****
****

GLOPEM DOCUMENTATION
Code and documentation written by Jen Small
Last update 8/18/2003

****

INPUT. FILES

All input data must follow a strict file naming convention- The file names given below must be followed exactly.

The same set of input data is required regardless of the temporal or spatial scale of the model run.

The input data format is raw binary 32 bit real unless (IEEE float) otherwise specified, with no headers.

The units for each input file are also given below.

Static inputs

****

The following variables are constant through time. 41.

biomass.flt	above ground biomass, kg/mA2
lati tude. fl t	latitude, degrees
whc, flt	soil water holding capacity (field capacity), mm
topt. flt	optimal air temperature for photosynthesis, Kelvin
mmm.clim_tair'.flt	monthly climatological mean air temperature, Kelvin mmm represents the month: jan,feb,mar,apr,may,jun, jul,aug,sep,oct,nov,dec (must be lowercase)
prev_swc.flt	initial soil water content, mm. The soil water content for the first time step of the model run. This can be set to half of field capacity, total field capacity, etc depending upon what information is available. This file must have the same path as the soil water content output files.

Time-specific inputs

****

All inputs specific to a certain time step will have file names that begin with a time identifier. The next part of the filename will identify which input variable the file contains. Lastly all files have a .flt extension (to denote a flat binary file). All files must be input at the same temporal frequency. Thus the structure of each file name is

    time_ident~fier.input_ide~tifier.flt
A

Note that each portion of the filename is sepirated by a period.

time identifier
****
****
The time identifier is:

    yYyy.ddd-eee

where yYyy is the 4-digit year, ddd is the first Julian day of the compositing period, and eee is the last Julian day of the compositing period.

The Julian days must each contain three digits, ie leading zeroes are required.

For example, for an input from 2001 that covers Julian days 5 to 10 the time identifier .is

    2001.005-010

For daily inputs put ddd=eee. For example for Julian day 12 of year 1995 the time identifier is

    1995.012-012

At present the finest time step allowed in GLOPEM is daily.

input   identifier
****

Following is a list of all GLOPEM time-specific inputs and their units.

tair

Canopy air temperature, Kelvin.
At present this represents an approximate maximum daily canopy air temperature as it uses AVHRR thermal observations corrected
t-.?~~-?~rpas~ t._~~ _o~-2: 3.°,Efi.\...

par (optional) ***

Photosynthetically active radiation, kiloJoules per square meter per (# of days in ~he compositing period) .
This input is optional: it may be calculated within the soil moisture routine by setting the appropriate flag in the parameter file (see Parameter File section). If PAR is to be calculated in the soil moisture routine then fractional sunshine (see sunfr input identifier) must be given as an input.

fpar
****
Fraction of photosynthetically active radiation absorbed by the canopy. A unitless fraction ranging from 0 to 1.

Vp
**
surface. vapor pressure, KPa.
Mean dally surface vapor pressure over the compositing period.

precip

Precipitation, mm.
This is the total precipitation in mm summed over all days in the compositing period.

sunfr (opt-ior).al)
****
Mean fraction of sunshine for all daylight hours in the compositing period, unitless ratio 0-1.
This input is optional; this input is not needed if BOTH photosynthetically radiation (see par input identifier) and net radiation (see par input identifier) and net radiation identifier (see rn input identifier) are given as inputs. See Parameter File section for details on specifying inputs

dtr
***
Diurnal temperature range, Kelvin

Mean diurnal temperature range over the compositing period.

rn (optional)
****
Net radiation, watts per square meter.
The mean net radiative flux at the surface over the compositing period. This input is optional; this input may be calculated in the soil moisture routine provided that fractional sunshine (see sunfr input identif.ier) is given as an input. See Parameter File
section for details on speci~ying inputs.

