**Grasstree GPP, phenological controls, and response to drought and cold**

Melannie Hartman

September 14, 2021

Note: this document contains notes about the DayCent code itself which may not be of interest to all readers; however, it also contains tables of the parameters that control GPP, autotrophic respiration, phenology, and death of grasstrees.

**Global variables**

dosene – this variable is reset to FALSE each day in subroutine schedl, but set to TRUE when SENM event is found in schedule file.

senecnt – set to 1 in dshootgt when (dosene .and. (seneday .eq. curday)) = TRUE, incremented daily in simsom up to a max of 30, then reset to 0. Crop and grasstree growth will not occur if senecnt > 0.

**Proposed photoperiod controls on senescence:**

* Let there be an option to control the commencement of senescence by photoperiod instead of by schedule file control.
* Senescence is triggered when daylength is decreasing (after the autumn equinox) and is less than or equal to a photoperiod parameter. This parameter specifies the threshold daylength (hours). At this point growth is reduced and senescence continues for the next 30 days.
* Senescence in DayCent stops growth for 30 days while also killing live above-ground biomass.

Could potentially use MXDYSENE parameter from GDD model as parameter.

Probably need a new parameter named SENEDAYLN.

Daylength is computed in initrad.c

How to implement photoperiod senescence vs. that forced from a schedule file?

* The GSEN event will allow photoperiod senescence to happen as soon as photoperiod criteria are met (daylength decreasing and daylength <= DYLENSEN < 12.0, then senescence will continue for the next GSENDYS days. It will be illegal to schedule a SENM event during the same period. However, a SENM event can occur before or after the GSEN event’s duration.

(.not. hrsinc) .and. (dayhrs .lt. 12.0)

Note: growth stops when senecnt > 0 (growth.f, grasstreegrow.f).

Two new grasstree.100 parameters: DYLENSENE and GSENEDYS.

| DYLENSEN | The daylength (hours) that triggers a GSEN event. Senescence will commence when daylength ≤ DYLENSEN and daylength is decreasing. | ≤ 12.0, and feasible for the latitude of the site. |
| --- | --- | --- |
| GSENEDYS | Number of days a GSEN event persists. Senescence will commence when daylength ≤ DYLENSEN and daylength is decreasing. | ~30 |

BTW: In the crop model, there was a scenfrac that controlled photosynthetically active leave carbon. However, for grasstrees we are considering a senescence triggered by photoperiod (not GDD) that will cause death of gtleavc thus automatically reducing aggreenc. From dailymoist.f:

|  |
| --- |
| c ..... Set an upper limit on the calculation for accumulating growing  c ..... degree days such that when the maximum temperature for the day  c ..... is capped at basetemp(2) degrees C, cak - 05/21/2008  thermtemp = tmxs\_mlt \* min(tempmax(curday), basetemp(2)) +  & tmns\_mlt \* max(tempmin(curday), basetemp(1))  dayunits = max(0.0, thermtemp - basetemp(1))  thermunits = thermunits + dayunits  c ..... Set the emergence flag to true when enough growing degree days  c ..... have occurred after planting (DDEMERG) for a grain filling  c ..... annual  if ((thermunits .ge. ddemerg) .and. (frtcindx .ge. 4)) then  emerg = .true.  endif  c ..... For a grain producing crop reaching ddbase starts the grain  c ..... filling period, cak 06/02/05  c ..... Add code to simulate the senescence of annual plants, beginning  c ..... at anthesis (DDBASE), the amount of photosynthetic active carbon  c ..... (aggreenc) decreases with age, based on water stress, affecting  c ..... potential growth and transpiration, 06/10/2014  c ..... scenfrac is a 0.0 to 1.0 multiplier used to indicate the  c ..... fraction of the above live carbon that is photosynthetic active  c ..... carbon  c ..... 1.0 = no senescence has occurred, 100% photosynthetic active  c ..... carbon  c ..... 0.0 = full senescence, 0% photosynthetic active carbon  if ((thermunits .ge. ddbase) .and. (frtcindx .ge. 4)) then  if (.not. grnfill) then  grnfill = .true.  gwstress = 0.0  grnfldys = 1  endif  c ....... Check to see if the plant has reached maturity  if (thermunits .ge. (ddbase + mxddhrv)) then  c ......... Stop plant growth  crpgrw = 0  cgrwdys = 0  scenfrac = 0.0  else  c ......... Use the grain water stress term to determine if we  c ......... have reached maturity, full senescence  hwstress = ramp(gwstress/grnfldys, 0.0, mnddhrv,  & 1.0, mxddhrv)  scenfrac = ramp(thermunits-ddbase, 0.0, 1.0, hwstress,  & 0.0)  endif  grnfldys = grnfldys + 1  endif  endif  c ... Include tree and grasstree leaf area in aggreenc for transpiration calculations. -mdh 6/7/2019  c aggreenc = aglivc \* scenfrac  aggreenc = aglivc \* scenfrac + rleavc + gtleavc |

**SUBROUTINE dshootgt: Death of grasstree leaves and stems from senescence or dry conditions. Additional death of leaves from shading.**

This subroutine is called from subroutine grasstrees once a day (unconditionally)

When there is death of leaves and stems, retranslocation of N happens only from leaves, not stems.

Partition sdethe(GTLEAF) between internal storage (gtstg) and dead attached leaves.

Future updates: Currently there is no leaf or stem death from cold. Add death of live leaves and stems from cold stress.

Could potentially use TMPKILL parameter from GDD model as parameter.

| TMPKILL | Temperature at which growth will stop when using the growing degree day submodel, will cause a SENM and LAST event for a perennial (GTFRTCINDX = 3) or a HARV and LAST event for an annual (GTFRTCINDX = 4) if the required number of thermal units have not been accumulated prior to trigger a SENM or a HARV event. | °C |
| --- | --- | --- |

Table . grasstree.100 parameters used in subroutine dshootgt

| GTFSDETH(1) | Maximum leaf death rate at very dry soil conditions (fraction/month); to get the daily leaf death rate, this fraction divided by the number of days in a month then multiplied by a reduction factor which decreases with increased soil moisture (dthppt = 1.0 – bgwfunc). Live leaves that die are transferred to dead attached leaf pool. | fraction |
| --- | --- | --- |
| GTFSDETH(2) | Maximum stem death rate at very dry soil conditions (fraction/month); to get the daily stem death rate, this fraction divided by the number of days in a month then multiplied by a reduction factor which decreases with increased soil moisture (dthppt = 1.0 – bgwfunc). Live stems that die are transferred to standing dead stem pool. | fraction |
| GTFSDETH(3) | Fraction of leaves which die on the day of the SENM event. Live leaves that die are transferred to dead attached leaf pool. | fraction |
| GTFSDETH(4) | Fraction of stems which die on the day of the SENM event. Live stems that die are transferred to standing dead stem pool. | fraction |
| GTFSDETH(5) | Additional fraction of leaves which die when aboveground live leaf C is greater than GTFSDETH(6) and GTFSDETH(3) < 0.4 (traditionally GTFSDETH(3) > 0.4 during senescence, but there is no code check for this condition). Live stems that die are transferred to dead attached leaf pool. | fraction |
| GTFSDETH(6) | Level of live leaf C which shading occurs and shoot death increases (see GTFSDETH(5)). | g C m-2 |

**SUBROUTINE drootgt: Death of grasstree fine and coarse roots.**

This subroutine is called from subroutine grasstrees once a day (unconditionally)

Future updates: Currently, soil temperature during the winter is often insulated by snow so soil temperatures alone may not be enough to calculate death of coarse roots (rhizomes). In the future, incorporate running average air temperature into the temperature effect on death of coarse roots since rhizomes sit near the surface. Coarse roots and fine roots will have different temperature effects.

