April 2002

User's Guide for Biome-BGC, Version 4.1.2

Source

Peter E. Thornton
Terrestrial Sciences Section
Climate and Global Dynamics Division
National Center for Atmospheric Research
1850 Table Mesa Dr.
Boulder, CO 80305

Steve W. Running
Numerical Terradynamics Simulation Group
School of Forestry
University of Montana
Missoula, MT 59812

General notes on Biome-BGC version 4.1.2

Biome-BGC is a computer model that simulates the storage and fluxes of water, carbon, and nitrogen within the vegetation, litter, and soil components of a terrestrial ecosystem. Biome-BGC is primarily a research tool, and many versions have been developed for particular purposes. The National Center for Atmospheric Research (NCAR) and the University of Montana's Numerical Terradynamic Simulation Group (NTSG) maintain benchmark code versions for public release, and update these benchmark versions periodically as new knowledge is gained on the research front. The code and executables accompanying this file represent the most recent benchmark version.

IMPORTANT NOTE: COPYING

Referencing Biome-BGC:

If you use Biome-BGC in your research, we request that you include the following acknowledgement in the relevant manuscripts:
"Biome-BGC version 4.1.2 was provided by Peter Thornton at the National Center for Atmospheric Research (NCAR), and by the Numerical Terradynamic Simulation Group (NTSG) at the University of Montana. NCAR is sponsored by the National Science Foundation."

Please also reference the following citation as the most recent and complete description of the current model version:
Thornton, P.E., Law, B.E., Gholz, H.L., Clark, K.L., Falge, E., Ellsworth, D.S., Goldstein, A.H., Monson, R.K., Hollinger, D., Falk, M., Chen, J. and Sparks, J.P., 2002. Modeling and measuring the effects of disturbance history and climate on carbon and water budgets in evergreen needleleaf forests. Agricultural and Forest Meteorology (in press).
(this citation will be updated as soon as possible on the model website).

If you have made any significant modifications to the code, please mention them in your manuscript.

This User's Guide is the only documentation released with Biome-BGC. The code itself contains extensive internal documentation, and users with specific questions about the algorithms used to estimate particular processes should read the comments in the appropriate source code files. The file bgc.c contains references to all the core science routines, and is a good starting point for this kind of inquiry. The file bgc_struct.h defines the data structures that are used to pass information between the process modules, and includes both a short text description and the units for each internal variable.

We are always interested to get reports from users about new applications of Biome-BGC, including any problems that come up. If you have questions about the code, appropriate model applications, possible programming errors, etc., please read this entire guide first, then feel free to contact:
Peter Thornton (thornton@ucar.edu)

UPDATES FROM PREVIOUS VERSIONS

Updates from version 4.1.1 to version 4.1.2

This code release (version 4.1.2) represents a minor update from version 4.1.1, primarily to fix bugs that were identified in version 4.1.1. These include:

1. An incorrect parameter was being used in the calculation of the daylight average air temperature in daymet.c. The parameter value in version 4.1.1 was 0.212, and the correct value, for consistency with the MT-CLIM and Daymet code, should be 0.45. The daylight average air temperature (tday) is used in the photosynthesis routine, and in the calculation of daytime leaf maintenance respiration. As an example of the net result of changing to the correct value, the example simulations described later in this guide show an increase in steady state leaf area index of about 10% and an increase in steady state net primary production of about 5%. Thanks to Michael Guzy at Oregon State University for finding this bug.
2. The logic in spinup_daily_allocation.c was not entirely consistent with that in daily_allocation.c. In particular, the modifications described below in going from version 4.1 to version 4.1.1 were implemented in daily_allocation.c, but not in spinup_daily_allocation.c. Changes were made to spinup_daily_allocation.c to correct this. This changes the time course of the spinup, but the changes in results once spinup is complete are insignificant. In other words, these changes do not have a noticeable impact on the steady state conditions following spinup. Thanks to Hirofumi Hashimoto at the University of Tokyo for finding this bug.
3. The treatment of deployment of retranslocated nitrogen to new growth, under conditions of non-limiting nitrogen availability, had the potential to send a flux of nitrogen from the plant N pool into the soil mineral N pool. This was not intended, and has been corrected (see daily_allocation.c and spinup_daily_allocation.c). Thanks to James Trembath at the Max Planck Institute for Biogeochemistry for finding this bug.
4. The header in the annual text output file indicated that the units for annual precipitation are cm/yr, but they are in fact mm/yr. The header text was corrected.
5. Several new variables have been added to the data structures and referenced in output_map_init.c. These changes have no impact on the model's numerical results.
6. Heterotrophic respiration fractions and base decomposition rates have been defined in bgc_constants.h. These were previously hardwired in several different places in the code.
7. Added a test in output_init.c to avoid allocation when the user has set ndaysout and nannout to 0 (e.g. in a spinup simulation).
8. The unix makefile had an error relating to multiple declarations for metarr_init.c. This caused problems for some compilers. It has been fixed.

Updates from version 4.1 to version 4.1.1

Version 4.1.1 represented a relatively minor update from version 4.1, which was released in February, 2000. Recent research efforts with Biome-BGC have concentrated on comparisons of model estimates of carbon and water fluxes with fluxes measured by eddy flux methods at a number of sites in the U.S. and in Europe. Through these comparisons, a number of model process representations were identified that did not agree with the flux measurements. Most of the changes from version 4.1 to version 4.1.1 are the result of these comparisons. The changes are mostly in the handling of C and N allocation across multiple years, for example in the deployment of N retranslocated from senescing leaves during subsequent growing seasons. These changes represent the philosophy of "simplest possible process representation" in operation. The original mechanisms (version 4.1) were based on the simplest possible treatment, in the absence of data to support more sophisticated mechanisms. Now that several years of flux data are available from multiple sites, we can see that the simple mechanisms capture the interannual dynamics, but err in the representation of the seasonal dynamics under some circumstances. On this basis, the next logical steps toward increasing complexity in the process representation were added, resulting in a much better representation of the seasonal cycles of fluxes without impairing the interannual behavior. Each change is described in detail below. Thanks to Galina Churkina, Bev Law, Peter Anthoni, and Emil Cienciala for providing data and/or assistance in the analysis of the model dynamics, which made these improvements possible.

1. **Deployment of retranslocated leaf N for new growth:** In version 4.1, the entire pool of retranslocated N was available for new growth on each day, which resulted in an unrealistic spike in productivity in the early part of the growing season for an N-limited system. In version 4.1.1, only a fixed proportion of the retranslocated N pool is available for new growth on each day. For more details, see the comments in the header of the source code file daily_allocation.c**.**
2. **Handling of C-pool deficit:** In version 4.1, any time there is a deficit in the C pool state variable (e.g. as occurs in the spring for a temperate-zone deciduous forest simulation, due to zero photosynthetic potential and moderate respiration demands from live wood through the winter), the early growing season production is used to eliminate this deficit. This resulted in an slight but abrupt shift in the early growing season at the point when the deficit was alleviated, since it is assumed that additions to the Cpool do not incur growth respiration costs. In version 4.1.1, this mechanism is made slightly more complicated, with allocations to eliminate a C pool deficit made at a rate that would alleviate the deficit if continued for a specified number of days. For more details, see the comments in the header of the source file daily_allocation.c.
3. **Accumulation of soil mineral N:** In version 4.1, the only explicit controls on the accumulation of mineral forms on N in the soil were due to leaching, fire, plant and microbial N uptake, and a fractional loss to denitrification as part of the mineralization process. Under non-limiting N conditions, mineral N could rise to unrealistic levels. A very simple additional denitrification pathway was added in version 4.1.1, which removes a constant fraction of the mineral N remaining on each day, after all the other processes have been reconciled. For more details, see the comments in the header of the source file daily_allocation.c.
4. **Default combustion of dead woody material:** This change is related to the very simple treatment of fire effects included in version 4.1. 5% of the dead (non-respiring) stem material was assumed to be consumed in a fire. This default has been changed to 20%. See further comments in the header for the source code file mortality.c.
5. **New variables in the default ASCII text output file:** Output for annual total precipitation and annual average air temperature have been added to the standard ascii text output file (see further discussion later in this User's Guide).

