cyls", "cones", "helpss".

transform.f

subroutine trans(uxl,uyl,uzl,a,b,uxo,uyo,uzo)

Coordinate transformation in a 3-D space.

twovecang.f

subroutine twovecang(ux1, uy1, uz1, ux2, uy2, uz2, cosa, a)

Computes angle and cosine between two unit vectors.

vegfeye.f

subroutine vegfeye(w, tgx, tgy, ichi, ikd)

This subroutine calculate the fish-eye (upper hemispherical) image from the canopy bottom at the location (tgx, tgy).

vegrad.f

subroutine vegrad(w, x, y, z, ux, uy, uz, lr, lt, cb, a, fd, ichi, ikd)

This function traces the photon trajectory to a given direction toward the top of the canopy. This is the function of local estimation in plant canopy.

vegrroulette.f

subroutine vegrroulette(w,e)

Russian roulette in plant canopy. The algorithm is the same as "**artrroulette.f**": survive or killed? If killed, the photon tracing is stopped and if survive photon gets weight, not to make a statistical bias owing to suddnly killed photons.

vegtrace.f

subroutine vegtrace(tau, x, y, z, uxr, uyr, uzr, sflag)

This subroutine simulates the optical thickness in the canopy along the given photon direction.

vegtrace_lrch.f

subroutine vegtrace(tau, x, y, z, uxr, uyr, uzr, sflag)

Same as "**vegtrace.f**" ,but for larch forest. This program is automatically generated by shell script. In default, this source code is not included in make.

wdata.f

subroutine wdata(np, nwl, stype, imode, cmode, amode,

     &   span, RF, RQ, cosq0, wl0, wls, tflx, bflx, dflx,

     &   tpfd, bpfd, dpfd, scmpf, scmpp,rflx, rbflx, rdflx)

This subroutine is for data writing module to utput files.

Reference publications

Kobayashi, H. and H. Iwabuchi (2008), A coupled 1-D atmosphere and 3-D canopy radiative transfer model for canopy reflectance, light environment, and photosynthesis simulation in a heterogeneous landscape, Remote Sensing of Environment, 112, 173-185.

Iwabuchi H. and H. Kobayashi, (2006), Modeling of radiative transfer in cloudy atmospheres and plant canopies using Monte Carlo methods, JAMSTEC Frontier Technical Report Vol 7., p.236.

Acknowledgment