



Technical documentation

Developers:

Heike Lischke, Thomas Löffler, Niklaus Zimmermann, David Böhm

Dynamic Macroecology, Swiss Federal Institute of Forest, Snow and Landscape Research
WSL

This is a documentation of the core TreeMig model code.

Table of Contents

1	<i>Technical and code documentation</i>	2
1.1	Calling sequence of routines	2
1.1.1	Preparation of simulation	2
1.1.2	Creating dispersal kernel.....	3
1.1.3	Time loop.....	4
1.1.4	Spatial loop.....	4
1.1.5	FFT seed dispersal	5
1.1.6	Local dynamics in one time step	6
1.1.7	Local forest dynamics.....	7
1.2	More details	8
1.2.1	Netcdf handling	8
1.2.2	Sequential and parallel implementation	8
1.2.2.1	Parallelisation	8
1.2.3	Preprocessing.....	8
1.2.4	Initialization.....	8
1.2.5	Spin up.....	9
1.2.6	How to apply TreeMig.....	9
1.2.7	Technical information	9

1 Calling sequence of routines

The black bordered cells in the graphs indicate links to other graphs. For more details, see the code documentation of the FORTRAN files.

1.1 Preparation of simulation

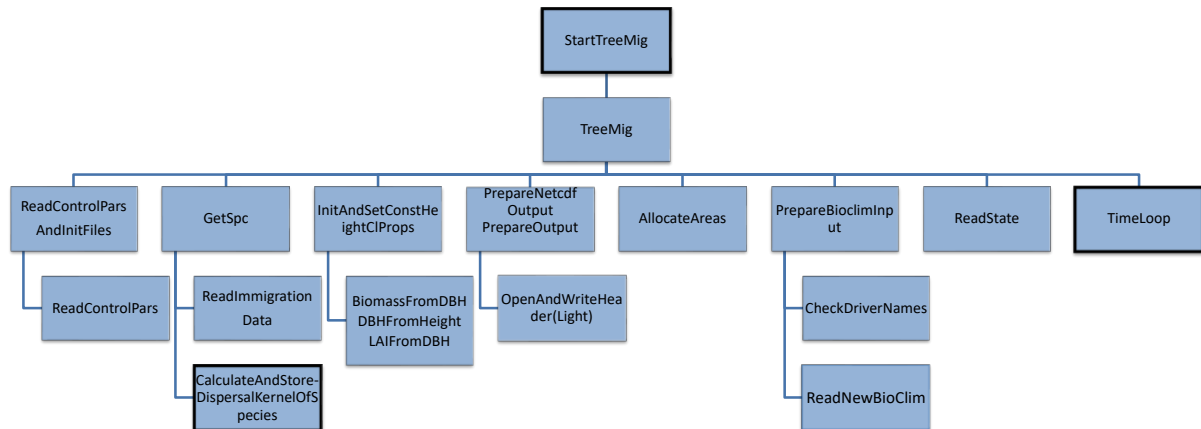


Figure 1: Preparation and start of the simulation

1. The program is started in StartTreeMig, which sets the flags and setting for the parallel computing and calls the TreeMig model.
2. The control settings are read in from a control file (ReadControlPars) whose name and path is given in the command line when calling TreeMig and read in in ReadControlParsAndInitFiles.
3. The output files are prepared (ReadControlParsAndInitFiles, PrepareOutput, PrepareNetcdfOutput), i.e. opened and given a header.
4. Species parameters (GetSpc) and immigration data (ReadImmigrationData) are read in, further species properties are derived from them and stored.
5. Also the dispersal kernels and their Fourier transformations (CalculateAndStoreDispersalKernelOfSpecies) are created (Figure 2).
6. State variable and help grids are allocated and initialized (AllocateAreas).
7. The input of the bioclimate driver data is prepared (PrepareBioclimInput)
8. Height and light dependent species properties are derived from the parameters or initialized and stored (InitAndSetConstHeightCIProps).
9. “Inoculation”: The simulation is run for a given inoculation time with seeds available everywhere either in the entire simulation domain or a subarea (1st call of TimeLoop). This is necessary to have seeds and even trees available, if the simulation is started from bare ground.
10. Optional: Initializing the state variables with a state snapshot (statefile), usually stemming from a previous simulation (ReadState)
11. Migration simulation: This is the actual simulation run with dispersal (2nd call of TimeLoop)
12. The files are closed.