INPUT FILES: OVERVIEW

Biome-BGC uses at least three input files each time it is executed. A brief description of all files is given first, followed by detailed discussions of each file.

The first required input file is called the initialization file. It provides general information about the simulation, including a description of the physical characteristics of the simulation site, a description of the time-frame for the simulation, the names of all the other required input files, the names for output files that will be generated, and lists of variables to store in the output files.

The second required input file is the meteorological data file. It contains daily values for temperature, precipitation, humidity, radiation, and daylength at the simulation site. It can contain any number of years of data.

IMPORTANT NOTE: LEAP-YEARS

The Biome-BGC code assumes that all years have 365 days, so meteorological data files should be edited to remove one day from leap years.

The third required input file is the ecophysiological constants file. It contains an ecophysiological description of the vegetation at a site, including parameters such as leaf C:N ratio, maximum stomatal conductance, fire and non-fire mortality frequencies, and allocation ratios.

There is also an optional input file called a restart file. This file provides all the information required to start one simulation from the endpoint of a previous simulation. The results of the restarted simulation are exactly the same as if the original simulation had been carried out for additional years. This option is typically used in conjunction with the spinup mode (described below) to initialize the soil carbon and nitrogen components for a site without adequate measurements of these characteristics. Users need only know the names of input and output restart files, and need not be concerned with the file contents.

INITIALIZATION FILE: DETAILS

There are several example initialization files (called ini files for short) in the ini directory of this code release, and you should refer to one while reading the following discussion.

The initialization file is broken into sections with a keyword starting each new section and a special keyword at the bottom of the file. This organization helps to ensure that the proper format for the initialization file is maintained while editing for new simulations. The order of the sections and the order of the lines within each section is critical, and changes in order will result in failed or flawed executions. It is highly recommended that the example ini files provided with this release be copied to a safe directory for future reference on proper formatting. There is a fair amount of error-checking on the ini file format, but it is still possible to scramble the order of lines in a way that the program does not detect. In this case the model will run, but the results will be garbage. Copying the template is the easiest way to make a new ini file with assurance of the correct format, but remember to replace ALL of the parameters with those for the new site. It is not necessary to have blank lines between the end of one section and the keyword for the next section, but keeping them in makes the ini file more readable.

Each line can contain up to 100 characters of comment information, after a keyword or after an input parameter. This information is for your reference in keeping the correct format in the ini file, and is ignored by the program. Every line is required, even if flags are set that mean the information on a line will be ignored.

The first line of the file is for header information that helps you keep track of which ini file is for which simulation when you are doing a sequence of simulations. This line can contain up to 100 characters of information. There is no keyword for this header line.

MET_INPUT block

The next section begins with the keyword MET_INPUT. It has the following two lines:
1) name (including path if appropriate) of input meteorological data file
2) number of header lines in meteorological data file
<EXAMPLE>

```
MET_INPUT     (keyword) start of meteorology file control block
metdata/miss5093.mtc41  meteorology input filename
4             (int)     header lines in met file
```

<END EXAMPLE>
NOTES: See the section of this guide titled METEOROLOGICAL DATA FILE: DETAILS for more discussion of header lines and file format.

RESTART block

The next section begins with the keyword RESTART. It has the following five lines:
1) flag (1 or 0) for reading (1) or not reading (0) a restart file from the end of a previous simulation
2) flag (1 or 0) for writing (1) or not writing (0) a restart file at the end of this simulation
3) flag (1 or 0) for met year from restart file (1) or met year reset to beginning of record (0)
4) input restart filename (including path if appropriate)
5) output restart filename (including path if appropriate)
<EXAMPLE>

```
RESTART       (keyword) start of restart control block
1             (flag)    1 = read restart file     0 = don't read restart file
0             (flag)    1 = write restart file    0 = don't write restart file
1             (flag)    1 = use restart metyear   0 = reset metyear
restart/test.endpoint    input restart filename
restart/test.endpoint    output restart filename
```

<END EXAMPLE>
NOTES: Lines 3) and 4) are only relevant when line flag in line 1) is set to 1. Line 5) is only relevant when flag in line 2) is set to 1. In most cases the flag in line 3) should be set to 1.

TIME_DEFINE block

The next section begins with the keyword TIME_DEFINE. It has the following five lines:
1) number of years of data in meteorological input data file
2) number of years to run for this simulation.

3) first simulation year (e.g. 1998)
4) flag (1 or 0) for spinup simulation (1) or normal simulation (0)
5) maximum number of years to run in spinup mode
<EXAMPLE>

```
TIME_DEFINE   (keyword - do not remove)
44            (int)       number of meteorological data years
100           (int)       number of simulation years
1950          (int)       first simulation year
0             (flag)      1 = spinup simulation    0 = normal simulation
6000          (int)       maximum number of spinup years (if spinup simulation)
```

<END EXAMPLE>
NOTES: If line 2) is greater than line 1), then the meteorological data file is "recycled" enough times to satisfy the requested number of simulation years. Line 3) defines the starting point for the simulation output, and is mainly used in the simple text output file (described below). Line 4) sets the simulation mode for either a spinup run or a normal run. If spinup mode is selected, the meteorological data file will be recycled enough times to establish steady-state conditions in the soil carbon and nitrogen pools. The number of years of simulation for a spinup run will depend on the climate and vegetation characteristics, but will not exceed the number of years specified on line 5). Line 2) has no effect for a spinup run. If a spinup run reaches the maximum number of years specified in line 5), it is likely that the resulting final soil carbon and nitrogen pools are not equilibrated with the climate, usually indicating a long-term net sink of carbon. This is observed for example in some boreal climates, where the model predicts long-term accumulation of organic matter (peat formation).

CLIM_CHANGE block
The next section begins with the keyword CLIM_CHANGE. It has the following five lines:
1) temperature offset for maximum temperature
2) temperature offset for minimum temperature
3) scaling factor for precipitation
4) scaling factor for vapor pressure deficit
5) scaling factor for incoming shortwave radiation
<EXAMPLE>

```
CLIM_CHANGE   (keyword - do not remove)
0.0           (deg C)   offset for Tmax
0.0           (deg C)   offset for Tmin
1.0           (DIM)     multiplier for Prcp
1.0           (DIM)     multiplier for VPD
1.0           (DIM)     multiplier for shortwave radiation
```

<END EXAMPLE>
NOTES: This section is included to facilitate simulation of simple climate change effects on ecosystem processes. The default values are shown in the example above, which result in no changes to the input meteorological data. Increases or decreases in temperature (Tmax and/or Tmin) can be introduced with lines 1) and 2). The values indicated are simply added to the daily values from the meteorological data file. For precipitation, VPD, and radiation, the daily values from the input file are multiplied by the values on lines 3), 4), and 5) to get new daily values. This mechanism does not allow for the simulation of seasonal differences in climate change.

CO2_CONTROL block
The next section begins with the keyword CO2_CONTROL. It has the following 3 lines:
1) flag (0,1,2) controlling CO2 concentration: 0=constant, 1=varying using values from a special file, 2=constant, but use the CO2 values in the specified file to control levels of N deposition (see below for N deposition options).
2) the value to use for constant CO2 concentration (ppm)
3) the filename for annual CO2 levels (see notes below for format information)
<EXAMPLE>

```
CO2_CONTROL   (keyword - do not remove)
0             (flag)    0=constant 1=vary with file 2=constant, file for Ndep
294.842       (ppm)     constant atmospheric CO2 concentration
xxxxxxxxxxx   (file)    annual variable CO2 filename
```

<END EXAMPLE>
NOTES: When line 1) is set to 0 or 2, then the value on line 2) sets the constant CO2 level for the entire simulation. When line 1) is set to 1), then the file named on line 3) is used to define the annual timeseries of CO2 concentration.