Soil temperature and moisture effect on fine root death rate:

tempeff = (soiltemp - 10.0)\*\*2 / 4.0 \* 0.00175 + 0.1 (Rhizomes use a running average air temperature)

tempeff = min(tempeff, 0.5)

watreff = maxswpot(gtlaypg) (Rhizomes use a shallower depth)

watreff = carctanf(watreff, 35.0, 0.5, 1.0, 0.05)

rtdh = max(tempeff, watreff)

rtdh = max(rtdh, 0.0)

|  |  |
| --- | --- |
|  |  |

Table . grasstree.100 parameters used in subroutine drootgt

| ROOTDR(1) | Maximum **juvenile fine root** death rate with very dry soil or very cold/hot soil conditions (fraction/month); to get the daily root death rate, this fraction is divided by the number of days in the month then multiplied by another fraction (rtdh) that decreases as soil moisture increases and soil temperature approaches optimal conditions (~10 °C).  rdfrac = min(0.95, rootdr(1) \* tfrac \* rtdh) | fraction |
| --- | --- | --- |
| ROOTDR(2) | Maximum **mature fine root** death rate with very dry soil or very cold/hot soil conditions (fraction/month); to get the daily root death rate, this fraction is divided by the number of days in the month then multiplied by another fraction (rtdh) that decreases as soil moisture increases and soil temperature approaches optimal conditions (~10 °C).  rdfrac = min(0.95, rootdr(2) \* tfrac \* rtdh) | fraction |
| ROOTDR(3) | Maximum **coarse root** death rate with very dry soil or very cold/hot soil conditions (fraction/month); to get the daily root death rate, this fraction is divided by the number of days in the month then multiplied by another fraction (rtdh) that decreases as soil moisture increases and soil temperature approaches optimal conditions (~10 °C).  rdfrac = min(0.95, rootdr(3) \* tfrac \* rtdh) | fraction |
| FRTDSRFC | Fraction of dead fine roots (juvenile and mature) that are transferred into the surface litter layer (STRUCC(1) and METABC(1)) upon death, the remainder of the dead fine roots will go to the soil litter layer (STRUCC(2) and METABC(2)) | fraction |
| CRTDSRFC | Fraction of dead coarse roots that are transferred into the surface litter layer (STRUCC(1) and METABC(1)) upon death, the remainder of the dead coarse roots will go to the soil litter layer (STRUCC(2) and METABC(2)) | fraction |

When coarse roots die, carbohydrate storage (carbostg) and internal N storage (gtstg) are removed in proportion to the fraction of live coarse roots that die.

**SUBROUTINE gtkillroot**

TREM event (harvest) for GrassTrees. Death of fine and coarse roots due to cutting or fire.

grasstree.100

| FRTDSRFC | Fraction of dead fine roots (juvenile and mature) that are transferred into the surface litter layer (STRUCC(1) and METABC(1)) upon death, the remainder of the dead fine roots will go to the soil litter layer (STRUCC(2) and METABC(2)) | fraction |
| --- | --- | --- |
| CRTDSRFC | Fraction of dead coarse roots that are transferred into the surface litter layer (STRUCC(1) and METABC(1)) upon death, the remainder of the dead coarse roots will go to the soil litter layer (STRUCC(2) and METABC(2)) | fraction |

trem.100

|  |  |  |
| --- | --- | --- |
| FD(1) | Fraction of fine root components that die. | fraction |
| FD(2) | Fraction of coarse root components that die. | fraction |

When coarse roots die, carbohydrate storage (carbostg) and internal N storage (gtstg) are removed in proportion to the fraction of live coarse roots that die.

**GrassTree production**

Summarize how actual production is calculated from potential production and any limitations related to carbohydrate storage (carbostg) and internal N storage (gtstg(1)).

Carbon from gross photosynthesis is added to carbostg in **subroutine potprd**.

c ....... Add the gross photosynthesis to the carbohydrate storage pool

c ....... cak - 08/12/2009

call csched(grossPsn,cisotf,1.0,

& csrsnk(UNLABL),carbostg(GTARYINDX,UNLABL),

& csrsnk(LABELD),carbostg(GTARYINDX,LABELD),

& 1.0,accum)

Carbon for actual production and autotrophic respiration will be removed from carbostg in **subroutine grasstreegrowth**.

**SUBROUTINE grasstreegrowth**

In future updates, we will want carbostg to regulate actual production. Currently, if the carbohydrate pool drops below 15 gC/m2, then 15 gC/m2 is simply added to it. Currently there is no restriction on actual production even when there is not enough carbostg to support actual production.

In future updates, we will want to implement the late growing season shift of C allocation from plant growth to carbohydrate storage. During the late growing season transpiration will also be reduced. The code to do this is only partially implemented.

Note: this subroutine increments grasstree grow days (gtgrwdys), the number of days the grasstree has been growing, but I could only find one place where it is used in the model. In **subroutine potprod**, when gtgrwdys > 0, gtpsndys is incremented and is used by calcPhotsyn…

The values of gtgrwdys and gtpsndys are reset to zero on GFST and GLST events (see subroutine simsom).

In future updates, gtgrwdys should be compared to gturgdys (grasstree.100) to determine when to shift C & N allocation from plant biomass to carbohydrate storage and internal N storage. See treegrow for an example of how to implement restricted growth days for trees.

From crop.100

|  |  |  |  |
| --- | --- | --- | --- |
| CURGDYS | Number of days of unrestricted growth in a grass/crop system. | number of days |  |
| CLSGRES | Grass/crop late season growth restriction factor. |  | 0.0 – 1.0 |

From tree.100

|  |  |  |  |
| --- | --- | --- | --- |
| FURGDYS | Number of days of unrestricted wood growth in a deciduous forest system | number of days |  |
| FLSGRES | Deciduous forest late season growth restriction factor. |  | 0.0 – 1.0 |

