Supplement of Geosci. Model Dev., 8, 1461–1471, 2015
http://www.geosci-model-dev.net/8/1461/2015/
doi:10.5194/gmd-8-1461-2015-supplement

Geoscientific
Model Development
Open Access

Supplement of

ORCHIDEE-SRC v1.0: an extension of the land surface model ORCHIDEE for simulating short rotation coppice poplar plantations

T. De Groote et al.

Correspondence to: T. De Groote (toon.degroote@uantwerpen.be)

Supplement

Legend:

Management modifications

🟦 coppicing of the trees

🟦 start with cuttings instead of sapplings

Growth modifications

🟦 update allometric relations

🟪 multi-stem resprouting

🟨 decoupling of root-leaf growth

Allocation modifications

🟥 no fruit production if the wood is younger than five years

🟫 no heartwood production if the wood is younger than five years

🟩 reach LAI$_{max}$ from year two onwards

🟩 maintain root-shoot ratio

Parameterization

🟪 parameterization

Code modifications:

```
Index: src_stomate/lpj_establish.f90
===================================================================
--- src_stomate/lpj_establish.f90  (revision 561)
+++ src_stomate/lpj_establish.f90  (working copy)
@@ -34,6 +34,9 @@
        precip_annual, gdd0, lm_lastyearmax, &
        cn_ind, lai, avail_tree, avail_grass, &
        leaf_age, leaf_frac, &
+       root_age, root_frac, &
        ind, biomass, age, everywhere, co2_to_bm,veget_max)

     !
@@ -82,6 +85,12 @@
     REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)  :: leaf_age
     ! fraction of leaves in leaf age class
     REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)  :: leaf_frac
+    ! root age (days)
+    REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)  :: root_age
+    ! fraction of roots in root age class
+    REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)  :: root_frac
     ! Number of individuals / m2
     REAL(r_std), DIMENSION(npts,nvm), INTENT(inout)            :: ind
     ! biomass (gC/(m**2 of ground))
@@ -454,6 +463,10 @@
              WHERE ( many_new(:) )
                 leaf_age(:,j,1) = 0.0
                 leaf_frac(:,j,1) = 1.0
```

```
+                root_age(:,j,1) = 0.0
+                root_frac(:,j,1) = 1.0
             ENDWHERE

             DO m = 2, nleafages
@@ -461,6 +474,10 @@
                WHERE ( many_new(:) )
                   leaf_age(:,j,m) = 0.0
                   leaf_frac(:,j,m) = 0.0
+                  root_age(:,j,m) = 0.0
+                  root_frac(:,j,m) = 0.0
                ENDWHERE

             ENDDO
Index: src_stomate/stomate_alloc.f90
===================================================================
--- src_stomate/stomate_alloc.f90  (revision 561)
+++ src_stomate/stomate_alloc.f90  (working copy)
@@ -37,7 +37,11 @@
   SUBROUTINE alloc (npts, dt, &
        lai, veget_max, senescence, when_growthinit, &
        moiavail_week, tsoil_month, soilhum_month, &
-       biomass, age, leaf_age, leaf_frac, rprof, f_alloc, &
+       biomass, age, leaf_age, leaf_frac, &
+       root_age, root_frac, &
+       rprof, f_alloc, &
        PFTpresent, height, bm_alloc, delta_bm, sapwood_age, &
        ls_longterm, EndOfYear, sum_sapconv, sum_sapalloc, &
        biomass_lastyear,lr_longterm,sum_rootm,lai_max_calc, &
@@ -97,6 +101,12 @@
     REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout) :: leaf_age
     ! fraction of leaves in leaf age class
     REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout) :: leaf_frac
+    ! root age (days)
+    REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout) :: root_age
+    ! fraction of roots in leaf age class
+    REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout) :: root_frac

     ! 0.3 output

@@ -186,6 +196,12 @@
     REAL(r_std), DIMENSION(npts)                               :: leaf_mass_young
     ! old leaf biomass (gC/m**2)
     REAL(r_std), DIMENSION(npts,nvm)                          :: lm_old
+    ! mass in youngest root age class (gC/m**2)
+    REAL(r_std), DIMENSION(npts)                               :: root_mass_young
+    ! old root biomass (gC/m**2)
+    REAL(r_std), DIMENSION(npts,nvm)                          :: rm_old
     ! maximum time (d) during which reserve is used
     REAL(r_std)                                                :: reserve_time
     ! lai on natural part of the grid cell, or of this agricultural PFT
@@ -464,6 +480,11 @@

     lm_old(:,:) = biomass(:,:,ileaf)

+    ! save old root mass
+    rm_old(:,:) = biomass(:,:,iroot)
+
     DO j = 2, nvm

        !
@@ -552,8 +586,63 @@

        ENDDO

+       !
+       ! 2.3 update root age
+       !
+
+       ! 2.3.1 Decrease root age in youngest class.
+
+       root_mass_young(:) = root_frac(:,j,1) * rm_old(:,j) + ( use_reserve(:) - transloc_leaf(:) )
+
+       WHERE ( ( ( use_reserve(:) - transloc_leaf(:) ) .GT. min_stomate ) .AND. ( root_mass_young(:) .GT. min_stomate ) )
+
+          root_age(:,j,1) = MAX( zero, root_age(:,j,1) * ( root_mass_young(:) - ( use_reserve(:) - transloc_leaf(:)) ) / &
+               root_mass_young(:) )
+
+       ENDWHERE
+
+       ! 2.3.2 new age class fractions (fraction in youngest class increases)
+
+       ! 2.3.2.1 youngest class: new mass in youngest class divided by total new mass
+
+       WHERE ( biomass(:,j,iroot) .GT. min_stomate )
+
+          root_frac(:,j,1) = root_mass_young(:) / biomass(:,j,iroot)
+
+       ENDWHERE
+
+       ! 2.3.2.2 other classes: old mass in root age class divided by new mass
+
+       DO m = 2, nleafages
+
+          WHERE ( biomass(:,j,iroot) .GT. min_stomate )
+
+             root_frac(:,j,m) = root_frac(:,j,m) * rm_old(:,j) / biomass(:,j,iroot)
+
+          ENDWHERE
+
+       ENDDO
+
     ENDDO         ! loop over PFTs

     !
     ! 3 Calculate fractional allocation.
     !   The fractions of NPP allocated to the different compartments depend on the
@@ -596,12 +685,12 @@
          ENDWHERE

          ! Disturbance-type allocation : if root/shoot ratio is very unbalanced, allocation helps to rebalance quickly
-         WHERE (forest_managed(:,j)>0 .AND. &
+         WHERE (three4 == .TRUE. .AND. forest_managed(:,j)>0 .AND. &
            SUM(biomass(:,j,(/isapabove,iheartabove/)))/SUM(biomass(:,j,(/isapabove,iheartabove,isapbelow,iheartbelow/))) > &
               1.1*alloc_sap_above(:))
             alloc_sap_above (:) = 0.05

-         ELSEWHERE (forest_managed(:,j) > 0 .AND. &
+         ELSEWHERE (three4 == .TRUE. .AND. forest_managed(:,j) > 0 .AND. &
           SUM(biomass(:,j,(/isapabove,iheartabove/)))/SUM(biomass(:,j,(/isapabove,iheartabove,isapbelow,iheartbelow/))) < &
              0.9*alloc_sap_above(:))
             alloc_sap_above (:) = 0.95
@@ -734,7 +823,7 @@

 ! VBADD
                 ! After coppicing, no allocation to sexual reproduction for 5 years (Pontaillier et al 1999)
-                IF (forest_managed(i,j) == 4 .AND. last_cut(i,j) <=5) THEN
+                IF (forest_managed(i,j) == 4 .AND. last_cut(i,j) <=5 .AND. three1 == .TRUE.) THEN
                    f_alloc(i,j,ifruit) = 0.
                 ENDIF
 ! ENDVBADD
@@ -749,20 +838,28 @@
 !!$            IF ( ( biomass(i,j,icarbres)*sla(j) ) .LT. 2*lai_max_calc(i,j) ) THEN
 !ENDVBMODIF
 !VBADD
-               IF (forest_managed(i,j) == 4) THEN
-                  carb_rescale(i) = 1. / ( 1. + 1.5 *ecureuil(j) * ( LtoLSR(i) + RtoLSR(i) ) )
-               ELSE
!ENDVBADD
                   carb_rescale(i) = 1. / ( 1. + ecureuil(j) * ( LtoLSR(i) + RtoLSR(i) ) )
 !VBADD
-               ENDIF
!ENDVBADD

                 ELSE
                    carb_rescale(i) = 1.
                 ENDIF

                 f_alloc(i,j,ileaf) = LtoLSR(i) * ( 1.-f_alloc(i,j,ifruit) ) * carb_rescale(i)

                 f_alloc(i,j,isapabove) = StoLSR(i) * alloc_sap_above(i) * &
Index: src_stomate/stomate_npp.f90
===================================================================
--- src_stomate/stomate_npp.f90    (revision 561)
+++ src_stomate/stomate_npp.f90    (working copy)
@@ -36,7 +36,11 @@
        PFTpresent, &
        tlong_ref, t2m, tsoil, lai, rprof, &