This file must have one line for each simulation year (the number given on line 2 of the TIME_DEFINE section), and the format of each line should be like this:

```
1895  294.842
```

where the first value on the line is the year, and the second value is the CO2 mole fraction (ppm). When line 1) is set to 2, the value on line 2) is used for the constant CO2 level for the entire simulation, but the file on line 3) is used to control the time sequence of changes in N deposition.

SITE block

The next section begins with the keyword SITE. It has the following nine lines:
1) rooting zone soil depth, after accounting for the fraction of the rooting zone occupied by rocks
2) soil texture: percent sand (by volume in rock-free soil)
3) soil texture: percent silt (by volume in rock-free soil)
4) soil texture: percent clay (by volume in rock-free soil)
5) site elevation in meters above mean sea level
6) site latitude in decimal degrees (negative values for southern hemisphere)
7) site shortwave albedo
8) annual rate of atmospheric nitrogen deposition (wet + dry deposition)
9) annual rate of symbiotic + asymbiotic nitrogen fixation

<EXAMPLE>

```
SITE          (keyword) start of site physical constants block
1.0           (m)       effective soil depth (corrected for rock fraction)
30.0          (%)       sand percentage by volume in rock-free soil
50.0          (%)       silt percentage by volume in rock-free soil
20.0          (%)       clay percentage by volume in rock-free soil
977.0         (m)       site elevation
46.8          (degrees) site latitude (- for S.Hem.)
0.2           (DIM)     site shortwave albedo
0.0001        (kgN/m2/yr) wet+dry atmospheric deposition of N
0.0008        (kgN/m2/yr) symbiotic+asymbiotic fixation of N
```

<END EXAMPLE>

NOTES: The values on lines 2, 3, and 4 must sum exactly to 100.

RAMP_NDEP block

The next section begins with the keyword RAMP_NDEP. It has the following three lines:
1) flag (1 or 0) for variable nitrogen deposition (1) or constant nitrogen depostition (0)
2) the reference year for industrial nitrogen deposition value
3) the reference value for industrial nitrogen deposition

<EXAMPLE>

```
RAMP_NDEP     (keyword - do not remove)
0             (flag) do a ramped N-deposition run? 0=no, 1=yes
2099          (int)  reference year for industrial N deposition
0.0001        (kgN/m2/yr) industrial N deposition value
```

<END EXAMPLE>

NOTES: when line 1) is set to 0, the nitrogen deposition level is constant and is determined by line 8) of the SITE section. When line 1) is set to 1, the nitrogen deposition levels vary according to the time trajectory of CO2 mole fractions, using two reference values to scale N deposition level to CO2 concentration. For the case of varying N deposition, the value on line 8) of the SITE section is taken as the N deposition level in the first simulation year, and the value specified on line 3) of this section is taken as the N deposition level in the year specified on line 2) of this section. In this case the values for CO2 concentration given in the file listed in line 3) of the CO2_CONTROL section are used to scale the N deposition value for each year according to changes in the CO2 concentration. The assumption here is that N deposition increases as the anthropogenic CO2 source increases .This assumption has some obvious limitations, but it can provide a plausible time sequence of N deposition when only a historical and a current value are available.

EPC_FILE block

The next section begins with the keyword EPC_FILE. It has a single line:
1) the name of the input ecophysiological constants file

<EXAMPLE>

```
EPC_FILE      (keyword - do not remove)
epc/enf.epc   (file) evergreen needleleaf forest
```
<END EXAMPLE>
NOTES: There is a default set of ecophysiological parameter files supplied with this release of the Biome-BGC code, which has been developed through extensive literature searches and statistical summarization. Details on the format and interpretation of these files are given in a later section. Although there is a lot of careful work behind the default values that are supplied with the code, these values are intentionally generalized to represent very coarse distinctions between vegetation types. If you have reliable data for a particular species or functional type on or near your site, you should use it to modify the default values that we have supplied.

W_STATE block
The next section begins with the keyword W_STATE. It has the following two lines:
1) initial snowpack water content (start of simulation)
2) initial soil water content as a proportion of saturation
<EXAMPLE>
```
W_STATE       (keyword) start of water state variable initialization block
0.0           (kg/m2)   water stored in snowpack
0.5           (DIM)     initial soil water as a proportion of saturation
```
<END EXAMPLE>
NOTES: When using a restart file these values are ignored. Otherwise, they set the initial conditions for the water state variables (storage components) on the first day of simulation. Line 1) sets the snowpack, and is in water equivalent units, where kg water/m2 is equivalent to mm of water. The second line controls the initial soil water content. This is set as a proportion of saturation so that the user is not required to know the saturation water content of the site (depends on texture and depth). For a spinup run, these values are used as the initial conditions.

C_STATE block
The next section begins with the keyword C_STATE. It has the following eleven lines:
1) peak leaf carbon to be attained during the first simulation year
2) peak stem carbon to be attained during the first year
3) initial coarse woody debris carbon (dead trees, standing or fallen)
4) initial litter carbon, labile pool
5) initial litter carbon, unshielded cellulose pool
6) initial litter carbon, shielded cellulose pool
7) initial litter carbon, lignin pool
8) soil carbon, fast pool
9) soil carbon, medium pool
10) soil carbon, slow pool
11) soil carbon, slowest pool
<EXAMPLE>
```
C_STATE       (keyword) start of carbon state variable initialization block
0.001         (kgC/m2)  first-year maximum leaf carbon
0.0           (kgC/m2)  first-year maximum stem carbon
0.0           (kgC/m2)  coarse woody debris carbon
0.0           (kgC/m2)  litter carbon, labile pool
0.0           (kgC/m2)  litter carbon, unshielded cellulose pool
0.0           (kgC/m2)  litter carbon, shielded cellulose pool
0.0           (kgC/m2)  litter carbon, lignin pool
0.0           (kgC/m2)  soil carbon, fast microbial recycling pool
0.0           (kgC/m2)  soil carbon, medium microbial recycling pool
0.0           (kgC/m2)  soil carbon, slow microbial recycling pool
0.0           (kgC/m2)  soil carbon, recalcitrant SOM (slowest)
```
<END EXAMPLE>
NOTES: As with the W_STATE section, these values are ignored when using an input restart file, since all the initial conditions are defined by the endpoint from the previous simulation (usually a spinup run). In starting a spinup run these values have to be supplied. The above example shows the best approach to these initial conditions for a spinup run: start with very low initial leaf area, and no carbon in any of the other pools. This is essentially a primary succession simulation, starting with no organic matter and a very meager colonizing plant cover. The development of soil organic matter depends on the site's climate and external fluxes of nitrogen (deposition, fixation, leaching,

volatilization), as well as on the vegetation type. The end result is that soil and litter pools are in equilibrium with climate and N deposition/loss rates. If these values are assigned without performing a spinup run, then the initial model dynamics are likely to represent a transient response to disequilibrium conditions between the specified initial conditions and the specified climate and N deposition/loss. This can produce very misleading signals in net ecosystem exchange of carbon that can persist for many hundereds of years. For this reason it is recommended that new simulations be started with a spinup run, then a restart from the spinup conditions, as in the examples described later in this guide.

N_STATE block

The next section begins with the keyword N_STATE. It has the following two lines:
1) litter nitrogen associated with labile litter carbon pool
2) soil mineral nitrogen pool
<EXAMPLE>

```
N_STATE       (keyword) start of nitrogen state variable initialization block
0.0           (kgN/m2)  litter nitrogen, labile pool
0.0           (kgN/m2)  soil nitrogen, mineral pool
```

<END EXAMPLE>
NOTES: As with W_STATE and C_STATE, the N_STATE values are ignored when using an input restart file. For the reasons outlineed above, it is recommended that new simulations be started with a spinup run, and then continued from the end of the spinup run with a restart file. In the case of a spinup run, it is recommended that the N_STATE values both be set to 0.0, as in the example shown here.