1.2 Creating dispersal kernel

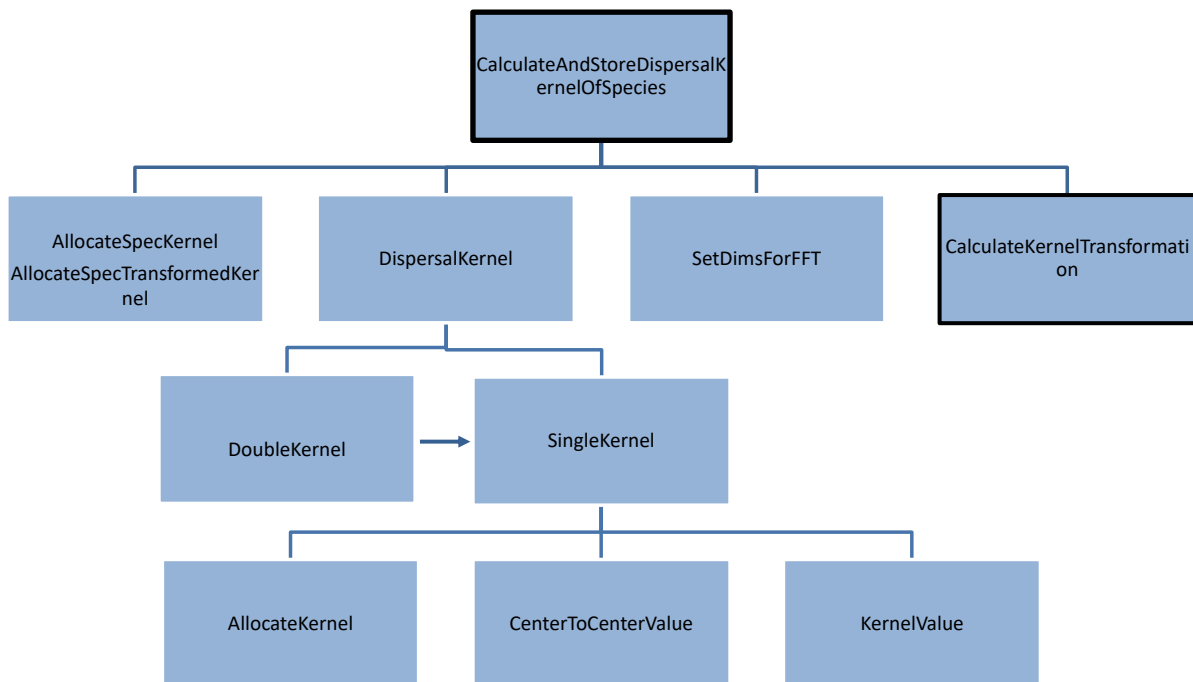


Figure 2: Creating the dispersal kernel

1. The memory for the kernel of the species as well as for the FFT-transformed kernel is allocated
2. The kernel is calculated
 - a. Either as double kernel consisting of a weighed sum of two single kernels with different parameters
 - b. Or as a single kernel, that is calculated (KernelValue) on a finer resolution and then summed up to get the actual resolution, and finally cut and normalized.
3. For the later FFT the dimensions of the small extended and large extended areas are set in **SetDimsForFFT**, where the small extended area has dimensions of the next power of 2, and the large one of the overnext power of 2.
4. The kernel is FFT transformed