As a final example suppose that we want to name the input files from 1992, with a compositing period from day 1 to 15. The filenames for these inputs would be:

```
1992.001-015.tair.flt
1992.001-015.par.flt
1992.001-015.fpar.flt
1992.001-015.vp.flt
1992.001-015.precip.flt
1992.001-015.sunfr.flt
1992.001-015.dtr.flt
1992.001-015.rn.flt
```

PARAMETER FILE

The parameter file MUST be named "parameter. file" , and must be located in the same directory as the executable code. This is a plain text (ASCII) file with the following format:

line #	variable name(s)	description of variable(s)
1	total-pixels	the total number of pixels (number of columns * number of rows) in each input image
2	year	the four digit year
3	radflag	Radiation calculation flag. This flag specifies which radiation parameters are input to the model and which are calculated within the model. Choices are: radflag=O Input sunlight fraction, calculate PAR and net radiation. radflag=l Input PAR and sunlight fraction, calculate net radiation. radflag=2 Input PAR and net radiation.
4	flag	the flag/missing dat~for all input files.
5	co2[12]	Monthly atmospheric CO2 partial pressure, Pa. The 12 monthly values must be entered on a single line seperated by white space.
6	num comps	Number of compositing periods for the model run.
7+	Jfirstjday lastjday	The first and last Julian days for each input file for the model run.

'It

Here is a sample parameter file:

10848672 1999
2
-999.0 37.30
37.38
37,45 37.61 37.59 37.53 37.42 37.18 36.94 36.99 37.15 37.29

```
36
001 010
011 020
021  031
032  041
042  051
052  059
060  069
070  079
080  090
091 100
101 110
111 120 121
13 0 131 140
141  151
152  161
162  171
172  181
182  191
192  201
202  212
213 222
223    232
233 243
244  253
254  263
264  273
274  283
284  293
294  304
305  314
315  324
325  334
335 344
345  354
355 365
```

This file specifies that each input year is file has 10848672 total pixels. The will
1999. PAR and net radiation data value is be input. The flag/missing
-999. The atmospheric months Jan-Dec are: CO2 partial pressures *for* the twelve
37.30 37.38 37.45 37.61 37.59 37.53 37.42 37.18 36.94 36.99 37.15 37.29 The remaining
lines contain the first and last Julian day *for* each
of the compositing periods *for* the year, including leading zeroes.

OUTPUT FILES

All GLOPEM output files are raw binary files with no headers. Unless otherwise noted, output files are 32 bit real (IEEE floating point) with the flag/missing data value specified in the parameter file.

Following is a list of GLOPEM output files and their units. All output files have the same time identifier as their corresponding inputs, with two exceptions:

pet
***
Potential evapotranspiration, mm/month.
This is output as a mean monthly value for the year. The filename is yyyy.pet.flt.

lhi

Local heat index, unitless.
This is output as a mean monthly value for the year. The filename" is yyyy.lhi.flt.

All of the following files have the time identifier yyyy.ddd-eee described in the INPUT FILES section.

potlue
*~****
Potential light use efficiency, grams Carbon per megaJoule.
Maximum light use efficiency assuming optimal environmental conditions.

stress_tair
****
Air temperature stress factor, 0-100%.
The percentage of potential LUE (epot) remaining after accounting for the deviation of canopy air temperature (tair) from optimal air temperature (topt). (100=no stress and O=full stress)
This file is 8 bit unsigned with flag value 101.

vpd
***
Vapor pressure deficit, KPa.

stress_vpd
****
Vapor pressure deficit stress factor, 0-100%.
The percentage of potential LUE (epot) remaining after accounting for the vapor pressure deficit. (100=no stress and O=full stress) This file is 8 bit unsigned with flag value 101.

stress-par
****
Par saturation stress factor, 0-100%.
The percentage of potential LUE (epot) remaining after accounting for light saturation. (100=no stress and O=full stress)
This file is 8 bit unsigned with flag value 101.

swc
***
Soil water content, mm.

stress_swc
****
Soil moisture stress factor, 0-100%.

The percentage of potential LUE (epot) remaining after accounting for available soil moisture. (100=no stress and O=full stress) This file is 8 bit unsigned with flag value 101.

par (optional) ***

Photosynthetically active radiation, kiloJoules per square meter per (# days in compositing period) This variable is output by setting a flag in the parameter file (see Parameter File), otherwise it is given as an input.

apar
****
Absorbed photosynthetically active radiation, kiloJoules per square meter per (# days in compositing period) .
par times fpar.

lue
***

Light use efficiency (LUE) , grams Carbon per megaJoule. Represents actual LUE after all stress factors have been applied.

gpp

Gross primary production, grams Carbon per square meter per (# days. in compositing period) .
The amount of carbon sequestered before respiration losses are factored in.

autoresp
****
Plant (autotrophic) respiration, grams carbon per square meter per (# days in compositing period) . The amount of gross productivity that is lost to respiration.

npp
***
Net primary production, grams carbon per square meter per (# .~ays in compositing period) . -~
The net carbon sequestered after subtracting respiration losses.