OUTPUT_CONTROL block

The next section begins with the keyword OUTPUT_CONTROL. It has the following 6 lines:
1) text string giving the prefix for all output files, including path if appropriate.
2) flag (1 or 0) to write (1) or not write (0) a binary output file with daily values
3) flag (1 or 0) to write (1) or not write (0) a binary output file with monthly averages of the daily variables
4) flag (1 or 0) to write (1) or not write (0) a binary output file with annual averages of the daily variables
5) flag (1 or 0) to write (1) or not write (0) a binary output file with year-end values
6) flag (1 or 0) to send (1) or not send (0) simultation progress information to the screen
<EXAMPLE>

```
OUTPUT_CONTROL   (keyword - do not remove)
outputs/test     (text) prefix for output files
1   (flag)  1 = write daily output   0 = no daily output
0   (flag)  1 = monthly avg of daily variables  0 = no monthly avg
0   (flag)  1 = annual avg of daily variables   0 = no annual avg
0   (flag)  1 = write annual output  0 = no annual output
1   (flag)  for on-screen progress indicator
```

<END EXAMPLE>
NOTES: As described below, the user can select two different groups of variables for output, one group that will be used for daily output and the monthly and annual averages of daily values, and a second group that is used for writing year-end (day 365) values for each simulation year. The flag for daily output on line 2) does not need to be set to 1 in order to get monthly and/or annual averages of the daily output variables. Setting any of the output flags to 1 (lines 2-5) will result in the creation of an output file, having the specified path and filename prefix (from line 1) and the following suffix depending on which type of output was requested:

daily output suffix	= ".dayout"
monthly average output suffix	= ".monavgout"
annual average output suffix	= ".annavgout"
year-end output suffix	= ".annout"

DAILY_OUTPUT block

The next section begins with the keyword DAILY_OUTPUT. The number of lines can vary depending on the number of output variables requested, as follows:
1) the number of daily output variables requested. This value can be 0.
2) the index number for the first requested daily output variable
3) etc.
<EXAMPLE>

```
DAILY_OUTPUT     (keyword)
3      (int) number of daily variables to output
43      wf.soilw_trans
509     epv.proj_lai
620     summary.daily_npp
```
<END EXAMPLE>
NOTES: The number of lines after line 1) must equal the number of daily output variables specified on line 1). There are more than 500 possible output variables, and the use of an index is the simplest way to allow the user access to all of them without introducing complicated parsing routines that would tend to clutter the code. The index value for each variable is defined in the file output_map_init.c, where variables are organized into logical groups (located in directory src/bgclib_v4.1.1 for UNIX release, and in directory src for PC release). For example, all the carbon flux variables are listed together, and all the water flux variables are listed together. The organization of the output index mapping is guided by the data structures that are used to pass information among process modules within the code. These structures are defined in the file bgc_struct.h (located in directory src/include for UNIX release, and directory src for PC release). This indexing system is admittedly awkward for new users, since it requires first knowing which data structure element to reference and then getting the index value for that element. On the other hand, it has the advantages of being unambiguous and providing a direct user interface to the logical organization of the data structures used in the code. Consulting the output_map_init.c and bgc_struct.h gives a comprehensive view of the possible output variables, and provides a useful introduction to the logical structure of process representation within Biome-BGC. As in all ini file lines, all text after the first value on each output specification line is ignored by the program. It is recommended that as you add or delete output variable index values, you also add a descriptive comment reminding you what variable is being requested. The variables requested in this section are available for daily output, for monthly average output, and for annual average output, specified by the flags in the OUTPUT_CONTROL section.

ANNUAL_OUTPUT block
The next section begins with the keyword ANNUAL_OUTPUT. It has exactly the same format as just described for the DAILY_OUTPUT section.
<EXAMPLE>
```
ANNUAL_OUTPUT    (keyword)
6               (int)   number of annual output variables
545     0 annual maximum projected LAI
636     1 vegetation C
637     2 litter C
638     3 soil C
639     4 total C
307     5 soil mineral N
```
<END EXAMPLE>
NOTES: The variables requested in this section are reported once each year (yearday 365) and are stored in the *.annout file. This provides a once-yearly snapshot of the system state and activity, and so it is most appropriate for recording system states that are changing relatively slowly, such as soil carbon, vegetation carbon, etc. Remember that if annual averages of system behavior or system state are desired, then it is necessary to specify these variables in the DAILY_OUTPUT section, and set the annual averaging of daily output flag to 1 (line 4 in OUTPUT_CONTROL section).

END_INIT block
The final section of the ini file consists only of the keyword END_INIT, signaling the end of the file. This signal is used to make sure that the proper number of lines have been read from the ini file.
<EXAMPLE>
```
END_INIT      (keyword) indicates the end of the initialization file
```
<END EXAMPLE>

METEOROLOGICAL DATA FILE: DETAILS
As described above, the MET_INPUT section of the ini file names a file containing the meteorological data used to drive a Biome-BGC simulation. This file can contain any number of header lines, followed by numeric values for the meteorological data. The number of header lines must be specified after the filename in the MET_INPUT section of the ini file. Below is an example of the top of a meteorological data file having four header lines:

<EXAMPLE>

```
Missoula, 1950-1993
MTCLIM v4.3 OUTPUT FILE : Mon Feb 7 10:15:00 2000
  year  yday    Tmax    Tmin    Tday    prcp      VPD     srad  daylen
              (deg C) (deg C) (deg C)    (cm)     (Pa)  (W m-2)     (s)
  1950     1   -3.90  -13.90   -6.65    0.10   158.19   123.31   30229
  1950     2   -7.80  -21.70  -11.62    0.00   136.27   183.78   30284
  1950     3  -16.10  -23.30  -18.08    0.00    53.36   140.67   30344
  1950     4  -11.70  -20.60  -14.15    0.10    83.07   119.72   30408
  1950     5  -13.90  -25.00  -16.95    0.00    78.17   177.64   30476
  1950     6    0.60  -14.40   -3.53    0.10   264.00   142.03   30549
  1950     7    4.40   -4.40    1.98    0.00   297.13   158.19   30626
  1950     8   -2.20  -11.70   -4.81    0.10   182.84   126.15   30707
```

<END EXAMPLE>

In this case, the meteorological input file is one generated by the MTCLIM (version 4.3) program, as indicated in the header lines. Meteorological data files generated by MTCLIM version 4.3 will always have four header lines, with the last two header lines describing the variables and their units. This example also illustrates the required input variables, their units, and their order. Biome-BGC requires these variables, using these units, and in this order, with no additional variables, for every simulation. The spacing between variables is not important, as long as there is some white space between each value on a line. Do not use commas or other non-white space separators.

The nine values required for each day are as follows:

1) year: the numerical year, repeated for each yearday
2) yday: the numerical day of the year, values must start with 1 and end with 365
3) Tmax: the daily maximum temperature (°C)
4) Tmin: the daily minimum temperature (°C)
5) Tday: the average daytime temperature (sunrise to sunset, °C)
6) prcp: the daily total precipitation, (cm)
7) VPD: the daylight average vapor pressure deficit (Pa)
8) srad: the daylight average shortwave radiant flux density (W/m^2)
9) daylen: the daylength (sunrise to sunset, seconds)

The meteorological input data file can contain any number of years of data, and each year must have exactly 365 days, with one line of data per day, and no separations between years. This release of the code has an example met data file containing 44 years of daily weather values for Missoula, MT, which shows the correct formatting for multiple-year files. NOTE: Biome-BGC expects each year to have 365 days of data, so leap years will have to be truncated by eliminating one day. We suggest eliminating December 31 of a leap year, since then the yearday numbering doesn't have to be adjusted. We have found that this truncation has negligible effects on most simulations. The rationale for requiring 365-day years is that the met data files are commonly of shorter duration than the intended simulation, so they must be "recycled" enough times to match the number of requested simulation years. Recycling is also essential in the spinup runs. If the number of years in the met data file is not a multiple of four, then the handling of leap days in the code becomes very tedious, and introduces several layers of testing and manipulation between the input and the process algorithms. We have accepted the small errors introduced by eliminating one day from leap years in exchange for code that is clear and easier to maintain.