1.3 Time loop

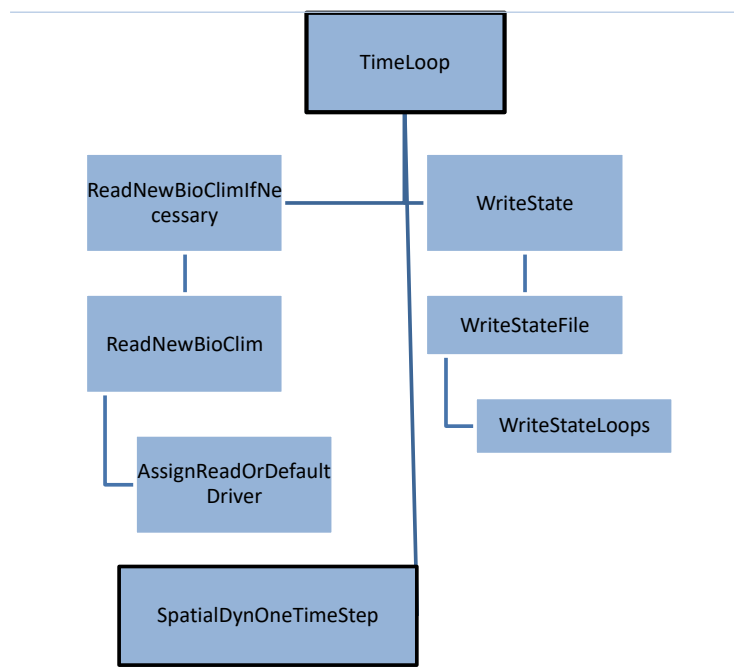


Figure 3: Time-Loop

In the time loop over the simulation time (Figure 3), within each time step

1. The new bioclimate for this time is read in (ReadNewBioclim). Non read-in cells are marked by a Null value and later in the simulation masked out. The read-in values are assigned to the correct places in the structure newBioClim. If no value is given for a driver other than DD, WIT and DrStr, then default values are assigned (AssignReadOrDefaultDriver).
2. The dynamics of the state variables in the entire grid are calculated (SpatialDynOneTimeStep, see below).
3. The current state is written out, if the simulation is at a time point where this is wished (WriteState).

1.4 Spatial loop

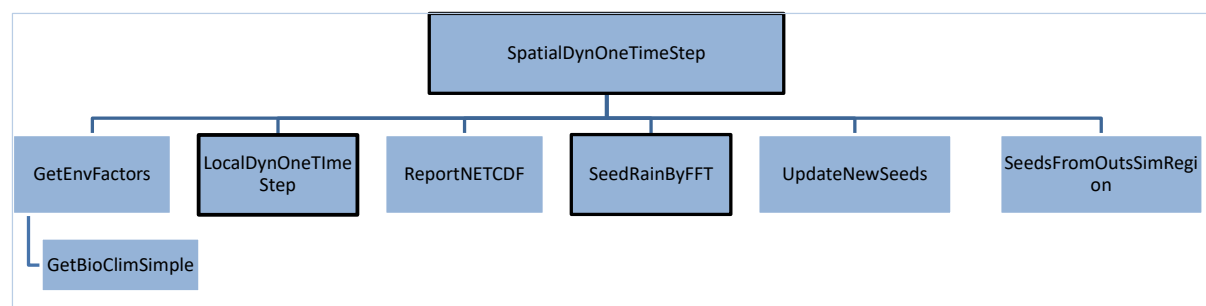


Figure 4: Spatial dynamics in one time step

In the spatial loop (SpatialDynOneTimeStep) the dynamics of the state variables in the entire grid for this time step are calculated.