        gpp, f_alloc, bm_alloc, resp_maint_part,&
-       biomass, leaf_age, leaf_frac, age, &
+       biomass, leaf_age, leaf_frac, &
+       root_age, root_frac, &
+       age, &
        resp_maint, resp_growth, npp)

     !
@@ -76,6 +80,12 @@
     REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)  :: leaf_age
     ! fraction of leaves in leaf age class
     REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)  :: leaf_frac
+    ! root age (days)
+    REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)  :: root_age
+    ! fraction of roots in leaf age class
+    REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)  :: root_frac
     ! age (years)
     REAL(r_std), DIMENSION(npts,nvm), INTENT(inout)             :: age

@@ -128,6 +138,12 @@
     REAL(r_std), DIMENSION(npts,nvm)                            :: leaf_mass_young
     ! leaf mass after maintenance respiration
     REAL(r_std), DIMENSION(npts,nvm)                            :: lm_old
+    ! root mass in youngest age class (gC/m**2 of ground)
+    REAL(r_std), DIMENSION(npts,nvm)                            :: root_mass_young
+    ! root mass after maintenance respiration
+    REAL(r_std), DIMENSION(npts,nvm)                            :: rm_old
     ! biomass created when biomass<0 because of dark respiration (gC/m**2 of ground)
     REAL(r_std), DIMENSION(npts,nvm)                            :: bm_create
     ! maximum part of allocatable biomass used for respiration
@@ -376,8 +405,14 @@
     !

     lm_old(:,:) = biomass(:,:,ileaf)
+    rm_old(:,:) = biomass(:,:,iroot)

     biomass(:,:,:) = biomass(:,:,:) + bm_alloc(:,:,:)

!BEGINNVADD
     npp_part (:,:,:)=  npp_part (:,:,:) + bm_alloc(:,:,:)
 !ENDNVADD
@@ -469,7 +504,61 @@

     ENDDO

+    !
+    ! 7.bis root age
+    !
+
+    !
+    ! 7..bis.1 Decrease root age in youngest class if new root biomass is higher than old one.
+    !
+
+    DO j = 2,nvm
+       root_mass_young(:,j) = root_frac(:,j,1) * rm_old(:,j) + bm_alloc(:,j,iroot)
+    ENDDO
+
+    DO j = 2,nvm
+       WHERE ( ( bm_alloc(:,j,iroot) .GT. zero ) .AND. &
+         ( root_mass_young(:,j) .GT. zero ) )
+
+          root_age(:,j,1) = MAX ( zero, &
+                & root_age(:,j,1) * &
+                & ( root_mass_young(:,j) - bm_alloc(:,j,iroot) ) / &
+                & root_mass_young(:,j) )
+
+       ENDWHERE
+    ENDDO
+
+    !
+    ! 7.bis.2 new age class fractions (fraction in youngest class increases)
+    !
+
+    ! 7.bis.2.1 youngest class: new mass in youngest class divided by total new mass
+
+    DO j = 2,nvm
+       WHERE ( biomass(:,j,iroot) .GT. min_stomate )
+
+          root_frac(:,j,1) = root_mass_young(:,j) / biomass(:,j,iroot)
+
+       ENDWHERE
+    ENDDO
+
+    ! 7.bis.2.2 other classes: old mass in root age class divided by new mass
+
+    DO m = 2, nleafages
+
+       DO j = 2,nvm
+          WHERE ( biomass(:,j,iroot) .GT. min_stomate )
+
+             root_frac(:,j,m) = root_frac(:,j,m) * rm_old(:,j) / biomass(:,j,iroot)
+
+          ENDWHERE
+       ENDDO
+
+    ENDDO
+
     !
     ! 8 Plant age (years)
     !

Index: src_stomate/stomate_phenology.f90
===================================================================
--- src_stomate/stomate_phenology.f90    (revision 561)
+++ src_stomate/stomate_phenology.f90    (working copy)
@@ -44,6 +44,9 @@
        gdd_m5_dormance, gdd_midwinter, ncd_dormance, ngd_minus5, &
        senescence, time_lowgpp, time_hum_min, &
        biomass, leaf_frac, leaf_age, &
+       root_age, root_frac, &
        when_growthinit, co2_to_bm, lai)
     !
@@ -99,6 +102,12 @@
     REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)  :: leaf_frac
     ! leaf age (days)
     REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)  :: leaf_age
+    ! fraction of roots in leaf age class
+    REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)  :: root_frac
+    ! root age (days)
+    REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)  :: root_age
     ! how many days ago was the beginning of the growing season
     REAL(r_std), DIMENSION(npts,nvm), INTENT(inout)            :: when_growthinit
     ! co2 taken up (gC/(m**2 of total ground)/day)
@@ -347,10 +356,16 @@
 
        WHERE ( age_reset(:) )
          leaf_frac(:,j,1) = 1.0
+          root_frac(:,j,1) = 1.0
       ENDWHERE
       DO m = 2, nleafages
          WHERE ( age_reset(:) )
            leaf_frac(:,j,m) = 0.0
+             root_frac(:,j,m) = 0.0
         ENDWHERE
      ENDDO
 
@@ -359,6 +374,9 @@
       DO m = 1, nleafages
          WHERE ( age_reset(:) )
            leaf_age(:,j,m) = 0.0
+             root_age(:,j,m) = 0.0
        ENDWHERE
     ENDDO
 
Index: src_stomate/stomate_lcchange.f90
===================================================================
--- src_stomate/stomate_lcchange.f90     (revision 561)
+++ src_stomate/stomate_lcchange.f90     (working copy)
@@ -36,7 +36,9 @@
 !!$,prod10_total,prod100_total,&
        convflux,&
 !!$,cflux_prod_total,
-       cflux_prod10,cflux_prod100, leaf_frac,&
+       cflux_prod10,cflux_prod100, leaf_frac, root_frac,&
       npp_longterm, lm_lastyearmax, litter, carbon)
     IMPLICIT NONE
@@ -109,6 +111,8 @@
     REAL(r_std), DIMENSION(npts,10), INTENT(inout)                     :: flux10
     REAL(r_std), DIMENSION(npts,100), INTENT(inout)                    :: flux100

+    ! fraction of roots in root age class
+    REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)         :: root_frac
     ! fraction of leaves in leaf age class
     REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)         :: leaf_frac

@@ -216,6 +220,9 @@

                 when_growthinit(i,j) = large_value
                 leaf_frac(i,j,1) = 1.0
+                root_frac(i,j,1) = 1.0
                 npp_longterm(i,j) = 10.
                 lm_lastyearmax(i,j) = bm_sapl(j,ileaf) * ind(i,j)
              ENDIF
Index: src_stomate/lpj_kill.f90
===================================================================
--- src_stomate/lpj_kill.f90 (revision 561)
+++ src_stomate/lpj_kill.f90 (working copy)
@@ -24,6 +24,9 @@
   SUBROUTINE kill (npts, whichroutine, lm_lastyearmax, &
        ind, PFTpresent, cn_ind, biomass, senescence, RIP_time, &
        lai, age, leaf_age, leaf_frac, &
+       root_age, root_frac, &
        when_growthinit, everywhere, veget, veget_max, bm_to_litter)

     !
@@ -61,6 +64,12 @@
     REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout) :: leaf_age
     ! fraction of leaves in leaf age class
     REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout) :: leaf_frac
+    ! root age (days)
+    REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout) :: root_age
+    ! fraction of roots in leaf age class
+    REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout) :: root_frac
     ! how many days ago was the beginning of the growing season
     REAL(r_std), DIMENSION(npts,nvm), INTENT(inout)           :: when_growthinit
     ! is the PFT everywhere in the grid box or very localized (after its introduction)
@@ -157,6 +166,10 @@

                    leaf_age(:,j,m) = 0.0
                    leaf_frac(:,j,m) = 0.0
```

```
+                   root_age(:,j,m) = 0.0
+                   root_frac(:,j,m) = 0.0

                 ENDWHERE

Index: src_stomate/stomate_io.f90
===================================================================
--- src_stomate/stomate_io.f90     (revision 561)
+++ src_stomate/stomate_io.f90     (working copy)
@@ -47,7 +47,11 @@
         &  PFTpresent, npp_longterm, lm_lastyearmax, lm_thisyearmax, &
         &  maxfpc_lastyear, maxfpc_thisyear, &
         &  turnover_longterm, gpp_week, biomass, resp_maint_part, &
-        &  leaf_age, leaf_frac, senescence, when_growthinit, age, &
+        &  leaf_age, leaf_frac, &
+        &  root_age, root_frac, &
+        &  senescence, when_growthinit, age, &
         &  resp_hetero, resp_maint, resp_growth, co2_fire, co2_to_bm_dgvm, &
         &  veget_lastlight, everywhere, need_adjacent, RIP_time, &
         &  time_lowgpp, time_hum_min, hum_min_dormance, &
@@ -196,6 +200,12 @@