~ ^^

RUNNING GLOPEM

****

Platforms/compiling
****
GLOPEM has been compiled and run on a DEC Alpha running Digital UNIX V4.0F and on a.Sun Ultra-10 running SunOS Release 5.7.

To compile GLOPEM type "make -f Make.glopem" at the UNIX prompt.

Along with the make file Make.glopem, the following files must be in the directory:

glopem.h (header file- contains paths to all input and output data)

allocate_binary.c (memory allocation function)
apar.c (apar function)
autotrophic_respiration.c (plant respiration function) get_month.c (determines month from Julian day) glopem_main.c (main program, calls all functions)
gpp.c (gpp function)
lue.c (light use efficiency function)
npp.c (npp function)
pet.c (pet and lhi function)
potential_lue.c (potential lue function)
read_binary.c (binary file read function) read-parameter_file.c (parameter file read function) stress-par.c (light saturation stress function) stress_swc.c (net radiation/soil moisture stress function) stress_tair.c (canopy air temperature stress function) stress_vpd.c (vapor pressure deficit stress function) write_binary.c (binary file write function)

Information on algorithms, inputs, and outputs for each function is below in the FUNCTIONS section.

Specifying absolute paths to inputs and outputs in the header file
****

The absolute file system path names to all input and output files must be specified in the header file "qlopem.h". This allows the inputs and outputs to be put into different directories.
For each input and output, there is a corresponding variable in the header file containing the path to the input or output. For example, for the npp the header file contains a line

```
#define npppath            "pathname"
```

The path must be enclosed in quotes, and the final forward slash must be included. For example, if the npp files should be written to a directory cal.led "/data/geogdoh/npp" then the line should be

```
#define npppath            "/data/geogdoh/npp/"
```

For a complete description of all variables defined in the header file see the HEADER FILE section.

The path variables for each input and output are:

input/output   variable   path   variable
********************   *************
biomass   biomasspath latitudepath
latitude   whcpath toptpath
water holding capacity   cltairpath
optimal air temperature climatological mean swcpath tairpath
air temperature initial soil water content   parpath
canopy air temperature photosynthetically   fparpath
active radiation fraction of par absorbed by
canopy

surface vapor pressure precipitation. vppath
sunshine fraction precippath
diurnal temperature range sunfrpath
net radiation dtrpath
potential evapotranspiration local rnpath
heat index petpath lhipath
potential light use efficiency potluepath
. air temperature stress factor vapor stress_tairpath
pressure deficit vpdpath
vapor pressure deficit stress par stress_vpdpath
saturation stress factor soil water factor stress-parpath
content swcpath
soil moisture stress factor available stress_swcpath
par (apar) aparpath
light use efficiency luepath
gross primary production autotrophic gpppath
respiration autoresppath npppath
net primary production

Required memory (RAM)

Memory for inputs is allocated as needed within each individual subroutine and then freed before the next subroutine is called. The subroutine that requires the most RAM is "stress_swc.c", the soil moisture stress function. This function allocates memory for a maximum of 11 floating point images (depending on the value of the parameter file radflag) plus one 8 bit image. The amount of required memory is

(number of pixels)*(45) bytes

For example if the input images have 5004 rows and 2168 columns, the minimum RAM required is

5004*2168*45 = 4.8819 * 10A8 bytes

or about 489 MB.

~

Required disk Space
****
Let N be the number of time steps in the year. Let P equal the total number of pixels (rows*columns) in one input image.

There are a maximum (depending on the parameter file radflag) of 15 floating point images and four 8 bit images for each time step. This requires N*P*68 bytes of disk space.

Also there are'7 floating point images that are static for each year. This requires P*28 bytes of disk space.

Finally there are the 12 climatological mean air temperatures which require P*48 bytes of disk space.

The total disk space required for the one year run is then

N*P*68 + P*28 + P*48 = P*(N*68 + 76) bytes.

For example, if images have 3000 rows and 2000 columns and there are 365 time steps (daily) then the total disk space required is:

3000*2000*(365*68 + 76) = 1.4938 * 10A11 bytes,

or about 150 GB.