1. There, in a loop over the grid cells
 - a. the environmental factors (DD, WiT, DrStr, brwpr, disturb, germDrought) are extracted from newBioClim (GetEnvFactors)
 - b. and the local dynamics are calculated (LocalDynOneTimeStep)
2. Optionally, the calculation of dispersal by fast Fourier transform is calculated (SeedRainByFFT, Figure 5, cf. **Error! Reference source not found.**)
3. The seeds in the seed bank are updated
4. Optionally, the Netcdf output file is updated with the results for this time

1.5 FFT seed dispersal

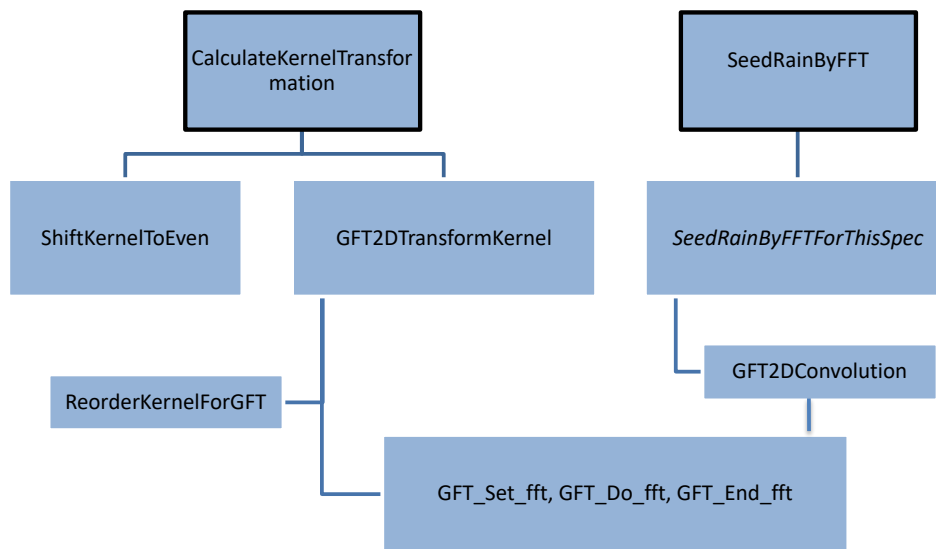


Figure 5: FFT seed dispersal

The FFT seed dispersal consists of

1. Transforming the kernel by
 - a. shifting it
 - b. Reordering it
 - c. doing the GFT transformation (GFT_Set, -Do, -End)
2. The convolution of the kernel with the seed production is done in each time step in **GFTConvolution2D**.
 - a. the seed production area is buffered at the top and right by stripes of the kernel-radius and extended to dimensions of the next power of two (intermediate area), so that the original simulation area is in the 0,0, corner.
 - b. Then this intermediate area is put to the corner of the large extended area that has dimensions double to those of the intermediate area.
 - c. This large area is then FF transformed, however indicating the dimensions of the intermediate area.
 - d. The resulting FF transformed seed production area is multiplied with the FF

transformed kernel

- e. and then the product is back-transformed. Very low values are interpreted as noise and are set to zero.
- f. Finally, the original area is cut out of the result that is on the size of the intermediate area.