Meteorological data files can be assembled from observations if all of the required parameters have been measured for the site of interest. They can also be generated using the MTCLIM program using only observations of temperature and precipitation. The MTCLIM code and documentation is available from the NTSG website:
www.forestry.umt.edu/ntsg (go to the section "Models to Download")

or from Peter Thornton (thornton@ucar.edu).

ECOPHYSIOLOGICAL CONSTANTS FILE: DETAILS

This input file defines the ecophysiological characteristics of the vegetation type being simulated. It is kept separate from the initialization file so that multiple initialization files can reference the same ecophysiology constants without cutting and pasting. Researchers at NTSG have spent considerable effort summarizing a large number of

ecophysiological studies from the literature to come up with a set of default parameterizations for a small number of highly aggregated vegetation classes. These *.epc files are supplied with this distribution of the Biome-BGC code, and users are encouraged to use them as a first approximation of the appropriate parameters for a new simulation. If you have good measurements from your site(s) relating to any of these parameters, you should replace the defaults with your observations. Users are cautioned that some of these parameters show strong covariance, and so replacing some but not others with local observations may reduce the quality of results. For example, canopy average specific leaf area, leaf C:N, and fraction of leaf N in Rubisco tend to covary, so if you replace any of these default values you should consider replacing all of them. Consult with your local ecophysiologist if you aren't sure about reparameterization of the default *.epc files.

Note that all *.epc files must have the same parameter lines, in the same order, but that not all lines are relevant to all vegetation types. Lines marked with (*) before the units in the default *.epc files indicate an irrelevant parameter for that vegetation type, and any numeric value can be substituted in these places without effect.

The following section describes each line of an epc file, in the required order. The line number is followed by the units for the parameter in question and the short text description included in the default epc files (some parameters are dimensionless, and these units are given as DIM). This is followed by a detailed description of the parameter(s) in question .

line 1

```
ECOPHYS       (keyword) start of canopy ecophysiological constants block
```

This is a keyword used by the code to interpret the start of a block of ecophysiological data. The first line of each *.epc file must start with ECOPHYS.

line 2

```
(flag)    1 = WOODY             0 = NON-WOODY
```

An integer flag specifying the growth form, where 1 for woody includes both tree and shrub vegetation types, and 0 for non-woody includes grasses as well as other primarily herbaceous plants.

line 3

```
(flag)    1 = EVERGREEN         0 = DECIDUOUS
```

An integer flag specifying the leaf habit, where 1 for evergreen includes leaf habits that retain at least some of their foliage year-round, and 0 for deciduous includes leaf habits in which all foliage is absent at some point during a year. Either value for this flag can apply to both woody and non-woody growth forms.

line 4

```
(flag)    1 = C3 PSN            0 = C4 PSN
```

An integer flag specifying the photosynthetic pathway, where 1 indicates that the C3 photosynthesis model should be invoked, and 0 indicates that the C4 model should be invoked. Although this flag can be set to 0 for any combination of the other parameters, use of the C4 model should be restricted to grasses and herbaceous plants.

line 5

```
(flag)    1 = MODEL PHENOLOGY   0 = USER-SPECIFIED PHENOLOGY
```

An integer flag specifying how the phenological control for a simulation will be exercised. A value of 1 invokes the internal phenology routine, while a value of 0 indicates that the user will supply information on the yeardays for the start of new growth and the end of the litterfall period. See below for more details on how to set these parameters for the case of user-specified phenology.

lines 6 and 7

```
(yday)    yearday to start new growth  (when phenology flag = 0)
(yday)    yearday to end litterfall  (when phenology flag = 0)
```

Two integers specifying the yearday for the start of new leaf growth, and the yearday for the end of the litterfall season, respectively. Relevant only when the phenology flag = 0 (see above). There are several IMPORTANT NOTES about setting these values:
1) To suppress new leaf growth entirely, for example in the case of a simulation concerned with bare soil processes, set both of these values to -1.
2) Yearday values for these parameters start at 0 and go to 364. Note that BIOME-BGC does not accept leap-years, i.e. all years are by definition 365 days long.
3) Northern and Southern hemisphere yeardays are treated differently for these parameters. In the northern hemisphere, yearday 0 = Jan 1, while in the southern hemisphere yearday 0 = July 2. This allows the same yearday values to be used to specify deciduous growth habit in the northern and southern temperate zones.
4) If the leaf habit flag is set to evergreen (1), and the phenology model
flag is set to user-specified (0), these values do not have any effect.

lines 8 and 9

```
(prop.)   transfer growth period as fraction of growing season
(prop.)   litterfall as fraction of growing season
```

These two parameters determine the duration of the transfer growth and litterfall periods, and are defined as proportions of the period between the start of new growth and the end of litterfall. These parameters must be set by the user REGARDLESS of whether internal model phenology or user-specified phenology is specified in the phenology model flag. These parameters can take any values from 0.0 to 1.0, where a value of 0.0 indicates that all transfer growth or all litterfall occurs in a single day, and a value of 1.0 indicates that transfer growth or litterfall occur throughout the growing season. Transfer growth is the growth derived from carbon and nitrogen resources stored over the course of the previous growing season. It is the growth that produces the first flush of new leaves in the spring for deciduous plants. NOTE that when the leaf habit flag is set for evergreen (1), both transfer growth and litterfall are assumed to occur at constant rates throughout the year, and the specification of these two parameters has no effect.

line 10

```
(1/yr)    annual leaf and fine root turnover fraction
```

Determines leaf and fine root turnover for evergreen plants. This is the fraction of the annual maximum leaf and fine root mass that will be dropped in the following year as litter. It is the reciprocal of the leaf longevity, so a plant that retains its leaves an average of two years would have a leaf/fine root turnover of 0.5. Note that leaf and fine root phenology are assumed to be entirely synchronized for all vegetation types. ALSO NOTE that when the leaf habit is specified as deciduous (0), this parameter is always assumed to be 1.0, and will be reset inside the code if the user specifies any other value.

line 11

```
(1/yr)    annual live wood turnover fraction
```

Determines livewood turnover to deadwood for all woody types (deciduous and evergreen). IMPORTANT NOTE about the definition of livewood and deadwood in BIOME-BGC: Livewood is defined as the actively respiring woody tissue, that is, the lateral sheathing meristem of phloem tissue, plus any ray parenchyma extending radially into the xylem tissue. Deadwood consists of all the other woody material, including the heartwood, the xylem, and the bark. It has been common in many tree models, including previous versions of BIOME-BGC, to divide the woody tissue into two compartments called "sapwood" and "heartwood", where sapwood is usually defined as the sum of phloem and xylem, with heartwood defined as the non-conducting woody tissue. The current treatment ignores the distinction between water-conducting xylem and non-conducting heartwood.

line 12

```
(1/yr)    annual whole-plant mortality fraction
```

This parameter specifies the fraction of all plant pools that will be removed and sent to the litter compartments over the course of a year. This is one mechanism by which woody material (live and dead) leaves the plant pool and enters the litter pools to be made available for subsequent decomposition. It is the conceptual equivalent of wind-throw, since all plant pools, living and dead, are attenuated at the same rate. This annual proportion is converted internally to a daily rate, and whole-plant mortality is assumed to go on at a constant rate throughout the year.

line 13
```
(1/yr)    annual fire mortality fraction
```

This parameter specifies the fraction of plant pools subject to fire, on average, each year. The current treatment ignores the timing of individual fire events, taking a long-term view of the fire process, in which some fraction of the community is subject to fire each year, at a rate commensurate with the long-term fire frequency. For example, in a system with a stand-replacing fire return interval of 100 yrs, this parameter would be set to 0.01.

line 14
```
(ratio)   (ALLOCATION) new fine root C : new leaf C
```

Sets the ratio of new fine root growth to new leaf growth. This is a constant, and applies to both current year growth, and growth resources stored for the next year's transfer growth.