^^

HEADER FILE

^^

This section describes all global variables defined in the header file. Except for the path variables (and the parameter file name if desired), these should not be set by the user. They are either calculated or read in. --.~ -- .

preprocessor variables
****
Preprocessor variables are preceded by #define.

PARMFILE is the name of the parameter file.
RAD is the degrees to radians conversion factor.
xxxxpath variables are full path names for each input and output.
See RUNNING GLOPEM section for a complete list of path variables.

global variables
****
filename holds a filename of up to 99 characters
total-pixels -is the total number of pixels (rows*cols) in each file
year is the four digit year
num_comps is the number of compositing periods in the year
composite is the loop counter for compositing periods
~ firstjday[366] is an array of the first Julian day for each compositing period
lastjday[366] is an array of the last Julian day for each compositing period
timeid holds a single time identifier
flagval is the flag value for the floating point images
co2[12] is an array of the twelve monthly atmospheric CO2 partial pressures atm_co2 holds the atmospheric CO2 partial pressure for a single month month holds the three letter abbreviation for a single month
radflag is the flag for the radiation calculation (see Parameter File section)

^^

FUNCTIONS
****

This section contains a brief description of each of the functions in GLOPEM.

allocate_binary.c
****

Allocates memory for a binary image of the specified data type and number of pixels. All data types are supported but at present GLOPEM uses only float and unsigned char data types.

arguments: numpx dtype imptr

numpx = number of pixels (rows*cols) to allocate for the dtype = the image datatype for the image. One of:
double = double precision floating point float = single precision floating point
int = signed integer
uint = unsigned integer
sint = signed short integer
usint = unsigned short integer
char = signed very short integer
uchar = unsigned very short integer
imptr = address of memory location (ie double pointer) where memory will be allocated, cast to type void

The function allocates numpx pixels of dtype starting at memory location *imptr.

inputs: none

outputs: none

apar.c
******

Calculates the amount of photosynthetically active radiation absorbed by the canopy. The calculation is

~ apar = fpar*par

where fpar is the fraction of photosynthetically active radiation absorbed by the canopy and par is the photosynthetically active radiation incident at top of canopy.

arguments: timeid

timeid = the time identifier (see INPUT FILES)

inputs: par, fpar

par = photosynthetically active radiation incident on canopy fpar = fraction of par absorbed by the canopy

outputs: apar

apar = amount of photosynthetically active radiation
absorbed by the canopy

autotrophic_respiration.c
****

Calculates the fraction of gross productivity remaining after carbon losses to plant respiration.

There are three components: maintenance respiration, growth respiration, and a correction factor that adjusts for the exponential response of respiration to canopy air temperature. The fractional respiration is calculated as the product of these three components.

maintenance respiration: Rm = 1 - (0.4/0.75)*(W/(W~50)
where W is biomass in megagrams per hectare.
(reference: Prince and Goward 1995; Hunt 1994)

growth respiration: Rg = 0.75
(reference: Prince and Goward 1995; Hunt 1994)

temperature correction factor: Rt = exp(0.5*((clim_tair-tair)/25)), where
clim_tair is the climatological mean air temperature
and tair is the canopy air temperature.
(reference: Goetz et al 1999 - note that Rm and Rg are specified
improperly in equation 7a)

fractional respiration: Rf = Rm*Rg*Rt

Rf is the fraction of GPP remaining after respiration, therefore an increase in Rf implies a decrease in carbon losses to respiration.

arguments: timeid month

timeid = the time identifier (see INPUT FILES)
month = the three-letter abbreviation for the month (from get_month)

inputs: biomass, climatological mean air temperature,
canopy air temperature

» output: fractional respiration

get_month.c
****

Determines the Julian day from the month and year, and assigns the atmospheric CO~ partial pressure (read from the parameter file) for that month.

arguments: jday year

jday = The julian day for the midpoint of the compositing period.
If the compositing period has an even number of days
then the earlier day is used as the midpoint.

year = the four digit year.

inputs: co2[12]

co2[12] = the array of monthly atmospheric CO2 partial pressures
(from the parameter file)

outputs: month atm_co2

month = the three-letter abbreviation for the month containing jday atm_co2 =
the atmospheric CO2 partial pressure for month

glopem_main.c
****

This is tqe main GLOPEM module. It calls all of the functions described in this section for each of the time steps in the year (except the read-parameter_file and pet functions which are called only once) .