|  |
| --- |
| From treegrow.f:  **Example of limiting C allocation to plants (allowing more to remain in carbostg) when there is restricted growth**  c ..... If the tree is deciduous and the time allotted to grow the  c ..... woody components of the trees has passed use the late season  c ..... growth restriction parameter value to determine how much  c ..... carbohydrate to flow out of the forest carbohydrate storage  c ..... pool for the woody components, cak - 03/11/2010  if ((decid .eq. 1) .and. (fgrwdys .gt. furgdys)) then  c ....... Growth of fine branches; split into labeled & unlabeled parts  call csched(mfprd(FBRCH)\*(1.0-flsgres),fraclblstg,1.0,  & carbostg(FORSYS,UNLABL),fbrcis(UNLABL),  & carbostg(FORSYS,LABELD),fbrcis(LABELD),  & 1.0,afbcis)  c ....... Growth of large wood; split into labeled & unlabeled parts  call csched(mfprd(LWOOD)\*(1.0-flsgres),fraclblstg,1.0,  & carbostg(FORSYS,UNLABL),rlwcis(UNLABL),  & carbostg(FORSYS,LABELD),rlwcis(LABELD),  & 1.0,alwcis)  c ....... Growth of coarse roots; split into labeled & unlabeled parts  call csched(mfprd(CROOT)\*(1.0-flsgres),fraclblstg,1.0,  & carbostg(FORSYS,UNLABL),crtcis(UNLABL),  & carbostg(FORSYS,LABELD),crtcis(LABELD),  & 1.0,acrcis)  else  c ....... Growth of fine branches; split into labeled & unlabeled parts  call csched(mfprd(FBRCH),fraclblstg,1.0,  & carbostg(FORSYS,UNLABL),fbrcis(UNLABL),  & carbostg(FORSYS,LABELD),fbrcis(LABELD),  & 1.0,afbcis)  c ....... Growth of large wood; split into labeled & unlabeled parts  call csched(mfprd(LWOOD),fraclblstg,1.0,  & carbostg(FORSYS,UNLABL),rlwcis(UNLABL),  & carbostg(FORSYS,LABELD),rlwcis(LABELD),  & 1.0,alwcis)  c ....... Growth of coarse roots; split into labeled & unlabeled parts  call csched(mfprd(CROOT),fraclblstg,1.0,  & carbostg(FORSYS,UNLABL),crtcis(UNLABL),  & carbostg(FORSYS,LABELD),crtcis(LABELD),  & 1.0,acrcis)  endif  c ..... Growth respiration  fgrspdyflux(LEAF) = mfprd(LEAF) \* fgresp(LEAF)  fgrspdyflux(FROOTJ) = mfprd(FROOTJ) \* fgresp(FROOTJ)  fgrspdyflux(FROOTM) = mfprd(FROOTM) \* fgresp(FROOTM)  if ((decid .eq. 1) .and. (fgrwdys .gt. furgdys)) then  fgrspdyflux(FBRCH) = mfprd(FBRCH) \* (1.0 - flsgres) \* fgresp(FBRCH)  fgrspdyflux(LWOOD) = mfprd(LWOOD) \* (1.0 - flsgres) \* fgresp(LWOOD)  fgrspdyflux(CROOT) = mfprd(CROOT) \* (1.0 - flsgres) \* fgresp(CROOT)  else  fgrspdyflux(FBRCH) = mfprd(FBRCH) \* fgresp(FBRCH)  fgrspdyflux(LWOOD) = mfprd(LWOOD) \* fgresp(LWOOD)  fgrspdyflux(CROOT) = mfprd(CROOT) \* fgresp(CROOT)  endif  **Example of limiting plant N uptake from internal storage when there is restricted growth**  c ....... Take up nutrients from internal storage pool  c ....... Don't allow uptake from storage if forstg is negative -mdh 8/8/00  c ....... If the tree is deciduous and the time allotted to grow the  c ....... woody components of the trees has passed use the late season  c ....... growth restriction parameter value to determine how much  c ....... nutrients to flow out of the forest nutrient storage pool for  c ....... the woody components, cak - 03/11/2010  if (forstg(iel) .gt. 0.0) then  amt = uptake(ESTOR,iel) \* euf(LEAF)  call flow(forstg(iel),rleave(iel),time,amt)  eupprt(LEAF,iel) = eupprt(LEAF,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodfdy(iel) = eprodfdy(iel) + amt  amt = uptake(ESTOR,iel) \* euf(FROOTJ)  call flow(forstg(iel),frootej(iel),time,amt)  eupprt(FROOT,iel) = eupprt(FROOT,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodfdy(iel) = eprodfdy(iel) + amt  amt = uptake(ESTOR,iel) \* euf(FROOTM)  call flow(forstg(iel),frootem(iel),time,amt)  eupprt(FROOT,iel) = eupprt(FROOT,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodfdy(iel) = eprodfdy(iel) + amt  if ((decid .eq. 1) .and. (fgrwdys .gt. furgdys)) then  amt = uptake(ESTOR,iel) \* euf(FBRCH) \* (1.0 - flsgres)  call flow(forstg(iel),fbrche(iel),time,amt)  eupprt(FBRCH,iel) = eupprt(FBRCH,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodfdy(iel) = eprodfdy(iel) + amt  amt = uptake(ESTOR,iel) \* euf(LWOOD) \* (1.0 - flsgres)  call flow(forstg(iel),rlwode(iel),time,amt)  eupprt(LWOOD,iel) = eupprt(LWOOD,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodfdy(iel) = eprodfdy(iel) + amt  amt = uptake(ESTOR,iel) \* euf(CROOT) \* (1.0 - flsgres)  call flow(forstg(iel),croote(iel),time,amt)  eupprt(CROOT,iel) = eupprt(CROOT,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodfdy(iel) = eprodfdy(iel) + amt  else  amt = uptake(ESTOR,iel) \* euf(FBRCH)  call flow(forstg(iel),fbrche(iel),time,amt)  eupprt(FBRCH,iel) = eupprt(FBRCH,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodfdy(iel) = eprodfdy(iel) + amt  amt = uptake(ESTOR,iel) \* euf(LWOOD)  call flow(forstg(iel),rlwode(iel),time,amt)  eupprt(LWOOD,iel) = eupprt(LWOOD,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodfdy(iel) = eprodfdy(iel) + amt  amt = uptake(ESTOR,iel) \* euf(CROOT)  call flow(forstg(iel),croote(iel),time,amt)  eupprt(CROOT,iel) = eupprt(CROOT,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodfdy(iel) = eprodfdy(iel) + amt  endif  endif  **Example of allocating mineral N to forstg when there is restricted growth**  c ........... Fine branch  namt = 0.0  if ((decid .eq. 1) .and. (fgrwdys .gt. furgdys))then  amt = calcup \* euf(FBRCH) \* flsgres  namt = namt + amt  call flow(minerl(lyr,iel),forstg(iel),time,amt)  eupprt(FBRCH,iel) = eupprt(FBRCH,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodfdy(iel) = eprodfdy(iel) + amt  amt = calcup \* euf(FBRCH) \* (1.0 - flsgres)  namt = namt + amt  call flow(minerl(lyr,iel),fbrche(iel),time,amt)  eupprt(FBRCH,iel) = eupprt(FBRCH,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodfdy(iel) = eprodfdy(iel) + amt  else  amt = calcup \* euf(FBRCH)  namt = namt + amt  call flow(minerl(lyr,iel),fbrche(iel),time,amt)  eupprt(FBRCH,iel) = eupprt(FBRCH,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodfdy(iel) = eprodfdy(iel) + amt  endif |