line 15
```
(ratio)   (ALLOCATION) new stem C : new leaf C
```

Sets the ratio of new stem growth to new leaf growth. Stem growth includes both the live and dead woody components, as described above. This ratio is constant, and applies to both current year growth and growth resources stored for the next year's transfer growth.

line 16
```
(ratio)   (ALLOCATION) new live wood C : new total wood C
```

Sets the ratio of new live wood to new total wood. This ratio is constant, and applies to both current year growth and growth resources stored for the next year's transfer growth.

line 17
```
(ratio)   (ALLOCATION) new croot C : new stem C
```

Sets the ratio of new coarse root growth to new stem growth. This ratio is constant, and applies to both current year growth and growth resources stored for the next year's transfer growth.

line 18
```
(prop.)   (ALLOCATION) current growth proportion
```

Sets the proportion of daily production that is displayed immediately as new growth, with the remainder stored for the next year's transfer growth.

line 19
```
(kgC/kgN) C:N of leaves
```

Mass ratio of carbon : nitrogen in live leaves.

line 20
```
(kgC/kgN) C:N of leaf litter, after retranslocation
```

Mass ratio of carbon: nitrogen in freshly fallen leaf litter. This can only be higher than or equal to the C:N for live leaves, i.e. retranslocation can only be positive or zero. Retranslocation is the removal of nitrogen from the leaves prior to litterfall. The model does not consider the possibility of retranslocation of carbon out of leaves prior to litterfall.

line21
```
(kgC/kgN) C:N of fine roots
```

Mass ratio of carbon : nitrogen in fine roots. The model assumes that there is no retranslocation of nitrogen out of fine roots prior to litterfall, so this is the only C:N parameter for fine roots.

line 22
`(kgC/kgN) C:N of live wood`

Mass ratio of carbon : nitrogen for live wood (phloem and ray parenchyma). This will typically be a much smaller value than the average C:N for all woody parts, and is typically found to be close to that for fine roots.

line 23
`(kgC/kgN) C:N of dead wood`

Mass ratio of carbon : nitrogen for dead wood (bark, xylem, heartwood).

line 24
`(DIM)  leaf litter labile proportion`

The proportion of leaf litter mass in the labile fraction, usually defined as that fraction soluble in hot water/alcohol. Labile, cellulose, and lignin fractions for leaf litter must sum to 1.0.

line 25
`(DIM)  leaf litter cellulose proportion`

The proportion of leaf litter mass in the cellulose fraction, usually defined as that fraction soluble in a mild acid solution, after extraction of the water/alcohol soluble fraction. Labile, cellulose, and lignin fractions for leaf litter must sum to 1.0.

line 26
`(DIM)  leaf litter lignin proportion`

The proportion of leaf litter mass in the lignin fraction, usually defined as the remaining fraction after labile and cellulose fractions are removed, as described above. Labile, cellulose, and lignin fractions for leaf litter must sum to 1.0.

line 27
`(DIM)  fine root labile proportion`

The proportion of fine root mass in the labile fraction, usually defined as that fraction soluble in hot water/alcohol. Labile, cellulose, and lignin fractions for fine root must sum to 1.0.

line 28
`(DIM)  fine root cellulose proportion`

The proportion of fine root mass in the cellulose fraction, usually defined as that fraction soluble in a mild acid solution, after extraction of the water/alcohol soluble fraction. Labile, cellulose, and lignin fractions for fine root must sum to 1.0.

line 29
`(DIM)  fine root lignin proportion`

The proportion of fine root mass in the lignin fraction, usually defined as the remaining fraction after labile and cellulose fractions are removed, as described above. Labile, cellulose, and lignin fractions for fine root must sum to 1.0.

line 30
`(DIM)  dead wood cellulose proportion`

The proportion of dead wood mass in the cellulose fraction, usually defined as that fraction soluble in a mild acid solution, after extraction of the water/alcohol soluble fraction. Cellulose and lignin fractions for dead wood must sum to 1.0.

line 31
```
(DIM)     dead wood lignin proportion
```

The proportion of dead wood mass in the lignin fraction, usually defined as the remaining fraction after labile and cellulose fractions are removed, as described above. Cellulose and lignin fractions for dead wood must sum to 1.0.

line 32
```
(1/LAI/d) canopy water interception coefficient
```

The proportion of daily rainfall that can be intercepted and retained on the canopy per unit of projected leaf area index.

line 33
```
(DIM)     canopy light extinction coefficient
```

The Beer's law extinction coefficient for attenuation of radiation in the canopy, using a projected leaf area basis.

line 34
```
(DIM)     all-sided to projected leaf area ratio
```

The ratio between the all-sided area and the projected area for leaves. Projected area for this and all other uses in BIOME-BGC is the projected area of the leaf laid flat with its two longest dimensions parallel to the measurement surface, while all-sided area is the total leaf surface area.

line 35
```
(m2/kgC)  canopy average specific leaf area (projected area basis)
```

Projected area per unit of leaf carbon mass, averaged over the canopy.

line 36
```
(DIM)     ratio of shaded SLA:sunlit SLA
```

Ratio between specific leaf area for leaves in the shaded canopy fraction and specific leaf area for leaves in the sunlit canopy fraction.

line 37
```
(DIM)     fraction of leaf N in Rubisco
```

The fraction of total live leaf nitrogen occurring in the RuBisCO enzyme.

line 38
```
(m/s)     maximum stomatal conductance (projected area basis)
```

The maximum stomatal conductance to water vapor, expressed on a projected leaf area basis. This is the conductance under saturating light, low VPD, leaf water potential near 0.0, and moderate temperatures. Reciprocal of minimum stomatal resistance.

line 39
```
(m/s)     cuticular conductance (projected area basis)
```

The conductance of the leaf cuticle to water vapor, expressed on a projected area basis. Assumed constant under all environmental conditions.

line 40
```
(m/s)     boundary layer conductance (projected area basis)
```

Leaf boundary layer conductance to water vapor, expressed on a projected area basis. This is also refered to as the aerodynamic conductance. It is defined as the conductance for water vapor entering the atmosphere from a free water surface on the leaf surface (a raindrop on the leaf). It is assumed constant under all environmental conditions, although it is in reality a strong function of wind speed. A constant windspeed of 1 m/s is assumed in defining values of this parameter for various leaf morphologies.

lines 41 and 42

```
(MPa)   leaf water potential: start of conductance reduction
(MPa)   leaf water potential: complete conductance reduction
```

These two parameters set the endpoints for a linear control on stomatal conductance due to leaf water potential. The first parameter sets the leaf water potential at which conductance reduction begins, and the second parameter sets the water potential at which stomatal conductance is reduced to 0.0. In the range between these values, reduction of stomatal conductance is a linear function of leaf water potential.

lines 43 and 44

```
(Pa)    vapor pressure deficit: start of conductance reduction
(Pa)    vapor pressure deficit: complete conductance reduction
```

These two parameters set the endpoints for a linear control on stomatal conductance due to the water vapor pressure difference (VPD) between the interior of the leaf and the air adjacent to the leaf. The first parameter sets the VPD at which conductance reduction begins, and the second parameter sets the VPD at which stomatal conductance is reduced to 0.0. In the range between these values, reduction of stomatal conductance is a linear function of VPD.

OUTPUT FILES: OVERVIEW

There are two different styles of output produced by Biome-BGC. The first type includes the binary output files controlled through the initialization file with information from the OUTPUT_CONTROL, DAILY_OUTPUT, and ANNUAL_OUTPUT sections. This is the most flexible output mechanism, since the user can control exactly which model variables to include in the output files, and what level of averaging to perform. The second type of output is a very simple formatted text file that contains annual summary information for each year of the simulation. This text output file is produced for all non-spinup simulations, and it uses a fixed list of output variables.

These two output types are described in greater detail below.