     REAL(r_std),DIMENSION(npts,nvm,nleafages),INTENT(out) :: leaf_age
     ! fraction of leaves in leaf age class
     REAL(r_std),DIMENSION(npts,nvm,nleafages),INTENT(out) :: leaf_frac
+    ! root age (days)
+    REAL(r_std),DIMENSION(npts,nvm,nleafages),INTENT(out) :: root_age
+    ! fraction of roots in leaf age class
+    REAL(r_std),DIMENSION(npts,nvm,nleafages),INTENT(out) :: root_frac
     ! is the plant senescent ?
     !(only for deciduous trees - carbohydrate reserve)
     LOGICAL,DIMENSION(npts,nvm),INTENT(out) :: senescence
@@ -869,6 +879,28 @@
        IF (ALL(leaf_frac(:,:,m) == val_exp)) leaf_frac(:,:,m) = zero
     ENDDO
     !-
+    root_age(:,:,:) = val_exp
+    DO m=1,nleafages
+       WRITE (part_str,'(I2)') m
+       IF ( m < 10 ) part_str(1:1) = '0'
+       var_name = 'root_age_'//part_str(1:LEN_TRIM(part_str))
+       CALL restget_p (rest_id_stomate, var_name, nbp_glo, nvm  , 1, itime, &
+             &                   .TRUE., root_age(:,:,m), 'gather', nbp_glo, index_g)
+       IF (ALL(root_age(:,:,m) == val_exp)) root_age(:,:,m) = zero
+    ENDDO
+    !-
+    root_frac(:,:,:) = val_exp
+    DO m=1,nleafages
+       WRITE(part_str,'(I2)') m
+       IF ( m < 10 ) part_str(1:1) = '0'
+       var_name = 'root_frac_'//part_str(1:LEN_TRIM(part_str))
+       CALL restget_p (rest_id_stomate, var_name, nbp_glo, nvm  , 1, itime, &
+            &                  .TRUE., root_frac(:,:,m), 'gather', nbp_glo, index_g)
+       IF (ALL(root_frac(:,:,m) == val_exp)) root_frac(:,:,m) = zero
+    ENDDO
+    !-
     senescence_real(:,:) = val_exp
     var_name = 'senescence'
     CALL restget_p (rest_id_stomate, var_name, nbp_glo, nvm  , 1, itime, &
@@ -1227,7 +1259,11 @@
        &  PFTpresent, npp_longterm, lm_lastyearmax, lm_thisyearmax, &
        &  maxfpc_lastyear, maxfpc_thisyear, &
        &  turnover_longterm, gpp_week, biomass, resp_maint_part, &
-       &  leaf_age, leaf_frac, senescence, when_growthinit, age, &
+       &  leaf_age, leaf_frac, &
+       &  root_age, root_frac, &
+       &  senescence, when_growthinit, age, &
       &  resp_hetero, resp_maint, resp_growth, co2_fire, co2_to_bm_dgvm, &
       &  veget_lastlight, everywhere, need_adjacent, RIP_time, &
       &  time_lowgpp, time_hum_min, hum_min_dormance, &
@@ -1367,6 +1403,12 @@
     REAL(r_std),DIMENSION(npts,nvm,nleafages),INTENT(in) :: leaf_age
     ! fraction of leaves in leaf age class
     REAL(r_std),DIMENSION(npts,nvm,nleafages),INTENT(in) :: leaf_frac
+    ! root age (days)
+    REAL(r_std),DIMENSION(npts,nvm,nleafages),INTENT(in) :: root_age
+    ! fraction of roots in leaf age class
+    REAL(r_std),DIMENSION(npts,nvm,nleafages),INTENT(in) :: root_frac
     ! is the plant senescent ?
     ! (only for deciduous trees - carbohydrate reserve)
     LOGICAL,DIMENSION(npts,nvm),INTENT(in) :: senescence
@@ -1831,6 +1873,24 @@
             &                   leaf_frac(:,:,m), 'scatter', nbp_glo, index_g)
     ENDDO
     !-
+    DO m=1,nleafages
+       WRITE(part_str,'(I2)') m
+       IF (m < 10) part_str(1:1) = '0'
+       var_name = 'root_age_'//part_str(1:LEN_TRIM(part_str))
+       CALL restput_p (rest_id_stomate, var_name, nbp_glo, nvm, 1, itime, &
+            &                  root_age(:,:,m), 'scatter', nbp_glo, index_g)
+    ENDDO
+    !-
+    DO m=1,nleafages
+       WRITE(part_str,'(I2)') m
+       IF (m < 10) part_str(1:1) = '0'
+       var_name = 'root_frac_'//part_str(1:LEN_TRIM(part_str))
+       CALL restput_p (rest_id_stomate, var_name, nbp_glo, nvm, 1, itime, &
+            &                   root_frac(:,:,m), 'scatter', nbp_glo, index_g)
+    ENDDO
+    !-
     var_name = 'senescence'
     WHERE ( senescence(:,:) )
        senescence_real = un
Index: src_stomate/stomate_data.f90
===================================================================
--- src_stomate/stomate_data.f90   (revision 561)
+++ src_stomate/stomate_data.f90   (working copy)
@@ -105,8 +105,10 @@

        ENDIF
 !VBADD
-      ! From Cafapietra 2005, average
-      IF (management == 4 .AND. j .EQ. 6) sla(j)= 171.5/260. * 2. * ( 10. ** ( 2.41 - 0.38 * LOG10(12./leaflife_tab(j)) ) )
   *1e-4
+  ! From POPFULL measurements, SLA = 0.025 mÂ²/gC (= 0.026 from unadapted equation)
+
 !!$      IF ( leaf_tab(j) .EQ. 1 ) THEN
 !!$
 !!$        ! broad leaved tree
@@ -139,8 +141,26 @@

           ! 5.1 trees
+
+           alpha = alpha_tree
+
+          bm_sapl(j,:) = 0.0
+          ! 20 cmÂ³ = 20e-6 mÂ³ => 20e-6 mÂ³ * 125397 gC/mÂ³ = 2.5 gC
+          ! 2% reserves and 98% sapwood
+          bm_sapl(j,icarbres) = 2.5*0.5
+          bm_sapl(j,isapabove) = 2.5*0.5
+
+          ELSE

          alpha = alpha_tree
```

```
           bm_sapl(j,ileaf) = &
                ( (4.*pipe_tune1 * ( x*4.*sla(j)/(pi*pipe_k1))**.8 ) / sla(j) ) ** 5.

@@ -162,6 +182,10 @@
           bm_sapl(j,iheartabove) = 2. * bm_sapl(j,isapabove)
           bm_sapl(j,iheartbelow) = 2. * bm_sapl(j,isapbelow)

+!TDG
+          ENDIF
+!endTDG
+
        ELSE

           ! 5.2 grasses
@@ -436,7 +460,17 @@

        IF ( ( bavard .GE. 1 ) .AND. ( pheno_crit%leafagecrit(j) .NE. undef ) ) &
             WRITE(numout,*) '       critical leaf age (d):', pheno_crit%leafagecrit(j)
+       !
+       ! 15.bis maximum lifetime of roots
+       !

+       pheno_crit%rootagecrit(j) = rootagecrit_tab(j)
+
+       IF ( ( bavard .GE. 1 ) .AND. ( pheno_crit%rootagecrit(j) .NE. undef ) ) &
+            WRITE(numout,*) '       critical root age (d):', pheno_crit%rootagecrit(j)
+
        !
        ! 16 time constant for leaf age discretisation (d)
        !
@@ -446,7 +480,18 @@
        IF ( bavard .GE. 1 ) &
             WRITE(numout,*) '       time constant for leaf age discretisation (d):', &
             leaf_timecst(j)
+       !
+       ! 16.bis time constant for root age discretisation (d)
+       !

+       root_timecst(j) = pheno_crit%rootagecrit(j) / REAL( nleafages,r_std )
+
+       IF ( bavard .GE. 1 ) &
+            WRITE(numout,*) '       time constant for root age discretisation (d):', &
+            root_timecst(j)
+
        !
        ! 17 minimum lai, initial
        !
Index: src_stomate/stomate_vmax.f90
===================================================================
--- src_stomate/stomate_vmax.f90   (revision 561)
+++ src_stomate/stomate_vmax.f90   (working copy)
@@ -30,6 +30,9 @@

   SUBROUTINE vmax (npts, dt, &
        leaf_age, leaf_frac, &
+       root_age, root_frac, &
!VBADD
        forest_managed,age_stand, &
 !ENDVBADD
@@ -58,6 +61,12 @@
     REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)  :: leaf_age
     ! fraction of leaves in leaf age class
     REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)  :: leaf_frac
+    ! root age (days)
+    REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)  :: root_age
+    ! fraction of roots in leaf age class
+    REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)  :: root_frac

     ! 0.3 output

@@ -82,6 +91,12 @@
     REAL(r_std), DIMENSION(npts,nvm,nleafages)                 :: d_leaf_frac
     ! new leaf age (d)
     REAL(r_std), DIMENSION(npts,nleafages)                      :: leaf_age_new
+    ! change of fraction of roots in age class
+    REAL(r_std), DIMENSION(npts,nvm,nleafages)                 :: d_root_frac
+    ! new root age (d)
+    REAL(r_std), DIMENSION(npts,nleafages)                      :: root_age_new
     ! sum of leaf age fractions, for normalization
     REAL(r_std), DIMENSION(npts)                                :: sumfrac
     ! relative leaf age (age/critical age)
@@ -111,6 +126,10 @@
     ! Temporary variable for sensitivity study
     REAL(r_std), SAVE        :: ss_decl_max
 !!ENDVBADD
+    ! SRC specific vcmax_opt and vjmax_opt
+    REAL(r_std), SAVE                        :: vcmax_src, vjmax_src
     ! =========================================================================

     IF (bavard.GE.3) WRITE(numout,*) 'Entering vmax'
```

```
@@ -136,12 +155,21 @@
        CALL getin_p ('opt_factor_9',opt_factor(9))
 !ENDVBADD

+        vcmax_src = 150 !79.1