**Grasstreegrow.f**

|  |
| --- |
| c ... If growth can occur  if (grasstreegrw .eq. 1 .and. pgrasstreec .gt. 0.0 .and.  & .not. (senecnt .gt. 0)) then  c ..... Calculate actual production values  c ..... Calculate impact of root biomass on available nutrients  rimpct = rtimp(riint, rictrl, gtfrootcj+gtfrootcm)  c ..... Determine actual production values, restricting the C/E ratios  c ..... When calling restrp we are only looking at allocation to  c ..... fine roots and leaves, cak - 07/02/02  c ..... Grasstree parts 1 & 2 are leaves and stems, not leaves and fine  c ..... roots, so cergrasstree and grasstree\_cfrac can not be passed to  c ..... restrp here. Copy leaf and fine root values into local arrays  c ..... (cerat and cfrac) to pass to restrp. Neither cerat nor cfrac  c ..... are updated by the call to restrp. -mdh 3/21/2014  do 57 iel = 1, MAXIEL  cerat(IMIN,1,iel) = cergrasstree(IMIN,GTLEAF,iel)  cerat(IMAX,1,iel) = cergrasstree(IMAX,GTLEAF,iel)  cerat(IMIN,2,iel) = cergrasstree(IMIN,GTFROOTJ,iel)  cerat(IMAX,2,iel) = cergrasstree(IMAX,GTFROOTJ,iel)  57 continue  cfrac(1) = grasstree\_cfrac(GTLEAF)  cfrac(2) = grasstree\_cfrac(GTFROOTJ)  call restrp(elimit, nelem, availm, cerat, 2,  & cfrac, pgrasstreec, rimpct, gtstg,  & snfxmx(GTARYINDX), cprodgtdy, eprodgtdy, uptake,  & grasstree\_a2drat, gtNfix, relyld)  else  cprodgtdy = 0.0  do 58 iel = 1, MAXIEL  eprodgtdy(iel) = 0.0  58 continue  endif  c ... If growth occurs...  if (cprodgtdy .gt. 0.) then  c ..... If the carbohydrate storage pool falls below a critical value  c ..... add a minimal amount of carbon from the csrsnk to allow plant  c ..... growth. This should only occur when the grasstree is small.  if (carbostg(GTARYINDX,UNLABL)+carbostg(GTARYINDX,LABELD)  & .lt. 15.0) then  if (gtfrtcindx .eq. 1) then  write(\*,\*) 'Warning, carbostg pool < minimal in grasstreegrow'  write(\*,\*) 'time carbostg = ', time,  & carbostg(GTARYINDX,UNLABL)+carbostg(GTARYINDX,LABELD)  endif  carbostg(GTARYINDX,UNLABL) = carbostg(GTARYINDX,UNLABL) +  & (15.0 \* (1.0 - fraclblstg))  carbostg(GTARYINDX,LABELD) = carbostg(GTARYINDX,LABELD) +  & (15.0 \* fraclblstg)  csrsnk(UNLABL) = csrsnk(UNLABL) - (15.0 \* (1.0 - fraclblstg))  csrsnk(LABELD) = csrsnk(LABELD) - (15.0 \* fraclblstg)  if (gtfrtcindx .eq. 1) then  write(\*,\*) 'grasstreegrow: reset carbostg =', carbostg(3,1)  endif  endif  c ..... Increment the counter that is tracking the number of days to  c ..... allow unrestricted stem and coarse root growth  c ..... Note 6/11/2021 - gtgrwdys is only used in subroutine potprod.  gtgrwdys = gtgrwdys + 1  c ..... Compute carbon allocation fractions for each tree part  c ..... Calculate how much of the carbon the roots use  cprodgtLeft = cprodgtdy \* (1.0 - grasstree\_cfrac(GTFROOT))  c ..... Calculate how much of the carbon the leaves use, allocate leaves  c ..... up to a optimal LAI  grasstree\_cfrac(GTLEAF) = gtleafa(gtleavc, gtstemc,  & cprodgtLeft, cprodgtdy)  remCfrac = 1.0-grasstree\_cfrac(GTFROOT)-grasstree\_cfrac(GTLEAF)  c ..... If we have leftover carbon allocate it to the stems and coarse  c ..... roots using a weighted average  if (remCfrac .lt. 1.0E-05) then  grasstree\_cfrac(GTSTEM) = 0.0  grasstree\_cfrac(GTCROOT) = 0.0  else  totCup = 0.0  grasstree\_cfrac(GTSTEM) = gtcfrac(GTSTEM)  grasstree\_cfrac(GTCROOT) = gtcfrac(GTCROOT)  totCup = grasstree\_cfrac(GTSTEM) + grasstree\_cfrac(GTCROOT)  if (totCup .gt. 0.0) then  grasstree\_cfrac(GTSTEM) = grasstree\_cfrac(GTSTEM)  & / totCup \* remCfrac  grasstree\_cfrac(GTCROOT) = grasstree\_cfrac(GTCROOT)  & / totCup \* remCfrac  else  write(\*,\*) 'Error in grasstreegrow'  write(\*,\*) 'gtcfrac(GTSTEM)+gtcfrac(GTCROOT) <= 0'  STOP  endif  endif  c ... Error checking  sum\_cfrac = 0.0  do 90 ipart = 1, GTLIVPARTS  if (grasstree\_cfrac(ipart) .lt. 0.0) then  write(\*,\*) 'Error in grasstreegrow, cfrac(ipart) < 0'  STOP  else if (grasstree\_cfrac(ipart) .gt. 1.0) then  write(\*,\*) 'Error in grasstreegrow, cfrac(ipart) > 1'  STOP  else  if (ipart .ne. GTFROOTM) then  sum\_cfrac = sum\_cfrac + grasstree\_cfrac(ipart)  endif  endif  90 continue  if (abs(1.0 - sum\_cfrac) .gt. 0.001) then  write(\*,\*) "Error in tree carbon allocation fractions!"  write(\*,\*) "sum\_cfrac = ", sum\_cfrac  c STOP  endif  c ..... Recalculate actual production values with updated C-allocation  c ..... fractions, restricting the C/E ratios -mdh 5/11/01  c ..... Calculate impact of root biomass on available nutrients  rimpct = rtimp(riint, rictrl, gtfrootcj+gtfrootcm)  c ..... Determine actual production values, restricting the C/E ratios  c ..... To be consistent with treegrow, do not include mature fine roots  c ..... -mdh 2/22/2019  call restrp(elimit, nelem, availm, cergrasstree, GTLIVPARTS-1,  & grasstree\_cfrac, pgrasstreec, rimpct, gtstg,  & snfxmx(GTARYINDX), cprodgtdy, eprodgtdy, uptake,  & grasstree\_a2drat, gtNfix, relyld)  c ..... Calculations for symbiotic N fixation accumulators moved from  c ..... nutrlm subroutine, cak - 10/17/02  c ..... Compute N fixation which actually occurs and add to the  c ..... N fixation accumulator.  