1.6 Local dynamics in one time step

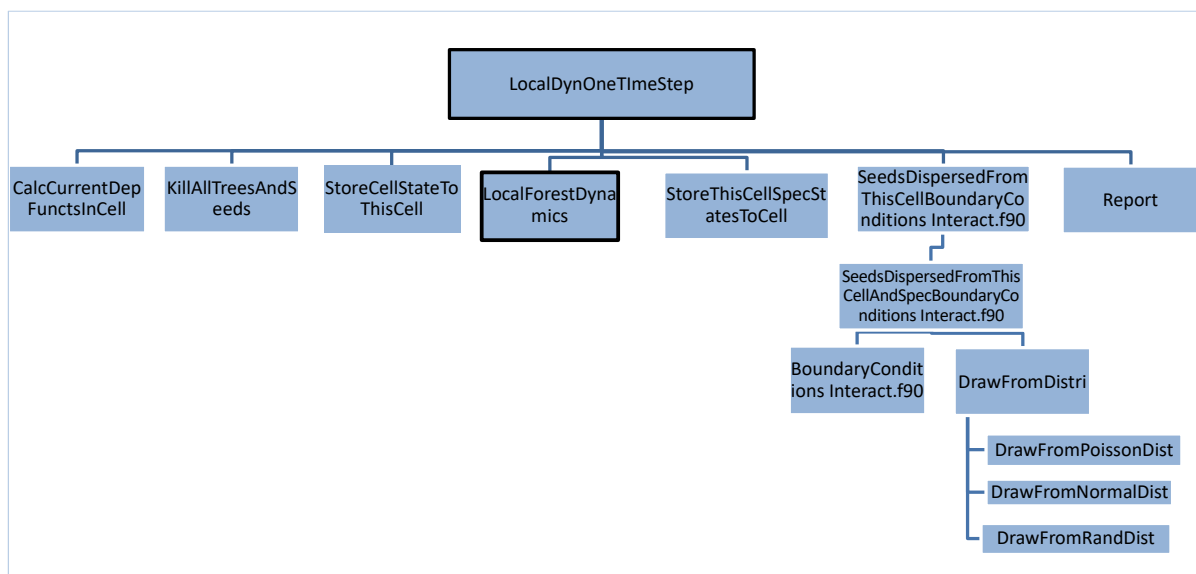


Figure 6: Local dynamics in one time step

1. the environmental dependencies are determined based on environmental factors (dd, wiT, drstr, brwpr)
2. the current state is fetched and stored to thiscell (remainder of the parallel implementation)
3. the local forest dynamics is calculated
4. and stored back
5. seeds are dispersed (if brute force and dispersal is on)
 - a. all sink cells around the source cell within the range of the species's dispersal kernel are reached and
 - b. the probability of seeds landing there is determined from the dispersal kernel
 - c. According to the boundary conditions (absorbing, cyclic,...) it is determined whether the sink cell is the correct one
 - d. The number of seeds landing in the sink cell is calculated either as a binomial distribution with the produced number of seeds (set to the next lower integer number) and the probability
 - e. or as a fraction of the produced seeds
 - f. This number of seeds landing in the sink cell is added to all seeds landing there.
6. the current results for this cell are written out (reported)

1.7 Local forest dynamics

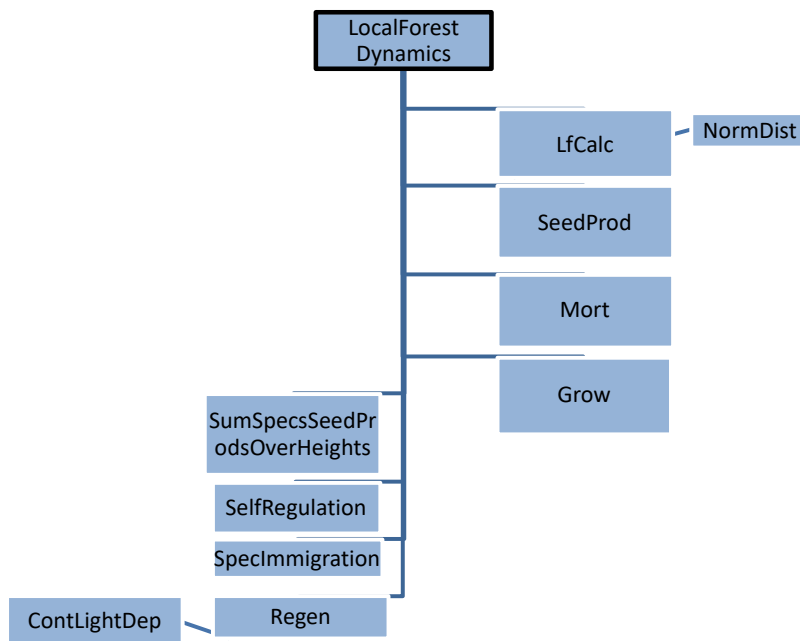


Figure 7: Local forest dynamics

1. current cell is initialized for this time step
 - a. numbers of trees in height classes and ingrowth
 - b. light and (total) leaf area structures
2. disturbance mortality is calculated
3. then for each height class from top to bottom
4. the light distribution is calculated (LfCalc)
5. for each species
 - a. the produced seeds are determined (SeedProd)
 - b. in each light class
 - i. mortality and growth are determined
 - ii. within these subroutines the current numbers are updated
6. the total seed production per species is calculated
7. the self regulation of the species on the level of seeds ist determined
8. immigration, if there is any, is evaluated
9. the regeneration, i.e. germination and competition for space is determined