+        vjmax_src = 200 !152.4
+        CALL getin_p ('vcmax_src',vcmax_src)
+        CALL getin_p ('vjmax_src',vjmax_src)
+
        WRITE(numout,*) 'vmax:'

        WRITE(numout,*) '   > offset (minimum vcmax/vmax_opt):' , vmax_offset
        WRITE(numout,*) '   > relative leaf age at which vmax attains vcmax_opt:', leafage_firstmax
        WRITE(numout,*) '   > relative leaf age at which vmax falls below vcmax_opt:', leafage_lastmax
        WRITE(numout,*) '   > relative leaf age at which vmax attains its minimum:', leafage_old
+       WRITE(numout,*) '   > vcmax_opt SRC override:', vcmax_src
+       WRITE(numout,*) '   > vjmax_opt SRC override:', vjmax_src

        firstcall = .FALSE.

@@ -173,6 +201,13 @@
              leaf_age(:,j,m) = leaf_age(:,j,m) + dt

           ENDWHERE
+          WHERE ( root_frac(:,j,m) .GT. min_stomate )
+
+             root_age(:,j,m) = root_age(:,j,m) + dt
+
+          ENDWHERE
        ENDDO

     ENDDO
@@ -188,11 +223,17 @@

        ! nothing goes into first age class
        d_leaf_frac(:,j,1) = zero
+       d_root_frac(:,j,1) = zero

        ! from m-1 to m
        DO m = 2, nleafages

           d_leaf_frac(:,j,m) = leaf_frac(:,j,m-1) * dt/leaf_timecst(j)
+          d_root_frac(:,j,m) = root_frac(:,j,m-1) * dt/root_timecst(j)

        ENDDO
```

```
@@ -201,6 +242,9 @@
        !                 new fraction

        leaf_age_new(:,:) = zero
+       root_age_new(:,:) = zero

        DO m = 2, nleafages-1
           !      DO m=2, nleafages
@@ -216,7 +260,15 @@
              !                               ( leaf_frac(:,j,m) + d_leaf_frac(:,j,m) )

           ENDWHERE
+          WHERE ( d_root_frac(:,j,m) .GT. min_stomate )

+             root_age_new(:,m) = ( ( (root_frac(:,j,m)- d_root_frac(:,j,m+1)) * root_age(:,j,m) )  + &
+                  ( d_root_frac(:,j,m) * root_age(:,j,m-1) ) ) / &
+                  ( root_frac(:,j,m) + d_root_frac(:,j,m)- d_root_frac(:,j,m+1) )
+
+          ENDWHERE
        ENDDO       ! Loop over age classes

        WHERE ( d_leaf_frac(:,j,nleafages) .GT. min_stomate )
@@ -226,7 +278,15 @@
                ( leaf_frac(:,j,nleafages) + d_leaf_frac(:,j,nleafages) )

        ENDWHERE
+       WHERE ( d_root_frac(:,j,nleafages) .GT. min_stomate )

+          root_age_new(:,nleafages) = ( ( root_frac(:,j,nleafages) * root_age(:,j,nleafages) )  + &
+               ( d_root_frac(:,j,nleafages) * root_age(:,j,nleafages-1) ) ) / &
+               ( root_frac(:,j,nleafages) + d_root_frac(:,j,nleafages) )
+
+       ENDWHERE
        DO m = 2, nleafages

           WHERE ( d_leaf_frac(:,j,m) .GT. min_stomate )
@@ -234,7 +294,13 @@
              leaf_age(:,j,m) = leaf_age_new(:,m)

           ENDWHERE
+          WHERE ( d_root_frac(:,j,m) .GT. min_stomate )

+             root_age(:,j,m) = root_age_new(:,m)
+
+          ENDWHERE
        ENDDO       ! Loop over age classes

        ! 2.2.3 calculate new fraction
@@ -246,7 +312,13 @@

           ! where it goes to
           leaf_frac(:,j,m) = leaf_frac(:,j,m) + d_leaf_frac(:,j,m)
+          ! where the change comes from
+          root_frac(:,j,m-1) = root_frac(:,j,m-1) - d_root_frac(:,j,m)

+          ! where it goes to
+          root_frac(:,j,m) = root_frac(:,j,m) + d_root_frac(:,j,m)
        ENDDO

        ! 2.2.4 renormalize fractions in order to prevent accumulation
@@ -256,6 +328,9 @@

        DO m = 1, nleafages
           leaf_frac(:,j,m) = MAX( zero, leaf_frac(:,j,m) )
+          root_frac(:,j,m) = MAX( zero, root_frac(:,j,m) )
        ENDDO

        ! total of fractions, should be very close to one where there is leaf mass
@@ -283,9 +358,40 @@
           ENDWHERE

        ENDDO
+       ! total of fractions, should be very close to one where there is leaf mass

+       sumfrac(:) = zero
+
+       DO m = 1, nleafages
+
+          sumfrac(:) = sumfrac(:) + root_frac(:,j,m)
+
+       ENDDO
+
+       ! normalize
+
+       DO m = 1, nleafages
+
+          WHERE ( sumfrac(:) .GT. min_stomate )
+
+             root_frac(:,j,m) = root_frac(:,j,m) / sumfrac(:)
```

```
+
+          ELSEWHERE
+
+             root_frac(:,j,m) = zero
+
+          ENDWHERE
+
+       ENDDO
+
     ENDDO          ! Loop over PFTs

+print *,'root_age: ',root_age(:,:,:)
+print *,'root_frac: ',root_frac(:,:,:)
     !
     ! 3 calculate vmax as a function of the age of leaves
     !
@@ -335,8 +441,12 @@
                 IF (forest_managed(i,j)>0) THEN
                    ! If TCR, values from Calfapietra 2005
                    IF (forest_managed(i,j) == 4 .AND. j == 6) THEN
-                      vcmax(i,j)=vcmax(i,j)*79.1/vcmax_opt(j)
-                      vjmax(i,j)=vjmax(i,j)*152.4/vjmax_opt(j)
+                      IF (four2 == .TRUE.) THEN
+                         vcmax(i,j)=vcmax(i,j)*vcmax_src/vcmax_opt(j)
+                      ENDIF
+                      IF (four3 == .TRUE.) THEN
+                         vjmax(i,j)=vjmax(i,j)*vjmax_src/vjmax_opt(j)
+                      ENDIF
                    ENDIF

                    ! Final age-related decline in photosynthesis efficiency
Index: src_stomate/stomate_constants.f90
===================================================================
--- src_stomate/stomate_constants.f90    (revision 561)
+++ src_stomate/stomate_constants.f90    (working copy)
@@ -188,6 +188,9 @@
     REAL(r_std), DIMENSION(nvm)                  :: lowgpp_time
     REAL(r_std), DIMENSION(nvm)                  :: leaffall
     REAL(r_std), DIMENSION(nvm)                  :: leafagecrit
+    REAL(r_std), DIMENSION(nvm)                  :: rootagecrit
     REAL(r_std)                       :: tau_hum_month
     REAL(r_std)                       :: tau_hum_week
     REAL(r_std)                       :: tau_t2m_month
@@ -329,6 +332,10 @@
   TYPE(pheno_type),SAVE :: pheno_crit
```

```
 ! time constant for leaf age discretisation (d)
   REAL(r_std), SAVE, DIMENSION(nvm)                     :: leaf_timecst
!TDG
+! time constant for root age discretisation (d)
+  REAL(r_std), SAVE, DIMENSION(nvm)                     :: root_timecst
   ! maximum LAI, PFT-specific
   REAL(r_std), SAVE, DIMENSION (nvm)                    :: lai_max

@@ -401,7 +408,10 @@

   ! critical leaf age, tabulated (d)
   REAL(r_std), SAVE, DIMENSION(nvm)                     :: leafagecrit_tab
-
+  ! critical root age, tabulated (d)
+  REAL(r_std), SAVE, DIMENSION(nvm)                     :: rootagecrit_tab
   ! which phenology model is used? (tabulated)
   CHARACTER(len=6), SAVE, DIMENSION(nvm)                :: pheno_model_tab
 ! List of avaible phenology models :
@@ -612,8 +622,16 @@
 !!$ &                 .12,     .12,      .0,      .0,      .0,      .0 /)
 ! maximum LAI, PFT-specific
   lai_max(2:nvm) =   &
- & (/     7.,     7.,      5.,      5.,      5.,     4.5, &
+ & (/     7.,     7.,      1.9,      5.,      2.5,     4.5, &
  &        4.5,    3.0,     2.5,     2.5,     5.,     5.  /)
+!!$ & (/     7.,     7.,      5.,      5.,      5.,     4.5, &
+!!$ &        4.5,    3.0,     2.5,     2.5,     5.,     5.  /)
+IF (four1 == .TRUE.) THEN
+ CALL getin_p ('LAI_MAX_SRC',lai_max(6))
+ENDIF
+
 ! residence time (y) of trees
   residence_time(2:nvm) =    &
  & (/    30.0,    30.0,    40.0,    40.0,    40.0,    80.0, &
@@ -694,8 +712,10 @@
 ! Maximum rate of carboxylation
 !Shilong
   vcmax_opt(2:nvm) =     &
- & (/      65.,     65.,     35.,     45.,     55.,     35.,      &
+ & (/      65.,     65.,     85.,     45.,     55.,     35.,      &
  &         45.,     35.,     70.,      70.,   70.,    70.      /)
+! & (/      65.,     65.,     35.,     45.,     55.,     35.,      &
+! &         45.,     35.,     70.,     70.,    70.,    70.      /)
   CALL getin_p("vcmax_opt", vcmax_opt)
 ! 1.9.3
 !!$  vcmax_opt(2:nvm) =     &
@@ -714,8 +734,10 @@
 !-
 ! Maximum rate of RUbp regeneration
   vjmax_opt(2:nvm) =     &
- & (/       130.,    130.,     70.,     80.,    110.,     70.,       &
+ & (/       130.,    130.,    160.,      80.,    110.,     70.,       &
  &           90.,     70.,    160.,    160.,    200.,    200.      /)