nfix = nfix + gtNfix  snfxac(GTARYINDX) = snfxac(GTARYINDX) + gtNfix  c ..... Add computation for nfixac -mdh 1/16/02  nfixac = nfixac + gtNfix  c ..... C/N ratio for production  if (eprodgtdy(N) .eq. 0.0) then  write(\*,\*) 'Error in grasstreegrow, eprodgtdy(N) = 0.0)'  STOP  endif  tcnpro = cprodgtdy/eprodgtdy(N)  c ..... Calculate production for each grasstree part  do 95 ipart = 1, GTLIVPARTS  if (ipart .eq. GTLEAF .or. ipart .eq. GTSTEM .or.  & ipart .eq. GTCROOT) then  mgtprd(ipart) = grasstree\_cfrac(ipart) \* cprodgtdy  else if (ipart .eq. GTFROOTJ) then  mgtprd(ipart) = grasstree\_cfrac(GTFROOTJ) \* cprodgtdy  & \* (1.0 - gtmrtfrac)  else if (ipart .eq. GTFROOTM) then  mgtprd(ipart) = grasstree\_cfrac(GTFROOTJ) \* cprodgtdy  & \* gtmrtfrac  else  write(\*,\*) 'Error in grasstreegrow, ipart out of bounds = ',  & ipart  STOP  endif  95 continue  c ..... GrassTree Growth  c ..... All growth comes from the carbohydrate pool, cak - 08/12/2009  if (isnan(carbostg(GTARYINDX,UNLABL))) then  write(\*,\*) 'grasstreegrowth 1: carbostg(3,UNLABL) is NaN'  STOP  endif  c ..... Growth of leaves split into labeled & unlabeled parts  if(isnan(mgtprd(GTLEAF))) then  write(\*,\*) 'grasstreegrow mgtprd(GTLEAF) is NaN'  STOP  endif  call csched(mgtprd(GTLEAF),fraclblstg,1.0,  & carbostg(GTARYINDX,UNLABL),gtlvcis(UNLABL),  & carbostg(GTARYINDX,LABELD),gtlvcis(LABELD),  & 1.0,agtlvcis)  c ..... Growth juvenile fine roots; split into labeled & unlabeled  c ..... parts  if(isnan(mgtprd(GTFROOTJ))) then  write(\*,\*) 'grasstreegrow mgtprd(GTFROOTJ) is NaN'  STOP  endif  call csched(mgtprd(GTFROOTJ),fraclblstg,1.0,  & carbostg(GTARYINDX,UNLABL),gtfrtcisj(UNLABL),  & carbostg(GTARYINDX,LABELD),gtfrtcisj(LABELD),  & 1.0,agtfrcisj)  c ..... Growth mature fine roots; split into labeled & unlabeled parts  if(isnan(mgtprd(GTFROOTM))) then  write(\*,\*) 'grasstreegrow mgtprd(GTFROOTM) is NaN'  STOP  endif  call csched(mgtprd(GTFROOTM),fraclblstg,1.0,  & carbostg(GTARYINDX,UNLABL),gtfrtcism(UNLABL),  & carbostg(GTARYINDX,LABELD),gtfrtcism(LABELD),  & 1.0,agtfrcism)  c ..... Growth of stems; split into labeled & unlabeled parts  if (isnan(carbostg(GTARYINDX,UNLABL))) then  write(\*,\*) 'grasstreegrowth 2: carbostg(3,UNLABL) is NaN'  STOP  endif  if (isnan(gtstmcis(UNLABL))) then  write(\*,\*) 'grasstreegrowth gtstmcis(UNLABL) is NaN'  STOP  endif  call csched(mgtprd(GTSTEM),fraclblstg,1.0,  & carbostg(GTARYINDX,UNLABL),gtstmcis(UNLABL),  & carbostg(GTARYINDX,LABELD),gtstmcis(LABELD),  & 1.0,agtstmcis)  c ..... Growth of coarse roots; split into labeled & unlabeled parts  if(isnan(mgtprd(GTCROOT))) then  write(\*,\*) 'grasstreegrow mgtprd(GTCROOT) is NaN'  STOP  endif  call csched(mgtprd(GTCROOT),fraclblstg,1.0,  & carbostg(GTARYINDX,UNLABL),gtcrtcis(UNLABL),  & carbostg(GTARYINDX,LABELD),gtcrtcis(LABELD),  & 1.0,agtcrtcis)  c ..... GROWTH RESPIRATION  c ..... Growth respiration is a fixed fraction of production  gtgrspdyflux(GTLEAF) = mgtprd(GTLEAF) \* gtgresp(GTLEAF)  gtgrspdyflux(GTFROOTJ) = mgtprd(GTFROOTJ) \* gtgresp(GTFROOTJ)  gtgrspdyflux(GTFROOTM) = mgtprd(GTFROOTM) \* gtgresp(GTFROOTM)  gtgrspdyflux(GTSTEM) = mgtprd(GTSTEM) \* gtgresp(GTSTEM)  gtgrspdyflux(GTCROOT) = mgtprd(GTCROOT) \* gtgresp(GTCROOT)  c endif  c ..... Growth respiration is subtracted from the carbohydrate  c ..... storage pool.  grspdyflux(GTARYINDX) = 0.0  do 105 ipart = 1, GTLIVPARTS  grspdyflux(GTARYINDX) = grspdyflux(GTARYINDX)  & + gtgrspdyflux(ipart)  105 continue  if(isnan(grspdyflux(GTARYINDX))) then  write(\*,\*) 'grasstreegrow grspdyflux(3) is NaN'  STOP  endif  call csched(grspdyflux(GTARYINDX),fraclblstg,1.0,  & carbostg(GTARYINDX,UNLABL),csrsnk(UNLABL),  & carbostg(GTARYINDX,LABELD),csrsnk(LABELD),  & 1.0,gtautoresp)  c ..... Actual Uptake  do 110 iel = 1, nelem  if (eprodgtdy(iel) .le. 0.0) then  print \*, 'Divide by zero in grasstreegrow: eprodgtdy(i)=',  & eprodgtdy(iel)  STOP  endif  eugt(GTLEAF) = eup(GTLEAF,iel) / eprodgtdy(iel)  eugt(GTFROOTJ) = (eup(GTFROOT,iel) \* (1.0 - gtmrtfrac))  & / eprodgtdy(iel)  eugt(GTFROOTM) = (eup(GTFROOT,iel)\*gtmrtfrac) / eprodgtdy(iel)  eugt(GTSTEM) = eup(GTSTEM,iel) / eprodgtdy(iel)  eugt(GTCROOT) = eup(GTCROOT,iel) / eprodgtdy(iel)  c ....... Reset eprodcdy(iel) to track the actual uptake which can be  c ....... restricted late in the growing season  eprodgtdy(iel) = 0.0  c ....... Take up nutrients from internal storage pool  c ....... Don't allow uptake from storage if gtstg is negative  if (gtstg(iel) .gt. 0.0) then  amt = uptake(ESTOR,iel) \* eugt(GTLEAF)  if(isnan(amt)) then  write(\*,\*) 'grasstreegrow amt 1 is NaN'  STOP  endif  call flow(gtstg(iel),gtleave(iel),time,amt)  eupgtprt(GTLEAF,iel) = eupgtprt(GTLEAF,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodgtdy(iel) = eprodgtdy(iel) + amt  eupstg(GTLEAF,iel) = eupstg(GTLEAF,iel) + amt  amt = uptake(ESTOR,iel) \* eugt(GTFROOTJ)  if(isnan(amt)) then  write(\*,\*) 'grasstreegrow amt 2 is NaN'  STOP  endif  call flow(gtstg(iel),gtfrootej(iel),time,amt)  