BINARY OUTPUT FILES: DETAILS

These are not text files, and you will not be able to read them with a text editor or word processor. Instead, they are data files containing binary representations of the values of the output variables. Each value is written as a single-precision IEEE floating point binary number (using 4 bytes of storage per number). These values can be read directly using simple code written in C/C++, FORTRAN, BASIC, PASCAL, and other programming languages. They can also be read by many commercially available software packages, typically those that specialize in manipulating large multi-dimensional data sets. One good example of a commercially available package for display and analysis of binary data files is IDL, from Research Systems, Inc. (www.rsinc.com). The current cost is about $500 for a single PC/Mac license, and about $900 for a single workstation license (U.S. educational institution pricing). We will be making a package of simple plotting routines for IDL users available in the near future on the NTSG website. There are lots of other possibilities for reading these files, and you might want to consult with a computer programmer about the methods best suited for the kind of analysis you want to do.

In order to read the binary files, you need to know how output values are ordered in the file. Consider a single simulation over 100 years, for which the user requested three daily output variables and two annual ouput variables, and requested that in addition to the daily output, the three variables also be summarized as monthly averages and annual averages. Below is a corresponding example of the OUTPUT_CONTROL, DAILY_OUTPUT, and ANNUAL_OUTPUT sections of the ini file:

<EXAMPLE>

```
OUTPUT_CONTROL   (keyword - do not remove)
outputs/enf_test1     (text) prefix for output files
1   (flag)  1 = write daily output   0 = no daily output
```

```
1   (flag)  1 = monthly avg of daily variables  0 = no monthly avg
1   (flag)  1 = annual avg of daily variables   0 = no annual avg
1   (flag)  1 = write annual output  0 = no annual output
1   (flag)  for on-screen progress indicator

DAILY_OUTPUT     (keyword)
3      (int) number of daily variables to output
43     5 wf.soilw_trans
509    8 epv.proj_lai
620    10 summary.daily_npp

ANNUAL_OUTPUT    (keyword)
2               (int)   number of annual output variables
545     0 annual maximum projected LAI
636     1 vegetation C
```
<END EXAMPLE>

Four binary output files will be created for this simulation, and each file can be considered as a multi-dimensional array of values. The first dimension in each output file is the number of variables (3 in the case of daily variables, 2 in the case of annual variables). The last dimension is the number of years of simulation (100 in this case). Daily and monthly files have another dimension which is either the number of days per year (365) or the number of months per year (12). The values in parentheses beside the filenames below show the dimensions for each file in this example:

outputs/enf_test1.dayout	(3 x 365 x 100)
outputs/enf_test1.monavgout	(3 x 12 x 100)
outputs/enf_test1.annavgout	(3 x 100)
outputs/enf_test1.annout	(2 x 100)

Values are stored sequentially in the file as dictated by these dimensions. As an example, for the *.dayout file, the first three values in the file are the three daily output variables for the first day of the first year of the simulation. The next three values in the file will be the three daily output variables for the second day of the first year of simulation, etc.

The units for all variables are defined in the bgc_struct.h file.

ANNUAL TEXT OUTPUT: DETAILS

Because it can take some time to get used to the binary output formats, we also include a formatted text output file with annual summary information. For many applications this may be all the information required. In any case it allows a quick look at the results before proceeding with more detailed analyses. The name of the file is the user-supplied output prefix plus the following suffix "_ann.txt". The file is self-documenting, and an example is shown below:

<EXAMPLE>
```
Annual summary output from Biome-BGC version 4.1.2
ann PRCP = annual total precipitation (mm/yr)
ann Tavg = annual average air temperature (deg C)
max LAI = annual maximum value of projected leaf area index (m2/m2)
ann ET = annual total evapotranspiration (mm/yr)
ann OF = annual total outflow (mm/yr)
ann NPP = annual total net primary production (gC/m2/yr)
ann NPB = annual total net biome production (gC/m2/yr)

  year  ann PRCP  ann Tavg   max LAI    ann ET    ann OF   ann NPP   ann NBP
  1996     869.4       8.4       2.0     359.7     483.0     385.7     -20.8
  1997     475.6       8.4       2.0     347.7     182.8     405.5      11.1
  1998     727.5       8.6       2.0     380.3     310.7     405.7       7.1
  1999     524.3       8.1       2.0     332.7     199.9     376.5       0.0
  2000     869.4       8.4       2.0     359.7     483.0     385.6     -20.8
```
<END EXAMPLE>

With just a little bit of simple programming, the formatting of this file can be changed to include whatever variables might be of interest, if the binary output options are not feasible.

RUNNING THE EXAMPLES

This release of the Biome-BGC code comes with a simple set of example input and output files that illustrate the recommended procedure for running a simulation at a new site. The example simulation is for a site in Missoula, MT, using 44 years of meteorological data as processed by the MTCLIM program. The vegetation type is assumed to be a generic evergreen needleleaf forest. Instructions are given below for both PC and UNIX users.

The first step is to run a spinup simulation that will start with very low initial levels of soil carbon and nitrogen and loop through the 44-year meteorological data file many times until the total carbon levels stabilize. This simulation does not use a restart input file, but it produces a restart output file for use in the next step. The second step is to run a normal simulation, which runs for one pass through the 44-year met data file.

To run the spinup simulation:
(UNIX) From the unixbgc412 directory, enter the following text at the command line and press enter:

```
bgc412 ini/enf_test1_spinup.ini
```

(PC – except Windows 2000) In the pcbgc412 directory, double-click the icon for bgc412.bat (NOT the bgc412.exe icon...). This opens a window that contains some text asking you to enter the name of an initialization file. Enter the following line of text and press enter

```
ini/enf_test1_spinup.ini
```

(**Windows 2000**): The supplied *.bat files do not work in Windows 2000. I recommend you use the Command Prompt utility, from Start, Programs, Accessories. This is essentially a DOS prompt, and you can navigate to the appropriate directory and enter the following command:

```
bgc412.exe ini/enf_test1_spinup.ini
```

(UNIX and PC) At this point the program is launched and you should see a progression of numbers scrolling down the left side of the screen or window. These record the number of years of simulation that have been completed. This particular spinup run should take 2112 years to reach a steady state, after which the following message will be displayed and the program will exit:

```
SPINUP: residual trend  = -0.001184
SPINUP: number of years = 2112
```

This indicates that the steady state requirement was met after 2112 years, and that there is still a slight trend in the total system carbon, in this case a change of -0.001184 kg C m^{-2} yr^{-1}.

The spinup run does not produce any output files except a restart file that is stored in the restart subdirectory. This is used in the next step to initialize a normal simulation that will produce a range of output files.

To run the 44-year simulation:
(UNIX) Again from the unixbgc411 directory, enter the following line of text and press enter:

```
bgc412 ini/enf_test1.ini
```

(PC – except Windows 2000) In the pcbgc412 directory, double-click the icon for bgc412.bat (NOT the bgc412.exe icon...). This opens a window that contains some text asking you to enter the name of an initialization file. Enter the following line of text and press enter

```
ini/enf_test1.ini
```

(**Windows 2000**): The supplied *.bat files do not work in Windows 2000. I recommend you use the Command Prompt utility, from Start, Programs, Accessories. This is essentially a DOS prompt, and you can navigate to the appropriate directory and enter the following command:

```
bgc412.exe ini/enf_test1.ini
```

(UNIX and PC) At this point the program is launched and you should again see a progression of numbers scrolling down the left side of the screen or window. This simulation is set to run once through the 44 years of meteorological data. In this case, the scrolling numbers start with 1950 and end with 1993, since 1950 was given as the starting year for the met data in the ini file.
This simulation will produce a complete set of output files, in the outputs directory, all having filenames beginning with enf_test1. Consult the enf_test1.ini file to see which variables are being output for the binary files. The simple

annual text file (enf_test1_ann.txt) is also created, and it is reproduced below in its entirety so that you can check your results.