+! & (/       130.,    130.,     70.,     80.,    110.,     70.,       &
+! &           90.,     70.,    160.,    160.,    200.,    200.      /)
 !-
   !DATA vjmax_opt_tab          /      0.,    130.,    130.,     75.,     90.,    120.,     75.,   &
   !                                  100.,     80.,    200.,    200.,    200.,    200. /
@@ -736,6 +758,13 @@
 ! & (/     730.,    180.,    910.,    730.,    180.,    910.,             &
 ! &        180.,    180.,    120.,    120.,     70.,     70.            /)
 !-
+! critical root age, tabulated (d)
+  rootagecrit_tab(2:nvm) =     &
+ & (/       730.,    180.,    910.,    730.,    180.,    910.,             &
+ &          180.,    180.,    120.,    120.,    90.,    90.            /)
+  CALL getin_p("rootagecrit_tab_6", rootagecrit_tab(6))
   ! which phenology model is used? (tabulated)
   pheno_model_tab(1:nvm) =   &
  & (/  'none  ',   'none  ',   'moi   ',   'none  ',   'none  ',    &
Index: src_stomate/stomate_forestry.f90
===================================================================
--- src_stomate/stomate_forestry.f90     (revision 561)
+++ src_stomate/stomate_forestry.f90     (working copy)
@@ -89,27 +93,54 @@
        Nmax=ss_selfth_curve*Nmax/10.**scale_dens

   END FUNCTION Nmax

   ! volume (m3/ha) = f(biomass (gC/m2))
-  REAL(r_std) FUNCTION vol_bm(biomass,pft)
+  REAL(r_std) FUNCTION vol_bm(biomass,pft,fm)
     REAL(r_std) :: biomass !(gC/m2)
-    INTEGER(i_std) :: pft
-    vol_bm = biomass*10**4/(ss_pipe_density*pipe_density(pft))
+    REAL(r_std) :: pipdens !(gC/m3)
+    INTEGER(i_std) :: pft, fm
+
+    IF ((fm == 4) .AND. (two1 == .TRUE.)) THEN
+      pipdens = 1.25e5 !5e5 !1.25e5
+    ELSE
+      pipdens = pipe_density(pft)
+    END IF
+
+    vol_bm = biomass*10**4/(ss_pipe_density*pipdens)
+
+    !IF (fm == 4) vol_bm = 1.5158*10**-10 * biomass ** 1.1538
   END FUNCTION vol_bm

   ! biomass (gC/m2) = f(volume (m3/ha))
-  REAL(r_std) FUNCTION bm_vol(volume,pft)
-  REAL(r_std) :: volume !(m3/ha)
-  INTEGER(i_std) :: pft
-    bm_vol = volume*(ss_pipe_density*pipe_density(pft))/10**4
+  REAL(r_std) FUNCTION bm_vol(volume,pft,fm)
+  REAL(r_std) :: volume !(m3/ha)
+  REAL(r_std) :: pipdens !(gC/m3)
+  INTEGER(i_std) :: pft, fm
+
+    IF ((fm == 4) .AND. (two1 == .TRUE.)) THEN
+      pipdens = 1.25e5 !5e5 !1.25e5
+    ELSE
+      pipdens = pipe_density(pft)
+    END IF
+
+    bm_vol = volume*(ss_pipe_density*pipdens)/10**4
+
+    !IF (fm == 4) bm_vol = 266928247 * volume ** 0.8539
   END FUNCTION bm_vol

-  ! total volume = f(circumference of each individual tree) in m3/ha
+  ! total volume = f(circumference of each individual tree) in m3/ha = f(m)
   ! based on volume table principle applied to IFN : V=7.4*circ**(1.
   !/3)*ba R2=0.91
-  REAL(r_std) FUNCTION vol_circ(circ,ntrees,pft)
-    INTEGER(i_std), INTENT(in) :: ntrees, pft
+  REAL(r_std) FUNCTION vol_circ(circ,ntrees,pft,fm)
+    INTEGER(i_std), INTENT(in) :: ntrees, pft, fm
     REAL(r_std), DIMENSION(ntrees), INTENT(in) :: circ
+    REAL(r_std) :: pipdens !(gC/m3)
     REAL(r_std)            :: kg_to_gC=0.5*1000.,m_to_cm=100.,a,b
     ! METHOD 1 : Zianis 2004 -> B=a*D**b with b=2.3679 and a=0.1165 B
     ! in kg and D in cm
@@ -127,17 +158,35 @@
 !!$    b=2.377
```

```
     a=7.03*b**(-4.76)
+
+    IF ((fm == 4) .AND. (two1 == .TRUE.)) THEN
+       a = 0.033
+       b = 2.609
+    END IF
+
+    IF ((fm == 4) .AND. (two1 == .TRUE.)) THEN
+      pipdens = 1.25e5 !5e5 !1.25e5
+    ELSE
+      pipdens = pipe_density(pft)
+    END IF
+
     vol_circ = sum(kg_to_gC*a/(ss_pipe_density*pipe_density(pft)) &
          &*(m_to_cm*1/ss_circ_bm*circ/pi)**b)

+!!    IF (fm == 4) vol_circ = 0.01850 * (sum(circ)/max(1,size(circ))) ** 0.3698
+!!    IF (fm == 4) vol_circ = sum(1.3742 * (circ) ** 2.6791)
+
     ! METHOD 2 : volume table
 !!$    vol_circ = sum(tarif*circ**(1./3)*circ**2/(4*pi*rwood_ratio))
   END FUNCTION vol_circ

-  ! circumference = f(volume of tree) in m
-  REAL(r_std) FUNCTION circ_vol(volume,pft)
+  ! circumference = f(volume of tree) in m = f(m3)
+  REAL(r_std) FUNCTION circ_vol(volume,pft,fm)
     REAL(r_std) :: volume
-    INTEGER(i_std) :: pft
+    REAL(r_std) :: pipdens !(gC/m3)
+    INTEGER(i_std) :: pft, fm
     REAL(r_std)            :: kg_to_gC=0.5*1000.,m_to_cm=100.,a,b
     SELECT CASE (pft)

@@ -151,20 +200,38 @@

        ENDSELECT
 !!$    b=2.377
+
     a=7.03*b**(-4.76)
+
+    IF ((fm == 4) .AND. (two1 == .TRUE.)) THEN
+       a = 0.032
+       b = 2.636
+    END IF
+
+    IF ((fm == 4) .AND. (two1 == .TRUE.)) THEN
+      pipdens = 1.25e5 !5e5 !1.25e5
+    ELSE
+      pipdens = pipe_density(pft)
+    END IF
+
     ! METHOD 1 : Zianis 2004 -> B=a*D**b with b=2.3679 and a=0.1165
     circ_vol = ss_circ_bm*pi/m_to_cm*(volume*ss_pipe_density &
          &*pipe_density(pft)/(kg_to_gC*a))**(1./b)
+
+    !!IF (fm == 4) circ_vol = 0.8666 * volume ** 0.3698
+
     ! METHOD 2 : volume table
 !!$    circ_vol = (4*pi*rwood_ratio*volume/tarif)**(3./7)
   END FUNCTION circ_vol

   ! height = f(circumference) in m
-  REAL(r_std) FUNCTION height_circ(circ,ba,dens,pft)
+  REAL(r_std) FUNCTION height_circ(circ,ba,dens,pft,fm)
     REAL(r_std) :: circ,ba
     ! Allometry parameters
     REAL(r_std),DIMENSION(nvm) :: a,c,d,p,q
-    INTEGER(i_std) :: pft,dens
+    INTEGER(i_std) :: pft,dens,fm

     ! Simple allometry: Chapman-Richards equation (Pretzsch et al
     ! 2002)
@@ -186,11 +253,17 @@
          &*circ*100/(2*pi)))**p(pft)
     height_circ = ss_height_circ*height_circ
 !!$    height_circ = pipe_tune2*(circ/pi)**pipe_tune3
+
+
+    IF ((fm == 4) .AND. (two1 == .TRUE.)) height_circ = 17.2684 * circ ** 0.6791
+
   END FUNCTION height_circ

   ! deltavol_ as a function of gamma
-  REAL(r_std) FUNCTION deltavol_(gamma,sigma,circ0,pft)
-    INTEGER(i_std) :: pft
+  REAL(r_std) FUNCTION deltavol_(gamma,sigma,circ0,pft,fm)
+    INTEGER(i_std) :: pft, fm
     REAL(r_std) ::gamma,sigma,sigma_cm
     REAL(r_std) :: delta_ba ! delta_ba =increase in basal area
     ! according to the model of Deleuze 2003
@@ -206,16 +279,18 @@
     !!**pipe_tune3*((circ**2/(4*pi)+delta_ba)**(1.+pipe_tune3/2)
     !!-(circ**2/(4*pi))**(1.+pipe_tune3/2)),0.)
     deltavol_=max( &
-         vol_circ( (/ SQRT(ba_new*4*pi) /),1,pft) - &
-         vol_circ((/ circ0 /),1,pft),0.)
+         vol_circ( (/ SQRT(ba_new*4*pi) /),1,pft,fm) - &
+         vol_circ((/ circ0 /),1,pft,fm),0.)
 !!$    deltavol_= max(vol_circ(sqrt(ba_new(1)*4*pi),1,pft)-vol_circ(circ0(1) &
 !!$         &,1,pft),0.)
      END FUNCTION deltavol_

   ! deltavol_tot as a function of gamma
-  REAL(r_std) FUNCTION deltavol_tot(gamma,sigma,circ0,ntrees,pft)
+  REAL(r_std) FUNCTION deltavol_tot(gamma,sigma,circ0,ntrees,pft,fm)
     REAL(r_std) ::gamma,sigma,temp
-    INTEGER(i_std) ::ntrees,i,pft
+    INTEGER(i_std) ::ntrees,i,pft,fm
     REAL(r_std), DIMENSION(ntrees) :: circ0 ! delta_ba =increase in
     ! basal area according to the model of Deleuze 2003
     DOUBLE PRECISION, DIMENSION(ntrees) :: circ
@@ -224,11 +299,13 @@
     DO i=1,ntrees
 !!$       If (circ0(i)>=sigma) temp = temp + deltavol_(gamma,sigma
        !!,circ0(i))
-    temp= temp+deltavol_(gamma,sigma,circ0(i),pft)
+    temp= temp+deltavol_(gamma,sigma,circ0(i),pft,fm)
     ENDDO
     deltavol_tot = temp
  END FUNCTION deltavol_tot

  ! Sum crowns
  REAL(r_std) FUNCTION sum_crowns(circ0,height0,ntrees,j)
    INTEGER(i_std) ::ntrees,j
@@ -271,7 +352,7 @@
                            when_growthinit, leaf_frac, &
 !!$, gpp_lastyear, gpp_sofar &
                            veget_max, &
-                           sigma, gamma, age_stand, rotation_n, last_cut, av_circ, ba, dom_height, av_height, &
+                           sigma, gamma, age_stand, rotation_n, last_cut, av_circ, ba, dom_height, av_height,av_height_bt, &
                            med_height,q1_height,q3_height, &
                            standing_ab_bm, deltavol, standing_ab_vol, harvestable_vol, harvestable_vol_inc, &
                            exported_bm, exported_vol, exported_vol_th, circ_lim, circ_min, circ_max, &