The structure of the main program is:

Read the parameter file (call read-parameter_file)

Calculate potential evapotranspiration and local heat index for
the year (call pet)

for each compositing period

set the time identifier

calculate the number of days in the compositing period

calculate the month for the middle day of the compositing period
period and set the atmospheric CO2 partial pressure for
the month (call get month)

calculate potential light use efficiency (call potential_lue)

calculate air temperature stress (call stress_tair)

~

calculate vapor pressure deficit (vpd) and vpd stress
(call stress_vpd)

calculate soil moisture stress and soil water content, and
possibly photosynthetically active radiation (call stress swc)

calculate light saturation stress (call stress-par)

calculate a~sorbed photosynthetically active radiation (call apar)

calculate light use efficiency (call lue)

calculate gross primary production (call gpp)

calculate fractional autotrophic respiration

(call autotrophic_respiration)

calculate net primary production and absolute autotrophic respiration (call npp)

end for

arguments: none

All inputs and outputs are to the individual functions called.

gpp.c

Calculates gross primary production (GPP) , the amount of taken up carbon factored by the vegetation before respiration losses are GPP is defined as in. apar times light use efficiency.
(reference - Goetz et al 1999)

arguments: timeid

timeid = the time identifier (see INPUT FILES)

inputs: apar, lue

apar = absorbed photosynthetically active radiation lue = light use efficiency

outputs: gpp

gpp = gross primary production

lue.c

Calculates light use efficiency (lue, also called carbon yield of apar) The lue is defined as the product of potential light use efficiency and the four stress factors - canopy air temperature stress, vapor pressure deficit stress, soil moisture stress, and light saturation stress.
~ Note that the stress factors are converted from percent (0-100) to ratios (0-1) before the calculation of lue.
(reference - Prince and Goward 1995, Goetz et al 1999 - although neither of these mentions the light saturation stress)

arguments: timeid

timeid = th~ time identifier (see INPUT FILES)

inputs: stress~vpd, stress-par, stress_tair, stress_swc, potlue

stress_vpd = vapor pressure deficit stress
stress-par = light saturation stress stress_tair = air temperature stress stress_swc ~ soil moisture stress

potlue = potential light use efficiency

outputs: lue

lue = light use efficiency

npp.c

*****

Calculates net primary production (NPP), the net amount of carbon sequestered by the vegetation after respiration losses. NPP is defined as:

NPP = GPP*Rf

G~P = gross primary production
Rf = fractional autotrophic respiration
(reference: Prince and Goward 1995)

This function also converts fractional respiration to absolute (ie carbon amount lost in grams per square meter) respiration (Ra):

Ra = GPP*(O.75*(1-Rf)+O.25)

reference: Hunt 1994 - Ra is one minus the sum of Rg (eqn 3a) and Rm (eqn 3b)

note: this has not been modified since addition of the temperature correction factor Rt to the calculation of Rf

arguments: timeid

timeid = the time identifier

inputs: gpp Rf

gpp = gross primary production
Rf = fractional autotrophic respiration

outputs: npp Ra

npp = net primary production
Ra = absolute autotrophic respiration (grams carbon per sq meter)

pet.c

*****

Calculates potential evapotranspiration (pet) and local heat index (lhi).

lhi is first calculated as:

lhi = sum((clim_tair*O.2)A1.514)

clim_tair = the climatological mean air temperature in Celcius. The sum is taken over the twelve months of the year.
(reference: Bugmann and Cramer 1998)

pet is then calculated as:

pet = 0.000001*(492390.0 + 17920.0*lhi - 77.1*lhiA2 + 0.675*lhiA3)

(reference: Bugmann and Cramer 1998; parameters are from MK Cao)

These are the only variables that are calculated on an annual basis, the others are all calculated for each input time step.

arguments: none

inputs: clim_tair year

clim_tair = climatological mean air temperature for each month of the year
year = the four digit year (used only to name pet and lhi files)

outputs: pet lhi

pet = potential evapotranspiration lhi =
local heat index

potential_lue.c
****

Calculates potential light use efficiency (potlue).

First leaf CO2 concentration is estimated as

leaf_co2 = 0.7*atm~co2

where atm_coi is the atmospheric CO2 partial pressure.