Contents of enf_test1_ann.txt

```
Annual summary output from Biome-BGC version 4.1.2
ann PRCP = annual total precipitation (mm/yr)
ann Tavg = annual average air temperature (deg C)
max LAI = annual maximum value of projected leaf area index (m2/m2)
ann ET = annual total evapotranspiration (mm/yr)
ann OF = annual total outflow (mm/yr)
ann NPP = annual total net primary production (gC/m2/yr)
ann NPB = annual total net biome production (gC/m2/yr)
```

year	ann PRCP	ann Tavg	max LAI	ann ET	ann OF	ann NPP	ann NBP
1950	376.0	5.9	1.4	338.8	0.0	160.1	-6.9
1951	390.0	5.8	1.5	372.3	0.0	211.4	20.0
1952	226.0	6.8	1.6	294.1	0.0	95.6	-76.9
1953	302.0	8.2	1.5	296.5	0.0	119.4	-42.6
1954	356.0	6.8	1.5	361.9	0.0	220.1	54.8
1955	415.0	5.2	1.5	339.7	0.0	138.3	-21.5
1956	409.0	6.8	1.6	461.6	0.0	280.3	60.2
1957	318.0	6.9	1.7	317.3	0.0	121.6	-42.6
1958	439.0	8.2	1.7	412.9	0.0	183.4	9.0
1959	422.0	6.4	1.7	425.4	0.0	234.4	28.1
1960	261.0	6.1	1.7	287.7	0.0	85.5	-83.0
1961	366.0	6.8	1.6	338.1	0.0	145.7	-24.9
1962	315.0	6.0	1.6	344.4	0.0	154.1	-19.8
1963	388.0	6.6	1.5	392.9	0.0	256.6	80.0
1964	399.0	5.0	1.6	347.9	0.0	177.9	31.7
1965	374.0	6.0	1.7	426.4	0.0	274.3	86.3
1966	291.0	7.8	1.8	293.2	0.0	108.5	-61.0
1967	318.0	7.9	1.7	294.3	0.0	95.9	-68.1
1968	324.0	7.1	1.6	341.0	0.0	213.0	42.5
1969	341.0	6.0	1.6	348.7	0.0	154.3	-20.3
1970	394.0	6.7	1.7	380.4	0.0	197.2	9.4
1971	343.0	7.2	1.7	327.7	0.0	123.5	-53.6
1972	351.0	6.6	1.6	365.2	0.0	223.2	23.5
1973	233.0	7.6	1.6	182.3	0.0	29.6	-129.0
1974	283.0	7.6	1.5	358.9	0.0	210.6	10.1
1975	476.0	6.0	1.5	397.2	0.0	244.6	58.5
1976	241.0	7.3	1.6	330.0	0.0	141.4	-31.2
1977	338.0	6.9	1.5	253.3	0.0	106.4	-42.8
1978	311.0	6.2	1.5	364.3	0.0	227.0	38.5
1979	273.0	6.9	1.5	281.5	0.0	99.2	-71.8
1980	499.0	7.1	1.7	490.0	0.0	314.3	118.5
1981	450.0	7.6	1.8	418.4	0.0	236.7	72.1
1982	400.0	6.4	1.9	438.2	0.0	226.3	20.9
1983	433.0	6.8	2.0	406.7	0.0	244.5	93.6
1984	345.0	6.9	2.0	364.9	0.0	158.5	-13.6
1985	328.0	5.8	1.9	328.5	0.0	146.9	-23.7
1986	432.0	7.4	1.9	422.6	0.0	247.9	50.5
1987	265.0	7.6	1.9	292.9	0.0	94.9	-68.0
1988	288.0	7.8	1.8	281.4	0.0	91.8	-70.4
1989	363.0	6.8	1.6	368.8	0.0	199.7	30.9
1990	350.0	7.6	1.5	330.4	0.0	174.7	11.1
1991	308.0	7.3	1.5	298.4	0.0	105.8	-57.5
1992	300.0	8.0	1.5	314.0	0.0	177.5	10.0

```
  1993      358.0        6.1        1.5      361.0        0.0      195.2       33.9
```

Looking at the example binary output in IDL
For users with access to the IDL software, there is an example IDL procedure in the main directory that will read and display the output stored in the enf_test1.dayout file. For UNIX users, start IDL in the unixbgc411 directory, then enter the command example_plot on the command line. For PC users, after starting IDL it may be necessary to change the working directory (under File, preferences menu) to the pcbgc411 directory. Then open the example_plot.pro file, compile, and execute. In both cases, a window will be created that will display three different plots for each of the 23 output variables requested in the enf_test1.ini file. You scroll through these plots by pressing any key. The first plot for each variable shows all the daily data for 44 years as a time series. The second plot for each variable shows one average year of daily data (the average of the 44 individual years). The third plot shows the average value for each year as a 44-year timeseries. This example IDL procedure is included to give you some ideas on how to read and display the binary data.

Compiling Biome-BGC source code

If you are working on an operating system other than IBM AIX or Windows 9x/2000, or if you want to make any changes to the code, you will need to recompile to get a new executable. The process is different on PC and UNIX systems, so two sets of instructions are given below.

UNIX compilation
This code release comes with a set of makefiles that need some alteration before you use them. The main makefile is in the subdirectory src, and this should not have to be changed. There are two other makefiles, one in the src/bgclib_v4.1.2 subdirectory and another in the src/pointbgc_v4.1.2 subdirectory that both need to be modified as follows:
1) Change the ROOTDIR variable so that it reflects the pathname for the src subdirectory. For example, if the src subdirectory is located at /u1/yourname/bgc/src, then you would replace the current text (/ntsg/peter/pointbgc_v4.1.2/src) with the new path.
2) Change the name of the C compiler to a valid name for your system. Do this by editing the line in both subdirectory makefiles that begins with 'CC=', replacing the text 'xlC' with the name of your C/C++ compiler. Some possible examples for the compiler name are 'cc' and 'gcc'.
3) From the src directory, issue the protect_user_off command, which removes write protections.
4) From the src directory, issue the command make. This executes the makefiles in both subdirectories and if successful produces a new executable called pointbgc in the parent directory to the src directory. The name of the new executable is different from the original executable supplied with this release so that the original is not overwritten.
5) In order to protect the code from accidental changes, issue the protect_on command from the src directory after the new executable has been created.

PC compilation
For the PC model release, all of the source code is located in the src directory. The executable provided with this release was generated by compiling the source code with the Microsoft Visual C/C++ compiler (version 6.0). All the project files used by MS Visual C/C++ are located in the subdirectory src/bgc412, and users with the Visual C/C++ software installed can open the bgc412.dsw file to access this project. Selecting "rebuild all" from the "Build" menu will recompile the code, putting a new executable called bgc412.exe in the src/bgc412/Release directory. There are many other compilers to choose from - contact your local PC system manager for more information.

Code History

The Biome-BGC ecosystem process model has a long heritage, and many people have contributed extensively to its development over many years. The following is a short synopsis of some of the most prominent code versions preceding the current version 4.1.2 benchmark code for public release.

- Biome-BGC version 4.1.1, Peter E. Thornton, 2001 (C/C++)
- Biome-BGC version 4.1, Peter E. Thornton, 2000 (C/C++)
- Biome-BGC version 2.0 (CRB-BGC), Peter E. Thornton, 1995 (C/C++)
- Biome-BGC version 1.37, E. Raymond Hunt, Jr., 1993 (Pascal)

- Forest-BGC, Joseph C. Coughlan, 1986 (Pascal)
- DAYTRANS-PSN, Steven W. Running, 1981
- DAYTRANS, Steven W. Running, 1975

Others who have contributed to the code in various ways and at various stages include: Galina Churkina, Tom Gower, Kathy Hibbard, Bob Keane, John Kimball, Lars Pierce, Joseph White, and Michael White.

Questions or comments?

If you have problems with the code, or if you have suggestions on how it could be improved, we'd love to hear from you. Also, please send us copies of any manuscripts or reports that include the use of Biome-BGC, especially if you have compared Biome-BGC estimates to measurements. This information helps us to better understand model performance, and improve it where we can. For questions on how to parameterize or run the code, please read everything in this file first.

Contact:
Peter E. Thornton
E-mail: thornton@ucar.edu