```

```
@@ -373,7 +454,7 @@
     ! Dominant height in the stand (m)
     REAL(r_std),DIMENSION(npts,nvm),INTENT(out)           :: dom_height
     ! Average height in the stand (m)
-    REAL(r_std),DIMENSION(npts,nvm),INTENT(out)           :: av_height
+    REAL(r_std),DIMENSION(npts,nvm),INTENT(out)           :: av_height,av_height_bt
     ! First quartile, median and 3rd quartile of height (m)
     REAL(r_std),DIMENSION(npts,nvm),INTENT(out)           :: med_height,q1_height,q3_height
     ! Standing aboveground biomass (gC/m2)
@@ -562,8 +643,8 @@
     LOGICAL, SAVE, DIMENSION(nvm) :: plantation
     !  Record number of trees or tree volume in circumference categories
     LOGICAL :: density_rec = .TRUE.
-    ! Itinerary type (orsay = 2 or popface = 1)
-    INTEGER(i_std), SAVE                     :: itinerary
+    ! Itinerary type (orsay = 2 or popface = 1)   (array of 1 and 0; every number is a year; 1 means coppice)
+    INTEGER(i_std), SAVE, DIMENSION(20)     :: itinerary
     ! Age at which clearcut is mandatory
     INTEGER(i_std), SAVE                     :: age_target_def
 !!$    ! Is the initial circumference distribution forced ?
@@ -684,16 +765,20 @@
           DO i=1,npts

              ! Minimum lai_max_calc set to 2 otherwise no enough reserves for forest unable to grow back
            IF ( lai_max_calc(i,j) .LT. 2.) THEN
               IF (j .NE. 9) THEN
                  lai_max_calc(i,j)=2.
               ELSE
                  lai_max_calc(i,j)=lai_max(j)
               ENDIF
+           ENDIF
               ! In SRC, LAI is limited only the first 2 years (cf. Pontailler 1999, Liberloo 2006)
               !  probably due to good conditions and higher stem density after first coppicing
+               !IF (forest_managed(i,j) == 4) lai_max_calc(i,j)=min(real(0+1)**2/4.,1.)*lai_max(j)
+            IF (clearfirst .AND. forest_managed(i,j) == 4 .AND. three3 == .TRUE.)
  lai_max_calc(i,j)=min(real(0+1)**2/4.,1.)*lai_max(j)
+           !ENDIF

 !!$             IF (forest_managed(i,j)>0 .AND. clearfirst .AND. (veget_max(i,j)>0)) THEN
@@ -727,8 +814,8 @@
 !!$                     biomass(i,j,:)=adapt_ratio*biomass(i,j,:)
                    biomass(i,j,:)=0.
                    biomass(i,j,icarbres)=50.
-                   biomass(i,j,isapabove)=min(200.,bm_vol(vol_circ(circ_ij0,ntrees0,j)/pipe_density(j),j))
-                   biomass(i,j,iheartabove)=max(0.,bm_vol(vol_circ(circ_ij0,ntrees0,j)/pipe_density(j),j) - &
+                   biomass(i,j,isapabove)=min(200.,bm_vol(vol_circ(circ_ij0,ntrees0,j,
   forest_managed(i,j))/pipe_density(j),j,forest_managed(i,j)))
+
   biomass(i,j,iheartabove)=max(0.,bm_vol(vol_circ(circ_ij0,ntrees0,j,forest_managed(i,j))/pipe_density(j),j,forest_managed(i
   ,j)) - &
                                                    biomass(i,j,icarbres)-biomass(i,j,isapabove))
                    biomass(i,j,isapbelow)=0.2*biomass(i,j,isapabove)
                    biomass(i,j,iheartbelow)=0.2*biomass(i,j,iheartabove)
@@ -794,16 +883,16 @@
 !!$                         circ_class_n(i,j,k)=count((circ_ij0 >=circ_class(i,j,k)) .AND. (circ_ij0 <=
   circ_class(i,j,k+1)))
                       deb=int(real(k-1)*ntrees0/real(ncircclass-1))
                       fin=int(real(k)*ntrees0/real(ncircclass-1))
-                      circ(i,j,k)=vol_circ(circ_ij0(deb:fin),fin-deb+1,j)
+                      circ(i,j,k)=vol_circ(circ_ij0(deb:fin),fin-deb+1,j,forest_managed(i,j))
                    ELSE IF (k==1) THEN
 !!$                         circ_class_n(i,j,k)=count((circ_ij0 >=circ_class(i,j,k)) .AND. (circ_ij0 < circ_class(i,j,k+1)))
                       fin=int(real(k)*ntrees0/real(ncircclass-1)-1)
-                      circ(i,j,k)=vol_circ(circ_ij0(1:fin),fin,j)
+                      circ(i,j,k)=vol_circ(circ_ij0(1:fin),fin,j,forest_managed(i,j))
                    ELSE
 !!$                         circ_class_n(i,j,k)=count((circ_ij0 >=circ_class(i,j,k)) .AND. (circ_ij0 < circ_class(i,j,k+1)))
                       deb=int(real(k-1)*ntrees0/real(ncircclass-1))
                       fin=int(real(k)*ntrees0/real(ncircclass-1)-1)
-                      circ(i,j,k)=vol_circ(circ_ij0(deb:fin),fin-deb+1,j)
+                      circ(i,j,k)=vol_circ(circ_ij0(deb:fin),fin-deb+1,j,forest_managed(i,j))
                    ENDIF
                 ENDDO ! k loop

@@ -899,7 +988,7 @@
                       ! Equation d'orchidee
                       deltavol(i,j)=vol_bm(biomass(i,j,isapabove)+biomass(i,j,iheartabove)&
                             &+biomass(i,j,icarbres)*frac_shoot-biomass_lastyear(i,j,isapabove)-biomass_lastyear&
-                            &(i,j,iheartabove)-biomass_lastyear(i,j,icarbres)*frac_shoot,j) !m3/ha/an
+                            &(i,j,iheartabove)-biomass_lastyear(i,j,icarbres)*frac_shoot,j,forest_managed(i,j)) !m3/ha/an

                    ENDIF
                    IF (forest_managed(i,j)>0 .AND. veget_max(i,j)>0 .AND. age_stand(i,j) <= old_growth_age) THEN
@@ -919,11 +1008,14 @@
                             age_target = 200
                          ENDIF
                       ELSE
-                         IF (itinerary > 1) THEN
-                            age_target=11
-                         ELSE
-                            age_target = 18
-                         ENDIF
+! changed way to determine age_target
+                          age_target=SIZE(itinerary)
+!                          age_target=20
+!                         IF (itinerary > 1) THEN
+!                            age_target=11
+!                         ELSE
+!                            age_target = 18
+!                         ENDIF
                       ENDIF


@@ -938,7 +1030,7 @@
                       ! Equation d'orchidee
                       deltavol(i,j)=vol_bm(biomass(i,j,isapabove)+biomass(i,j,iheartabove)+biomass(i,j,icarbres)&
                          &*frac_shoot-biomass_lastyear(i,j,isapabove)-biomass_lastyear(i,j,iheartabove)-&
-                         &biomass_lastyear(i,j,icarbres)*frac_shoot,j)/10.**scale_dens !m3/(ha/10.**scale_dens)/yr
+                         &biomass_lastyear(i,j,icarbres)*frac_shoot,j,forest_managed(i,j))/10.**scale_dens
   !m3/(ha/10.**scale_dens)/yr

                       ! numproc est le numéro du processeur - 1
                       numproc=0
@@ -960,7 +1052,7 @@
                       IF (bavard_f >=1) THEN
                          write(*,*) 'tata',biomass(i,j,isapabove),biomass(i,j,iheartabove),&
                            &biomass_lastyear(i,j,isapabove),biomass_lastyear(i,j,iheartabove)
-                         write(*,*) 'tata2',vol_bm(biomass(i,j,isapabove)+biomass(i,j,iheartabove),j)
+                         write(*,*) 'tata2',vol_bm(biomass(i,j,isapabove)+biomass(i,j,iheartabove),j,forest_managed(i,j))
                       ENDIF


@@ -1016,14 +1110,14 @@

                       ! The last tree gets its value
                       circ_ij0(ntrees0)=circ_class(i,j,ncircclass)
-                      vol_ij(ntrees0)=vol_circ((/ circ_ij0(ntrees0) /),1,j)
+                      vol_ij(ntrees0)=vol_circ((/ circ_ij0(ntrees0) /),1,j,forest_managed(i,j))

                    DO k=1, ncircclass-1
                          fin=int(real(k*ntrees0)/real(ncircclass-1)-1)

                       IF (circ_class(i,j,k) == circ_class(i,j,k+1)) THEN
                            circ_ij0(deb:fin)=circ_class(i,j,k)
-                           vol_ij(deb:fin)=vol_circ(circ_ij0(deb:fin),fin-deb+1,j)
```

```
+                            vol_ij(deb:fin)=vol_circ(circ_ij0(deb:fin),fin-deb+1,j,forest_managed(i,j))

                             deb=fin+1

@@ -1045,7 +1139,7 @@

                          DO m= deb, fin, 1
                             circ_ij0(m)=circ_class(i,j,k)+(circ_class(i,j,k+1)-circ_class(i,j,k))*real(m-deb)/delta
-                            vol_ij(m)=vol_circ((/ circ_ij0(m) /),1,j)
+                            vol_ij(m)=vol_circ((/ circ_ij0(m) /),1,j,forest_managed(i,j))