eupgtprt(GTFROOTJ,iel) = eupgtprt(GTFROOTJ,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodgtdy(iel) = eprodgtdy(iel) + amt  eupstg(GTFROOTJ,iel) = eupstg(GTFROOTJ,iel) + amt  amt = uptake(ESTOR,iel) \* eugt(GTFROOTM)  if(isnan(amt)) then  write(\*,\*) 'grasstreegrow amt 3 is NaN'  STOP  endif  call flow(gtstg(iel),gtfrootem(iel),time,amt)  eupgtprt(GTFROOTM,iel) = eupgtprt(GTFROOTM,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodgtdy(iel) = eprodgtdy(iel) + amt  eupstg(GTFROOTM,iel) = eupstg(GTFROOTM,iel) + amt  amt = uptake(ESTOR,iel) \* eugt(GTSTEM)  if(isnan(amt)) then  write(\*,\*) 'grasstreegrow amt 4 is NaN'  STOP  endif  call flow(gtstg(iel),gtsteme(iel),time,amt)  eupgtprt(GTSTEM,iel) = eupgtprt(GTSTEM,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodgtdy(iel) = eprodgtdy(iel) + amt  eupstg(GTSTEM,iel) = eupstg(GTSTEM,iel) + amt  amt = uptake(ESTOR,iel) \* eugt(GTCROOT)  if(isnan(amt)) then  write(\*,\*) 'grasstreegrow amt 5 is NaN'  STOP  endif  if (amt .lt. 0.0) then  write(\*,\*) 'grasstreegrow neg amt: gtstg->gtcroote', amt  endif  call flow(gtstg(iel),gtcroote(iel),time,amt)  eupgtprt(GTCROOT,iel) = eupgtprt(GTCROOT,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodgtdy(iel) = eprodgtdy(iel) + amt  eupstg(GTCROOT,iel) = eupstg(GTCROOT,iel) + amt  endif  c ....... Take up nutrients from soil  c ....... Nutrients for uptake are available in the top gtlaypg layers.  do 100 lyr = 1, gtlaypg  if (minerl(lyr,iel) .gt. toler) then  fsol = 1.0  c ........... The fsol calculation for P is not needed here to compute  c ........... the weighted average, cak - 04/05/02  c if (iel .eq. P) then  c fsol = fsfunc(minerl(SRFC,P), pslsrb, sorpmx)  c endif  call cmpnfrac(lyr,ammonium,nitrate,minerl,  & frac\_nh4,frac\_no3,no3pref(GTARYINDX))  calcup = uptake(ESOIL,iel)\*minerl(lyr,iel)\*  & fsol/availm(iel)  c ........... Leaves  c ........... minerl -> gtleave  amt = calcup \* eugt(GTLEAF)  if (iel .eq. N) then  namt = -1.0\*amt  call update\_npool(lyr, namt, frac\_nh4, frac\_no3,  & ammonium, nitrate, subname)  endif  if(isnan(amt)) then  write(\*,\*) 'grasstreegrow amt 6 is NaN'  STOP  endif  call flow(minerl(lyr,iel),gtleave(iel),time,amt)  eupgtprt(GTLEAF,iel) = eupgtprt(GTLEAF,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodgtdy(iel) = eprodgtdy(iel) + amt  eupsoil(GTLEAF,iel) = eupsoil(GTLEAF,iel) + amt  c ........... Juvenile fine roots  c ........... minerl -> gtfrootej  amt = calcup \* eugt(GTFROOTJ)  if (iel .eq. N) then  namt = -1.0\*amt  call update\_npool(lyr, namt, frac\_nh4, frac\_no3,  & ammonium, nitrate, subname)  endif  if(isnan(amt)) then  write(\*,\*) 'grasstreegrow amt 7 is NaN'  STOP  endif  call flow(minerl(lyr,iel),gtfrootej(iel),time,amt)  eupgtprt(GTFROOTJ,iel) = eupgtprt(GTFROOTJ,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodgtdy(iel) = eprodgtdy(iel) + amt  eupsoil(GTFROOTJ,iel) = eupsoil(GTFROOTJ,iel) + amt  c ........... Mature fine roots  c ........... minerl -> gtfrootem  amt = calcup \* eugt(GTFROOTM)  if (iel .eq. N) then  namt = -1.0\*amt  call update\_npool(lyr, namt, frac\_nh4, frac\_no3,  & ammonium, nitrate, subname)  endif  if(isnan(amt)) then  write(\*,\*) 'grasstreegrow amt 8 is NaN'  STOP  endif  call flow(minerl(lyr,iel),gtfrootem(iel),time,amt)  eupgtprt(GTFROOTM,iel) = eupgtprt(GTFROOTM,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodgtdy(iel) = eprodgtdy(iel) + amt  eupsoil(GTFROOTM,iel) = eupsoil(GTFROOTM,iel) + amt  c ........... Stems  c ........... minerl -> gtsteme  amt = calcup \* eugt(GTSTEM)  if (iel .eq. N) then  namt = -1.0\*amt  call update\_npool(lyr, namt, frac\_nh4, frac\_no3,  & ammonium, nitrate, subname)  endif  if(isnan(amt)) then  write(\*,\*) 'grasstreegrow amt 9 is NaN'  STOP  endif  call flow(minerl(lyr,iel),gtsteme(iel),time,amt)  eupgtprt(GTSTEM,iel) = eupgtprt(GTSTEM,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodgtdy(iel) = eprodgtdy(iel) + amt  eupsoil(GTSTEM,iel) = eupsoil(GTSTEM,iel) + amt  c ........... Coarse roots  c ........... minerl -> gtcroote  amt = calcup \* eugt(GTCROOT)  c write(\*,\*) 'Lyr ', lyr, ': calcup \* eugt(GTCROOT)=', amt  if (iel .eq. N) then  namt = -1.0\*amt  call update\_npool(lyr, namt, frac\_nh4, frac\_no3,  & ammonium, nitrate, subname)  endif  if(isnan(amt)) then  write(\*,\*) 'grasstreegrow amt 10 is NaN'  STOP  endif  if (amt .lt. 0.0) then  write(\*,\*) 'grasstreegrow neg amt: minerl->gtcroote',amt  endif  call flow(minerl(lyr,iel),gtcroote(iel),time,amt)  eupgtprt(GTCROOT,iel) = eupgtprt(GTCROOT,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodgtdy(iel) = eprodgtdy(iel) + amt  eupsoil(GTCROOT,iel) = eupsoil(GTCROOT,iel) + amt  endif  100 continue  c ....... Take up nutrients from nitrogen fixation  if (iel .eq. N .and. gtNfix .gt. 0) then  c ......... Leaves  c ......... fixation -> gtleave  amt = uptake(ENFIX,iel) \* eugt(GTLEAF)  if(isnan(amt)) then  write(\*,\*) 'grasstreegrow amt 11 is NaN'  STOP  endif  call flow(esrsnk(iel),gtleave(iel),time,amt)  eupgtprt(GTLEAF,iel) = eupgtprt(GTLEAF,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodgtdy(iel) = eprodgtdy(iel) + amt  