Then the proportion of C3 (pct_C3) and C4 (pct_C4) vegetation is calculated as a function of biomass (W) and climatological mean air temperature (clim tair):

if W > 2 then

pct_C3 = 1
pct_C4 = 0

otherwise

pct_C4 =,1/(1 + exp(-0.5*(clim_tair-295))
pct_C3 = 1 - pct_C4

(reference: Prince and Goward 1995)

The carbon yield of C3 vegetation (Egc3) is then:

Egc3 = 55.2*alpha

where

alpha = 0.08*((leaf_co2-gamma)/(leaf_co2+(2*gamma))) gamma
= 20900.0/(5200.0*0.57A((tair-293)/10)))
tair = canopy air temperature

(references: Prince and Goward 1995, Collatz et al 1991)

The carbon yield of C4 vegetation (Egc4) is a constant:

Egc4 = 2.76

(reference: Prince and Goward 1995)

The 24 hour mean temperature in celcius (tcelcius) is then estimated as a function of canopy air temperature (tair) and diurnal temperature range (dtr) - this value is constrained to [0,40]

tcelsius = (tair - 0.5*dtr) - 273.15

This assumes that the canopy tair recovery is approximately a maximum tair for the day, and that tmean = tmax - 0.5*dtr.

The Q10 effect on carbon yield (q10 adj) is approximated as

q10_adj = 2.5/(1 + 4*4A(-0.05*tcelcius))

(reference: unpublished, MK Cao)

Finally, the potential light use efficiency (potlue) is

potlue = ((Egc4*pct_C4)+(Egc3*pct_C3))*q10_adj

(reference: Prince and Goward 1995, but note that this was before the Q10 component was added)

arguments: timeid

timeid = the time identifier

inputs: clim_tair tair dtr biomass atm_co2

clim_tair =,climatological mean air temperature tair =
canopy air temperature
dtr = diurnal temperature range
biomass = above ground biomass
atm_co2 = atmospheric CO2 partial pressure

outputs: potlue

potlue = potential light use efficiency, ie the light use efficiency with perfect environmental conditions

read_binary.c
****

Reads a binary file into allocated memory. The function reads numpx pixels of data type dtype from file fname into memory starting at address *imptr.

arguments: numpx fname dtype imptr

numpx = number of pixels (rows*cols) in the image fname = filename for the binary file
dtype = data type (see allocate_binary. c)
imptr = address of the pointer (ie double pointer) to the memory allocated for the image, cast to type void

inputs: none

outputs: none

read-parameter_file.c
****

Reads the parameter file (see Parameter File section for structure) .

arguments: none

inputs: parameter file 'parameter. file'

outputs: total-pixels year radflag flagval co2[12] num_comps firstjday[num_comps] lastjday[num_comps]

total-pixels = number of pixels (rows*cols) in each input image year = four digit year for the run
radflag = radiation calculation flag (see Parameter File section) flagval = flag/missing data value for all floating point inputs co2[12] = array of monthly CO2 partial pressures
num_comps = the number of compositing periods in the year firstjday[num_comps] = array of first Julian days for compositing periods
lastjday[num_comps] = array of last Julian days for compositing periods

~

stress-par.c
****

Calculates light saturation stress. The output is an 8 bit unsigned binary image representing the percentage (0-100) of GPP remaining after reductions for light saturation, ie the higher the light saturation stress value the lower the reduction in GPP.
The flag value is 101.

The saturation point increases with the leaf area index (LAI) of the canopy. This function uses fpar as a proxy for LAI. The input par is first converted to daily par in Joules per square centimeter per day. The stress factor is then calculated as (rounded to nearest integer) :

```
   stress-par = 100*(1 - l/(l+exp(exponent)))
```

where

```
   exponent = (8*fpar) - (4* (daily-par-1000)/1000)
```

reference: unpublished, algorithm from MK Cao

arguments: timeid compdays

```
   timeid = the time identifier
   compdays = the number of days in the compositing period
```

inputs: fpar par

```
   fpar = fraction of photosynthetically active radiation
          absorbed by the canopy

   par = photosynthetically active radiation incident on
         the canopy
```

outputs: stress-par

```
   stress-par = the amount of gross primary production remaining after
                accounting for reductions due to light saturation.
```

stress_swc.c
****

Calculates soil moisture stress. Stress is output as an 8 bit unsigned binary image representing the percentage (0-100) of GPP remaining after reductions for soil moisture stress, ie the higher the soil moisture stress value the ~ lower the reduction in GPP.
The flag value is 101.