                             sum_circ=sum_circ+vol_ij(m)
                          ENDDO
                          sum_circ=sum(vol_ij(deb:fin))
@@ -1062,7 +1156,7 @@
                                  sum_circ0=sum_circ
                                  pos=maxloc(circ_ij0,MASK=circ_ij0<circ_class(i,j,k+1))
                                  delta=size(circ_ij0(deb:pos(1)))
-                                 vcirc_class = vol_circ((/ circ_class(i,j,k+1) /),1,j)
+                                 vcirc_class = vol_circ((/ circ_class(i,j,k+1) /),1,j,forest_managed(i,j))
                                  vol_ij(deb:pos(1))=MIN(vcirc_class,vol_ij(deb:pos(1))+(circ(i,j,k)-sum_circ)/delta)
                                  sum_circ=sum(vol_ij(deb:fin))
                                  no_infinite=no_infinite+1
@@ -1103,7 +1197,7 @@
                                pos=minloc(circ_ij0,MASK=circ_ij0>circ_class(i,j,k))
                                delta=size(circ_ij0(pos(1):fin))

-                               vcirc_class=vol_circ((/ circ_class(i,j,k) /),1,j)
+                               vcirc_class=vol_circ((/ circ_class(i,j,k) /),1,j,forest_managed(i,j))
                                vol_ij(pos(1):fin)=max(vcirc_class,vol_ij(pos(1):fin)&
                                   &                   -(sum_circ-circ(i,j,k))/delta)
                                sum_circ=sum(vol_ij(deb:fin))
@@ -1132,7 +1226,7 @@
                          ENDIF

                          DO m=deb,fin
-                            circ_ij0(m) = circ_vol(vol_ij(m),j)
+                            circ_ij0(m) = circ_vol(vol_ij(m),j,forest_managed(i,j))
                          ENDDO

                          deb=fin+1
@@ -1153,8 +1247,13 @@
                    ENDDO ! k loop

                    ba(i,j)=sum(circ_ij0**2/(4*pi))*10.**scale_dens
+!                   IF (forest_managed(i,j)==4 .AND. j==6) THEN
+!                      lai_max(j)=ba(i,j)*0.5
+!                   ENDIF

-                   vol0=bm_vol(real(vol_circ(circ_ij0,ntrees0,j)),j)
+                   vol0=bm_vol(real(vol_circ(circ_ij0,ntrees0,j,forest_managed(i,j))),j,forest_managed(i,j))

                    IF (bavard_f >=1) THEN

@@ -1166,7 +1265,7 @@
                       !volume of individuals after self-thinnings

                print *,"sum(vol_ij)",sum(vol_ij),"sum(circ(i,j,:))",sum(circ(i,j,:))&
-                   &,"vol_circ(circ_ij0)",vol_circ(circ_ij0,ntrees0,j)
+                   &,"vol_circ(circ_ij0)",vol_circ(circ_ij0,ntrees0,j,forest_managed(i,j))
                    ! Check if biomass and volume match
 !!$                           IF (vol_bm(biomass(i,j,isapabove)+biomass(i,j,iheartabove)+biomass(i,j,icarbres)*frac_shoot) /= &
 !!$                                 vol_circ(circ_ij0,ntrees0)) THEN
@@ -1261,7 +1360,7 @@
                       inf=1.

                       no_infinite=0.
-                      DO WHILE (deltavol_tot(inf,sigma(i,j),circ_ij0,ntrees0,j)>= vol_to_dist)
+                      DO WHILE (deltavol_tot(inf,sigma(i,j),circ_ij0,ntrees0,j,forest_managed(i,j))>= vol_to_dist)
                          inf=inf/10.
                          no_infinite=no_infinite+1
                          IF (no_infinite > 10) THEN
@@ -1271,7 +1370,7 @@
                       ENDDO

                       no_infinite=0.
-                      DO WHILE (deltavol_tot(sup,sigma(i,j),circ_ij0,ntrees0,j)<= vol_to_dist)
+                      DO WHILE (deltavol_tot(sup,sigma(i,j),circ_ij0,ntrees0,j,forest_managed(i,j))<= vol_to_dist)
                          sup=sup*10.
                          no_infinite=no_infinite+1
                          IF (no_infinite > 10) THEN
@@ -1282,8 +1381,8 @@
                       no_infinite=0.
                       DO WHILE (sup-inf > 10.**(-5))
                          z=inf+(sup-inf)/2.
-                         IF (deltavol_tot(z,sigma(i,j),circ_ij0,ntrees0,j)==vol_to_dist) exit
-                         IF (deltavol_tot(z,sigma(i,j),circ_ij0,ntrees0,j)<vol_to_dist) THEN
+                         IF (deltavol_tot(z,sigma(i,j),circ_ij0,ntrees0,j,forest_managed(i,j))==vol_to_dist) exit
+                         IF (deltavol_tot(z,sigma(i,j),circ_ij0,ntrees0,j,forest_managed(i,j))<vol_to_dist) THEN
                             inf=z
                          ELSE
```

```
                             sup=z
@@ -1301,7 +1400,7 @@

                          print *,'sigma(i,j)',sigma(i,j),'gamma(i,j)',gamma(i,j),'deltavol_tot(gamma(i,j),&
                            &sigma(i,j),circ_ij0,ntrees0)',deltavol_tot(gamma(i,j),sigma(i,j),&
-                           &circ_ij0,ntrees0,j),'deltavol(i,j)',deltavol(i,j)
+                           &circ_ij0,ntrees0,j,forest_managed(i,j)),'deltavol(i,j)',deltavol(i,j)
                          write(*,*) count(circ_ij0>=sigma(i,j)),real(count(circ_ij0>=sigma(i,j)))/count(circ_ij0>0.)

                       ENDIF
@@ -1331,7 +1430,7 @@
 !!$                         vol_inc(:)=gamma(i,j)*(1-sigma(i,j)/circ_ij0(:)) ! vol_inc in m3/year/tree
 !!$             vol_inc(:)=form*h_100*2*sqrt(pi)*((circ0(:)**2/(4*pi)+gamma(i,j)*(circ0(:)-sigma(i,j)))&
 !!$                *sqrt((circ0(:)**2/(4*pi)+gamma(i,j)*(circ0(:)-sigma(i,j))))-circ0(:)**3/(8*pi**1.5))
-                        vol_inc(m)=deltavol_(gamma(i,j),sigma(i,j),circ_ij0(m),j)
+                        vol_inc(m)=deltavol_(gamma(i,j),sigma(i,j),circ_ij0(m),j,forest_managed(i,j))
 !!$             vol_inc(:) = form*h_100*(2*sqrt(pi))**pipe_tune3*((circ_ij0(:)**2/(4*pi)+gamma(i,j)*&
 !!$              (circ_ij0(:)-sigma(i,j)))**(1.+pipe_tune3/2)-(circ_ij0(:)**2/(4*pi))**(1.+pipe_tune3/2))

@@ -1341,7 +1440,7 @@
                       ENDDO
 !!$                         vol_inc=vol_inc+(deltavol(i,j)-vol_to_dist)/ntrees0
                       harvestable_vol_inc(i,j)=sum(vol_inc, vol_inc > deltavol_(gamma(i,j),&
-                               &sigma(i,j),harv_circ,j))*(1-branch_ratio(j))
+                               &sigma(i,j),harv_circ,j,forest_managed(i,j)))*(1-branch_ratio(j))
                       IF (bavard_f >=1) write(*,*) 'crois vol',sum(vol_inc), deltavol(i,j), vol_inc(ntrees0)

                       !---------------
@@ -1355,23 +1454,25 @@
 !!$                         circ_inc(:)=(vol_inc(:)*4*pi/(h_100*form)+&
 !!^                                     circ_ij0(:)**(2.+pipe_tune3))**(real(1)/(2.+pipe_tune3))-circ_ij0(:)
                       DO k=1,ntrees0
-                         circ_inc(k)=circ_vol(vol_inc(k)+vol_circ((/ circ_ij0(k) /),1,j),j)-circ_ij0(k)
+                         circ_inc(k)=circ_vol(vol_inc(k)+vol_circ((/ circ_ij0(k)
  /),1,j,forest_managed(i,j)),j,forest_managed(i,j))-circ_ij0(k)
                          circ_ij0(k)=circ_ij0(k)+circ_inc(k)

                          ! Allometric attribution of height
-                         height_ij(k) = height_circ(circ_ij0(k),ba(i,j),nint(ntrees0*10.**scale_dens),j)
+                         height_ij(k) = height_circ(circ_ij0(k),ba(i,j),nint(ntrees0*10.**scale_dens),j,forest_managed(i,j))
+print *,"Individual D and H inc: height_ij(k)",height_ij(k)
+
                       ENDDO

-                      vol1=bm_vol(real(vol_circ(circ_ij0,ntrees0,j)),j)
```

```
+                      vol1=bm_vol(real(vol_circ(circ_ij0,ntrees0,j,forest_managed(i,j))),j,forest_managed(i,j))
                       Dg=(sum((100*circ_ij0/pi)**2.)/(count(circ_ij0 .GT. 0.0)))**0.5 !cm

                       DO k=1,ntrees0
-                         vol_ij(k)=vol_circ((/ circ_ij0(k) /),1,j)
+                         vol_ij(k)=vol_circ((/ circ_ij0(k) /),1,j,forest_managed(i,j))
                       ENDDO

                       IF (bavard_f >=1) THEN

-                         print*,'deltabm',bm_vol(deltavol(i,j),j),'vol0',vol0,'vol1',vol1,vol1-vol0
+                         print*,'deltabm',bm_vol(deltavol(i,j),j,forest_managed(i,j)),'vol0',vol0,'vol1',vol1,vol1-vol0
                          write(*,*) 'test circ',height_ij(ntrees0),height_inc(ntrees0),&