eupnfix(GTLEAF,iel) = eupnfix(GTLEAF,iel) + amt  c ......... Juvenile fine roots  c ......... fixation -> gtfrootej  amt = uptake(ENFIX,iel) \* eugt(GTFROOTJ)  if(isnan(amt)) then  write(\*,\*) 'grasstreegrow amt 12 is NaN'  STOP  endif  call flow(esrsnk(iel),gtfrootej(iel),time,amt)  eupgtprt(GTFROOTJ,iel) = eupgtprt(GTFROOTJ,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodgtdy(iel) = eprodgtdy(iel) + amt  eupnfix(GTFROOTJ,iel) = eupnfix(GTFROOTJ,iel) + amt  c ......... Mature fine roots  c ......... fixation -> gtfrootem  amt = uptake(ENFIX,iel) \* eugt(GTFROOTM)  if(isnan(amt)) then  write(\*,\*) 'grasstreegrow amt 13 is NaN'  STOP  endif  call flow(esrsnk(iel),gtfrootem(iel),time,amt)  eupgtprt(GTFROOTM,iel) = eupgtprt(GTFROOTM,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodgtdy(iel) = eprodgtdy(iel) + amt  eupnfix(GTFROOTM,iel) = eupnfix(GTFROOTM,iel) + amt  c ......... Stems  c ......... fixation -> gtsteme  amt = uptake(ENFIX,iel) \* eugt(GTSTEM)  if(isnan(amt)) then  write(\*,\*) 'grasstreegrow amt 14 is NaN'  STOP  endif  call flow(esrsnk(iel),gtsteme(iel),time,amt)  eupgtprt(GTSTEM,iel) = eupgtprt(GTSTEM,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodgtdy(iel) = eprodgtdy(iel) + amt  eupnfix(GTSTEM,iel) = eupnfix(GTSTEM,iel) + amt  c ......... Coarse roots  c ......... fixation -> gtcroote  amt = uptake(ENFIX,iel) \* eugt(GTCROOT)  c write(\*,\*) 'uptake(ENFIX,iel) \* eugt(GTCROOT)=', amt  if(isnan(amt)) then  write(\*,\*) 'grasstreegrow amt 15 is NaN'  STOP  endif  if (amt .lt. 0.0) then  write(\*,\*) 'grasstreegrow neg amt: Nfix->gtcroote', amt  endif  call flow(esrsnk(iel),gtcroote(iel),time,amt)  eupgtprt(GTCROOT,iel) = eupgtprt(GTCROOT,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodgtdy(iel) = eprodgtdy(iel) + amt  eupnfix(GTCROOT,iel) = eupnfix(GTCROOT,iel) + amt  endif  c ....... Take up nutrients from automatic fertilizer.  if (aufert .ne. 0) then  if (uptake(EFERT,iel) .gt. 0.) then  c ........... Automatic fertilizer added to plant pools  c ........... Leaves  c ........... Autofert -> gtleave  amt = uptake(EFERT,iel) \* eugt(GTLEAF)  if(isnan(amt)) then  write(\*,\*) 'grasstreegrow amt 16 is NaN'  STOP  endif  call flow(esrsnk(iel),gtleave(iel),time,amt)  eupgtprt(GTLEAF,iel) = eupgtprt(GTLEAF,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodgtdy(iel) = eprodgtdy(iel) + amt  eupaufert(GTLEAF,iel) = eupaufert(GTLEAF,iel) + amt  c ........... Juvenile fine roots  c ........... Autofert -> gtfrootej  amt = uptake(EFERT,iel) \* eugt(GTFROOTJ)  if(isnan(amt)) then  write(\*,\*) 'grasstreegrow amt 17 is NaN'  STOP  endif  call flow(esrsnk(iel),gtfrootej(iel),time,amt)  eupgtprt(GTFROOTJ,iel) = eupgtprt(GTFROOTJ,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodgtdy(iel) = eprodgtdy(iel) + amt  eupaufert(GTFROOTJ,iel) = eupaufert(GTFROOTJ,iel) + amt  c ........... Mature fine roots  c ........... Autofert -> gtfrootem  amt = uptake(EFERT,iel) \* eugt(GTFROOTM)  if(isnan(amt)) then  write(\*,\*) 'grasstreegrow amt 18 is NaN'  STOP  endif  call flow(esrsnk(iel),gtfrootem(iel),time,amt)  eupgtprt(GTFROOTM,iel) = eupgtprt(GTFROOTM,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodgtdy(iel) = eprodgtdy(iel) + amt  eupaufert(GTFROOTM,iel) = eupaufert(GTFROOTM,iel) + amt  c ........... Stems  c ........... Autofert -> gtsteme  amt = uptake(EFERT,iel) \* eugt(GTSTEM)  if(isnan(amt)) then  write(\*,\*) 'grasstreegrow amt 19 is NaN'  STOP  endif  call flow(esrsnk(iel),gtsteme(iel),time,amt)  eupgtprt(GTSTEM,iel) = eupgtprt(GTSTEM,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodgtdy(iel) = eprodgtdy(iel) + amt  eupaufert(GTSTEM,iel) = eupaufert(GTSTEM,iel) + amt  c ........... Coarse roots  c ........... Autofert -> gtcroote  amt = uptake(EFERT,iel) \* eugt(GTCROOT)  c write(\*,\*) 'uptake(EFERT,iel) \* eugt(GTCROOT)=', amt  if(isnan(amt)) then  write(\*,\*) 'grasstreegrow amt 20 is NaN'  STOP  endif  if (amt .lt. 0.0) then  write(\*,\*) 'grasstreegrow neg amt: afert->gtcroote', amt  endif  call flow(esrsnk(iel),gtcroote(iel),time,amt)  eupgtprt(GTCROOT,iel) = eupgtprt(GTCROOT,iel) + amt  eupacc(iel) = eupacc(iel) + amt  eprodgtdy(iel) = eprodgtdy(iel) + amt  eupaufert(GTCROOT,iel) = eupaufert(GTCROOT,iel) + amt  c ........... Automatic fertilizer added to mineral pool  if (favail(iel) .eq. 0.0) then  write(\*,\*) 'Error in grasstreegrow, favail(iel) = 0'  STOP  endif  amt = uptake(EFERT,iel) \* (1.0/favail(iel) - 1.0)  fertot(iel) = fertot(iel) + uptake(EFERT,iel) + amt  fertac(iel) = fertac(iel) + uptake(EFERT,iel) + amt  if (iel .eq. N) then  lyr = SRFC  call update\_npool(lyr, amt, frac\_nh4\_fert,  & frac\_no3\_fert, ammonium, nitrate,  & subname)  endif  if(isnan(amt)) then  write(\*,\*) 'grasstreegrow amt 21 is NaN'  STOP  endif  call flow(esrsnk(iel),minerl(SRFC,iel),time,amt)  endif  endif  110 continue  c ... Else there is no production this time step due to nutrient  c ... limitation  else  cprodgtdy = 0.0  do 140 iel = 1, MAXIEL  eprodgtdy(iel) = 0.0  do 130 ipart = 1, GTLIVPARTS  eup(ipart,iel) = 0.0  130 continue  140 continue  endif |