Soil moisture stress is calculated as a function of the ratio of actual evapotranspiration (AET) to water demand:

```
   stress swc = 100*(R*(1 - 0.2*R))
```

where

```
   R = AET/water demand
```

water demand
""""""""""""

The water demand is a function of net radiation (rn), latent heat (lh), the rate of increase in saturation vapor pressure with air temperature (delta vpsat), and the psychrometer constant (psy):

```
water_demand = 3600.0*(delta_vpsat/(delta_vpsat+psy))*(rnet/lh)
```

The variables delta_vpsat, Ih, and psy are all functions of canopy air temperature tair (converted to Celcius) :

```
delta_vpsat= 2503000.0*exp(17.269*tair/(237.3+tair))/(237.3+tair)A2 Ih=
  2513.0 - (tair+5.0)*2.4, constrained to (2513,2394)
  psy= 64.3 + 0.06*(tair+5.0), constrained to (64.3,67.5)
```

The net radiation (rn) may be input (see rad_flag in Parameter File) If not input, it is calculated as total shortwave downward radiation (swdown) minus total longwave upward radiation (lwup), where the total is taken over the compositing period:

```
rn = swdown - lwup
```

The shortwave downward radiation is calculated hourly and summed over the compositing period:

```
for each jday in compositing period

   sc = 1360.0*(1.0+0.0335*cos(360.0*jday/365.0) dec =
   -23.45*cos(360.0*(jday+10.0)/365.0)

   for each hour in jday

      hang = hour*15.0
      cossza = (sin(lat)*sin(dec)) + (cos(lat)*cos(dec)*cos(hang)) swdown =
      swdown + (0.25+(0.5*sf))*0.83*sc*cossza

   end hourly loop

end jday loop
```

In this calc~lation sc is the solar constant, dec is solar declination, hang is the hour angle, cossza is the cosine of the solar zenith angle, sf is the mean sunlight fraction for the compositing period, lat is the latitude, and ((0.25+(0.5*sf))*0.83*sc*cossza) is the shortwave. downward radiation for a single hour of a single Julian day. Note that the code converts all angles to radians before perfoming trig functions. If photosynthetically active radiation (par) is not given as an input to the model (see rad_flag specification in Parameter File section), then it is estimated here as half of the total shortwave downward radiation for the compositing period.

~

The longwave upward radiation is calculated daily and summed over the compositing period:

```
for each jday in compositing period

   lwup = lwup + (0.2+0.8*sf)*(107.0-tair)*24

end jday loop
```

In this calculation tair is the canopy air temperature and sf is the mean sunlight fraction for the compositing period. Note that tair is converted to Celcius before performing the calculation.

actual evapotranspiration
****

Actual evapotranspiration (aet) is the minimum of the water demand and the water supply:

aet = min(water_demand, water_supply)

Details for the water demand calculation are given above.

The water supply is a function of maximum evapotranspiration (max_et), soil water content in the previous time step (prev_swc), and soil water holding capacity (whc):

water_supply = (max_et*prev_swc)/whc

The maximum evapotranspiration (max_et) is estimated as a function of air temperature (tair, converted to Celcius), local heat index (lhi) and potential evapotranspiration (pet):

max_et = 16.0*(10.0*tair/lhi)Apet

The previous soil water content (prev_swc) is either given as an input (for the first compositing period of the year), or it was calculated in this function for the preceding compositing period (for" all but the first compositing period of the year). The calculation of soil water content (swc) for the current time step is:

swc = prev_swc + precip - aet

Here precip is the total precipiation for the compositing period, prev_swc is the soil water content for the previous compositing period, and aet is the actual evapotranspiration for this compositing period calculated as given above. The swc is constrained to be between zero and the soil water holding capacity (whc).

reference: Bugmann and Cramer 1998

arguments: timeid firstday compdays yr rad_flag

timeid = the time identifier
firstday = the first Julian day of the compositing period
compdays = the number of Julian days in the compositing period four-digit
yr = the year containing the compositing period
rad flag = the radiation calculation flag (see Parameter File)

inputs: tair precip Ihi <par> <sunfr> prev_swc dtr latitude whc
pet <rn>

NOTE: <> denotes optional inputs; see rad_flag values in Parameter File dection.

tair = canopy air temperature
precip = total precipitation for the compositing period
lhi = local heat index
par = phototsynthetically active radiation incident on the canopy sunfr = mean sunlight fraction
prev_swc = soil water content for the previous compositing period dtr = mean diurnal temperature range
latitude = latitude
whc = soil water holding capacity
.pet = potential evapotranspiration
rn = net radiation

outputs: swc stress swc

swc = soil water content stress_swc = soil moisture stress

stress_tair.c
****

Calculates air temperature stress. Stress is output binary as an 8 bit unsigned GPP image representing the percentage (0-100) of after reductions remaining higher the air for air temperature stress, ie the temperature stress value GPP. the lower the reduction in The flag value is 101.