2 More details

2.1 Netcdf handling

Optionally the output of TreeMig can be a single, multidimensional Netcdf-file.

In `PrepareNetcdfOutput` the output Netcdf file is prepared. It is created, the global attributes of the file are defined. Time, x,y. and species axes are defined with their attributes. The different output variables are defined, allocated and initialized. In `ReportNETCDF`, all output data of one year are added to the netcdf output file. The values have been stored before to the `_NC` variables (e.g. `Biomass_NC`) in `Report`, which has been called for each cell in the same time step before

2.2 Sequential and parallel implementation

TreeMig is in its standard use sequential, in particular the version used in the TM-Package. There was a working parallel version several years ago, but the parallel version has not been tested lately, in particular not in combination with the seed dispersal by FFT. So, it cannot be used as such. However, to simplify experienced users the parallel implementation, we have left the corresponding parts and files in the code.

2.2.1 Parallelisation

The main idea of the parallelization is that the local and spatial simulation is run in parallel on different (overlapping) stripes of the simulation area on different “slave” processors. A “master” processor a) reads in the driver data and b) collects the seeds passing over the stripe boundaries and distributes both to the correct stripes and their processors. The parallelization is based on MPI (message passing interface). The relevant parts and settings are activated by a compiler preprocessor, with the compiler flags “-DPARALLEL” and “-DWITHMPI”. The parallel version is controlled by compiler directives and preprocessor commands, i.e. te lines starting with # and the blocks between them. The files in which they appear must end with F90 instead f90. The preprocessor commands and blocks are currently outcommented. The additional files only needed in the parallel version are located in the folder “CurrentlyNotNeeded/MPI”.

2.3 Preprocessing

In a preprocessing step, bioclimatic variables for each year and cell are calculated: degree-day sum $> 5.6^{\circ}\text{C}$, coldest monthly mean, and a drought index based on water holding capacity, slope, aspect, monthly mean temperatures, and precipitation sums (cf. Bugmann and Cramer 1998). This is now handled by the R-package and GUI.

2.4 Initialization

At the beginning of each simulation run, all grid cells of the landscape are initialized with the stockability (0, 1) from an external input file. Simulations can either start from an empty landscape (without any trees present), or from an already forested landscape. In the latter case, the state variables have to be provided in a state text file. For the start in an empty landscape, the first a user defined number years are run with global seed input and without dispersal (inoculation).

2.5 Spin up

The easiest way for a spin up is to run a simulation over or longer than the spin up period, and to write out a statefile at the end of the spin up period. Subsequent simulations, e.g. for different scenarios can then be started from this statefile.

2.6 How to apply TreeMig

The compiled application must be located in a folder with subfolders E (containing the bioclimate files), P containing the species parameter files, C containing the control files, and subfolders R and S for writing the results and statefiles, respectively. The simulation is then controlled by the control file, which specifies the simulation settings and the input files for bioclimate, species parameter and the state- and immigration filexs, if used.

2.7 Technical information

The model is written in Fortran90/95, and can be compiled by GFortran via a makefile. It consists of 63 files, containing 90 subroutines and functions, and ca 15000 lines of code (with comments). Currently, compiled applications are available for WINDOWS and MacOS.

The Fast Fourier Transforms are performed using the GFT (General Fourier Transform) package v1.0.0 of Jalel Chergui, formerly at IDRIS, France. The parts of the package required for TreeMig are included in the TreeMig code.

The Fortran Netcdf library netcdf is linked statically to the model.