**Autotrophic respiration and carbohydrate storage**

GTGRESP(\*) multiplies daily production of the grasstree part to compute the amount of growth respiration for that part, but does not effect production itself. Maintenance respiration is a product of GTKMRSPMX(\*) \* live plant C for the tree part \* temperature\_effect \* moisture\_effect \* carbohydrate\_storage\_effect. All autotrophic respiration is removed from the carbohydrate storage pool. The GTKMRSPMX and GTGRESP parameters are not supposed to add to 1.0.

Table 3. Parameters in grasstree.100 that control maintenance and growth respiration.

| **GTKMRSPMX(\*)** | **These parameters determine the relative maintenance respiration rate for each plant part and are used in subroutine grasstreegrow. Maintenance respiration flux is proportional to live carbon and increases with average daily temperature. For roots, maintenance respiration is reduced by dry soils.** |  |  |
| --- | --- | --- | --- |
| GTKMRSPMX(1) | Maximum fraction of live leaf C that goes to maintenance respiration for grasstrees. | fraction | 0.0 – 1.0 |
| GTKMRSPMX(2) | Maximum fraction of live stem C that goes to maintenance respiration for grasstrees | fraction | 0.0 – 1.0 |
| GTKMRSPMX(3) | Maximum fraction of live juvenile fine root C that goes to maintenance respiration for grasstrees | fraction | 0.0 – 1.0 |
| GTKMRSPMX(4) | Maximum fraction of live coarse root C that goes to maintenance respiration for grasstrees | fraction | 0.0 – 1.0 |
| GTKMRSPMX(5) | Maximum fraction of live mature fine root C that goes to maintenance respiration for grasstrees | fraction | 0.0 – 1.0 |
| **GTMRSPLAI(\*)** | **See *mrspReduce.xlsx* for more information about the maintenance respiration parameters below. These parameters control the carbohydrate\_storage\_effect on total maintenance respiration.** |  |  |
| GTMRSPLAI(1) | X1 value for line function that decreases maintenance respiration based on optimal leaf carbon when the amount of carbon in the carbohydrate storage pool is less than  (GTMRSPLAI (3) \* optimal leaf carbon) for a grasstree system |  |  |
| GTMRSPLAI(2) | Y1 value for line function that decreases maintenance respiration based on optimal leaf carbon when the amount of carbon in the carbohydrate storage pool is less than (GTMRSPLAI(3) \* optimal leaf carbon) for a grasstree system |  |  |
| GTMRSPLAI(3) | X2 value for line function that decreases maintenance respiration based on optimal leaf carbon when the amount of carbon in the carbohydrate storage pool is less than (GTMRSPLAI(3) \* optimal leaf carbon) for a grasstree system  OR  X1 value for line function that decreases maintenance respiration based on optimal leaf carbon when the amount of carbon in the carbohydrate storage pool is between (GTMRSPLAI(3) \* optimal leaf carbon) and (GTMRSPLAI(5) \* optimal leaf carbon) for a grasstree system |  |  |
| GTMRSPLAI(4) | Y2 value for line function that decreases maintenance respiration based on optimal leaf carbon when the amount of carbon in the carbohydrate storage pool is less than (GTMRSPLAI(3) \* optimal leaf carbon) for a grasstree system  OR  Y1 value for line function that decreases maintenance respiration based on optimal leaf carbon when the amount of carbon in the carbohydrate storage pool is between (GTMRSPLAI(3) \* optimal leaf carbon) and (GTMRSPLAI(5) \* optimal leaf carbon) for a grasstree system |  |  |
| GTMRSPLAI(5) | X2 value for line function that decreases maintenance respiration based on optimal leaf carbon when the amount of carbon in the carbohydrate storage pool is between (GTMRSPLAI(3) \* optimal leaf carbon) and (GTMRSPLAI(5) \* optimal leaf carbon) for a grasstree system |  |  |
| GTMRSPLAI(6) | Y2 value for line function that decreases maintenance respiration based on optimal leaf carbon when the amount of carbon in the carbohydrate storage pool is between (GTMRSPLAI(3) \* optimal leaf carbon) and (GTMRSPLAI(5) \* optimal leaf carbon) for a grasstree system  OR  Y value for line function that decreases maintenance respiration based on optimal leaf carbon when the amount of carbon in the carbohydrate storage pool is greater than (GTMRSPLAI(5) \* optimal leaf carbon) for a grasstree system |  |  |
|  |  |  |  |
| **GTGRESP(\*)** | **Growth respiration is a fixed fraction of new production. See subroutine grasstreegrow.** |  |  |
| GTGRESP(1) | Maximum fraction of live leaf production that goes to growth respiration for grasstrees | fraction | 0.0 – 1.0 |
| GTGRESP(2) | Maximum fraction of live stem production that goes to growth respiration for grasstrees | fraction | 0.0 – 1.0 |
| GTGRESP(3) | Maximum fraction of live juvenile fine root production that goes to growth respiration for grasstrees | fraction | 0.0 – 1.0 |
| GTGRESP(4) | Maximum fraction of live coarse root production that goes to growth respiration for grasstrees | fraction | 0.0 – 1.0 |
| GTGRESP(5) | Maximum fraction of live mature fine root production that goes to growth respiration for grasstrees | fraction | 0.0 – 1.0 |

Table . Parameters in grasstree.100 controlling photosynthesis (GPP)

| **Photosynthesis model** |  |  |  |
| --- | --- | --- | --- |
| CARBOSTG(3,1) | Initial value for unlabeled carbohydrate storage pool | g C m-2 | 0 – 250 |
| CARBOSTG(3,2) | Initial value for labeled carbohydrate storage pool | g C m-2 | 0 –1 |
| AMAX(3) | Maximum net CO2 assimilation rate assuming maximum possible PAR, all intercepted, no temperature, water or vapor pressure deficit stress. | nmol CO2 g‑1 (leaf biomass) sec-1 |  |
| AMAXFRAC(3) | Average daily maximum photosynthesis as a fraction of AMAX(3). | fraction | 0.0 – 1.0 |
| AMAXSCALAR1(3) | Multiplier used to adjust aMax based on growthDays1 days since germination | scalar |  |
| AMAXSCALAR2(3) | Multiplier used to adjust aMax based on growthDays2 days since germination. | scalar | 0.8 – 1.6 |
| AMAXSCALAR3(3) | Multiplier used to adjust aMax based on growthDays3 days since germination. | scalar | 0.7 – 1.5 |
| AMAXSCALAR4(3) | Multiplier used to adjust aMax based on growthDays4 days since germination. | scalar | 0.3 – 0.8 |
| ATTENUATION(3) | Light attenuation coefficient. |  |  |
| BASEFOLRESPFRAC(3) | Basal foliage respiration rate, as fraction of maximum net photosynthesis rate  respPerGram = base foliar respiration, unmodified by temp, water, light, vpd (nmol CO2 (g leaf)-1 sec-1):  respPerGram = **basefolrespfrac** \* **amax**  \* **amaxscalar**  grossAMax = **amax** \* **amaxscalar** \* **amaxfrac**  + respPerGram |  |  |
| CFRACLEAF(3) | factor for converting leaf biomass to carbon (leaf biomass \* cFracLeaf = leaf C)  See equation in **leafcspwt(3)** definition. | (g C) / (g biomass) |  |
| DVPDEXP(3) | Exponential value in vapor pressure deficit effect on photosynthesis equation.  dVpd = dVpdSlope \* exp(vpd\*dVpdExp) |  |  |
| DVPDSLOPE(3) | Slope value in vapor pressure deficit effect on photosynthesis equation.  dVpd = dVpdSlope \* exp(vpd\*dVpdExp) |  |  |
| GROWTHDAYS1(3) | Number of days after germination to start using AMAXSCALAR1. | number of days |  |
| GROWTHDAYS2(3) | Number of days after germination to start using AMAXSCALAR2. | number of days |  |
| GROWTHDAYS3(3) | Number of days after germination to start using AMAXSCALAR3. | number of days |  |
| GROWTHDAYS4(3) | Number of days after germination to start using AMAXSCALAR4. | number of days |  |
| HALFSATPAR(3) | Photosynthetically active radiation (PAR) at which photosynthesis occurs at 1/2 of theoretical maximum. | Einsteins \* m-2 ground area \* day-1 |  |
| LEAFCSPWT(3) | Grams of carbon in a square meter of leaf area.  convert units from nmol CO2 (g leaf)-1 sec-1 to gC m-2 day-1  conversion = 12 \* (10-9)  \* (**leafcspwt** / **cfracleaf**)  \* lai \* SEC\_PER\_DAY;  potGrossPsn = grossAMax  \* dTemp \* dVpd \* lightEff  \* conversion | g C (m2 leaf area)-1 |  |
| PSNTMIN(3) | Minimum temperature at which net photosynthesis occurs | ºC |  |
| PSNTOPT(3) | Optimal temperature at which net photosynthesis occurs | ºC |  |