The air temperature stress is a function of the deviation of canopy air temperature (tair) from optimal air temperature (topt). The stress is calculated as (rounded to the nearest integer):

stress tair = 100*R*(1 - O.l*R)

Where

R = ((tair-tmin)*(tair-tmax))/
((tair-tmin)*(tair-tmax) - (tair-topt)A2)

Here tmin = 268.15 Kelvin, tmax = 318.15, tair is constrained to [268.15,318.15] Kelvin, and topt is constrained to [293.15,301.15]. Kelvin.

~

reference: unpublished, algorithm from MK Cao

arguments: timeid

timeid = the time identifier

inputs: tair topt

tair = canopy air temperature topt = optimal air temperature

outputs: stress_tair

stress_tair the percentage of GPP remaining after reduction for air temperature stress (deviation from optimal temperature)

stress_vpd.c
****

(

Calculates vapor pressure deficit stress. Stress is output as an 8 bit unsigned binary image representing the percentage (0-100) of GPP remaining after reductions for vapor pressure deficit stress, ie the higher the vapor pressure deficit stress value the lower the reduction in GPP. The flag value is 101. Also calculates vapor pressure deficit.

The vapor pressure deficit stress (stress_vpd) is a function of the ratio of surface vapor pressure (vp) to saturated vapor pressure (satvp). The calculation is (rounded to the nearest integer) :

stress vpd = 100*R*(1 - 0.2*R)

Where

R = vp/satvp

The saturated vapor pressure is calculated from canopy air temperature (tair) and diurnal temperature range (dtr):

satvp = 0.611*exp(17.27*(dhtair-273.0)/(dhtair-36.0))

where dhtair = (tair - 0.25*dtr) is an estimate of the mean temperature for the daylight hours. This assumes that the tair retrieval is an approximate daily maximum.

The vapor pressure deficit (vpd) is simply the difference between the saturated and surface vapor pressures:

vpd = satvp - vp

references: Montieth (eqn 2.23) for the saturated vapor pressure calculation. The stress algorithm is unpublished, from MK Cao.

~

arguments: timeid

timeid = the time identifier

inputs: vp tair dtr

vp = surface vapor pressure tair = canopy air temperature dtr = diurnal temperature range

outputs: vpd stress_vpd

vpd = vapor pressure deficit
stress_vpd = vapor pressure deficit stress

write_binary.c
****

Writes a binary file from allocated memory. The function writes numpx pixels of data type dtype from memory starting at address * imptr .

arguments: numpx fname dtype imptr

numpx = number of pixels (rows*cols) in the image
fname = filename for the binary file
dtype = datatype (see allocate_binary. c)
imptr = address of the pointer (ie double pointer) to the memory allocated for the image, cast to type void

inputs: none

outputs: none

REFERENCES

H Bugmann, W Cramer (1998)
Improving the behaviour of forest gap models along drought gradients. Forest Ecology and Management 103: 247-263.

GJ Collatz, JT Ball, C Grivet, JA Berry (1991)
Physiological and environmental regulation of stomatal conductance, photosynthesis and transpiration: a model that includes a laminar boundary layer. Agricultural and Forest Meteorology 54: 107-136.

SJ Goetz, MM Thawley, J Small, SD Prince, SN Goward (1999) Satellite remote sensing of primary production: An improved efficiency modeling production approach.
Ecological Modelling 122 (3): 239-255.

ER Hunt, Jr. (1994)
Relationship between woody biomass and PAR conversion efficiency for estimating net primary production from NOVI.
International Journal of Remote Sensing 15(8): 1725-1730.

JL Montieth (1973)
Principles of Environmental Physics. New York, American Elsevier Publishing.

SO Prince, SN Goward (1995)
Global primary production: a remote sensing approach. Journal of Biogeography 22: 815-835.