                                                 circ_ij0(ntrees0),circ_inc(ntrees0)
                          write(*,*) 'IND**********',ind(i,j), Dg,sum(circ_ij0),count(circ_ij0 .GT. 0.0)
@@ -1439,41 +1540,76 @@
                             CALL thinning(.TRUE.,circ_ij0,ntrees0,psi_rdi,rdi_inf,rdi(i,j),circ_lim(i,j),&
                               &npts,i,j,ind,vol_thinned,height_ij,vol_ij,biomass,frac_shoot,bm_to_litter &
 !!$                                   ,exported_bm_lastyear &
-                              &,exported_bm,ba,av_height,dia,bavard_f,density_rec,circ_class_n_bt,&
+                              &,exported_bm,ba,av_height,av_height_bt,dia,bavard_f,density_rec,circ_class_n_bt,&
                               &forest_managed,scale_dens,class_width,class_max)

-                         IF (exported_bm(i,j) > 0) exported_vol_th(i,j)=vol_bm(exported_bm(i,j),j)
+                         IF (exported_bm(i,j) > 0) exported_vol_th(i,j)=vol_bm(exported_bm(i,j),j,forest_managed(i,j))

                          ENDIF
                       ENDIF ! END of THINNING OF a high stand

                       ! THINNING OF A COPPICE
+
+! coppice itinerary is now read from run.def, instead of loading a fixed scheme
                       ! Coppicing itinerary
-                      IF (itinerary > 1) THEN
-                         ! Thin after first year and then every two years
-                         coppice_th=(mod(age_stand(i,j)+1,2)==0)
-                      ELSE
-                         ! Thin every three years
-                         coppice_th=(mod(age_stand(i,j),3)==0)
-                      ENDIF
+!                      IF (itinerary > 1) THEN
+!                         ! Thin after first year and then every two years
+!                         coppice_th=(mod(age_stand(i,j)+1,2)==0)
+!                      ELSE
+!                         ! Thin every three years
+!                         coppice_th=(mod(age_stand(i,j),3)==0)
+!                      ENDIF

+                       ! New method of entering coppicing itinerary
+write(*,*) "one1",one1
+                       IF (one1 == .TRUE.) THEN
+                          coppice_th=itinerary(age_stand(i,j))
+                       ELSE
+                          coppice_th=0
+                       ENDIF
+          print *,'coppice_th',coppice_th

+! change fm==1 to 4 to test original model with coppice against coppice model
                      IF ((age_stand(i,j).LT.age_target).AND.(early_cut<2.OR.((ntrees0*10.**scale_dens&
              &.GT.dens_target(j).AND.early_cut==2))).AND.forest_managed(i,j)==4.AND.coppice_th) THEN

                         last_cut(i,j) = 0

-                        IF (age_stand(i,j)<=3) nb_stems=2.
-                        IF (age_stand(i,j)>3) nb_stems=4.
+                        rotation_n(i,j) = rotation_n(i,j)+1

+!                        IF (age_stand(i,j)<=3) nb_stems=2.
+!                        IF (age_stand(i,j)>3) nb_stems=4.
+
+! add new way to determine number of shoots per stool/coppice cycle
+                        !Determine number of shoots per stool/coppice cycle
+                        IF (two2 == .TRUE.) THEN
+                           IF (rotation_n(i,j)==2) nb_stems=2.
+                           IF (rotation_n(i,j)>=3) nb_stems=4.
+                        ELSE
+                           nb_stems=1.
+                        ENDIF
+
                         ntrees0=nint(nb_stems*nmaxtrees(j))
+!NT0
+print *,'ntrees0=nint(nb_stems*nmaxtrees(j))',ntrees0
                         IF (ntrees0>10000) THEN
                            ntrees0=nint(real(ntrees0)/10.)
                            scale_dens=1.
                         ENDIF
@@ -1491,7 +1627,7 @@
                         CALL thinning(.TRUE.,circ_ij0,ntrees0,psi_rdi,rdi_inf,rdi(i,j),circ_lim(i,j),&
                           &npts,i,j,ind,vol_thinned,height_ij,vol_ij,biomass,frac_shoot,bm_to_litter &
 !!$                                  ,exported_bm_lastyear &
```

```
-                           &,exported_bm,ba,av_height,dia,bavard_f,density_rec,circ_class_n_bt,&
+                           &,exported_bm,ba,av_height,av_height_bt,dia,bavard_f,density_rec,circ_class_n_bt,&
                            &forest_managed,scale_dens,class_width,class_max)

                       ENDIF   ! END OF THINNING OF A COPPICE
@@ -1529,8 +1667,9 @@
                             vol_ij(:)=0.
                             height_ij(:)=0.
                             DO k=1,ntrees0
-                               vol_ij(k)=vol_circ((/ circ_ij0(k) /),1,j)
-                               height_ij(k) = height_circ(circ_ij0(k),ba(i,j),nint(ntrees0*10.**scale_dens),j)
+                               vol_ij(k)=vol_circ((/ circ_ij0(k) /),1,j,forest_managed(i,j))
+                               height_ij(k) =
   height_circ(circ_ij0(k),ba(i,j),nint(ntrees0*10.**scale_dens),j,forest_managed(i,j))
+print *,"FINAL CUT - early cut: height_ij(k)",height_ij(k)
                             ENDDO

                             IF (bavard_f >=1) print *,'ntrees0futur',count(circ_ij0 .GT. 0.0)
@@ -1593,8 +1734,9 @@
                          vol_ij(:)=0.
                          height_ij(:)=0.
                          DO k=1,ntrees0
-                            vol_ij(k)=vol_circ((/ circ_ij0(k) /),1,j)
-                            height_ij(k) = height_circ(circ_ij0(k),ba(i,j),nint(ntrees0*10.**scale_dens),j)
+                            vol_ij(k)=vol_circ((/ circ_ij0(k) /),1,j,forest_managed(i,j))
+                            height_ij(k) =
   height_circ(circ_ij0(k),ba(i,j),nint(ntrees0*10.**scale_dens),j,forest_managed(i,j))
+print *,"FINAL CUT - not early cut: height_ij(k)",height_ij(k)
                          ENDDO

                          IF (bavard_f >=1) print *,'ntrees0futur',count(circ_ij0 .GT. 0.0)
@@ -1622,8 +1764,12 @@
                       ! SELF-THINNING !
                       !---------------!
                       ! trees actually dead with self-thinning if there was no human cut this year
-                      IF (vol_thinned .EQ. 0.) THEN
+                      !IF (vol_thinned .EQ. 0.) THEN                              ! self thinning enabled for SRC
+                      IF (vol_thinned .EQ. 0. .AND. forest_managed(i,j)<=3) THEN ! self thinning disabled for SRC

+!NT0 stop self-thinning
                          !self-thinning equation :
 !!$                            IF (age_stand(i,j) .GT. 0.) THEN
 !!$                               ind_killed(i,j)=MAX(0,ntrees0-int(Nmax(Dg,j,scale_dens)))
@@ -1639,7 +1785,7 @@
                             CALL thinning(.FALSE.,circ_ij0,ntrees0,psi_rdi,1.+delta_rdi,rdi(i,j),circ_lim(i,j),&
                                      &npts,i,j,ind,vol_thinned,height_ij,vol_ij,biomass,frac_shoot,bm_to_litter&
 !!$                                 ,exported_bm_lastyear &
-                                     &,exported_bm,ba,av_height,dia,bavard_f,density_rec,circ_class_n_bt,&
+                                     &,exported_bm,ba,av_height,av_height_bt,dia,bavard_f,density_rec,circ_class_n_bt,&
                                      &forest_managed,scale_dens,class_width,class_max)

                          ENDIF ! Ind_killed(i,j) > 0.
@@ -1673,7 +1819,8 @@

                    m=1
                    DO m=1,ntrees1
-                      height_ij1(m)=height_circ(circ_ij1(m),ba(i,j),nint(ntrees1*10.**scale_dens),j)
+                      height_ij1(m)=height_circ(circ_ij1(m),ba(i,j),nint(ntrees1*10.**scale_dens),j,forest_managed(i,j))
+print *,"SELF-THINNING: height_ij1(m)",height_ij1(m)
                    ENDDO

                    k=1
@@ -1703,15 +1850,15 @@
                       IF (k==(ncircclass-1)) THEN
                          deb=int(real((k-1)*ntrees1)/real(ncircclass-1))
                          fin=int(real(k*ntrees1)/real(ncircclass-1))
-                         circ(i,j,k)=vol_circ(circ_ij1(deb:fin), fin-deb+1,j)
+                         circ(i,j,k)=vol_circ(circ_ij1(deb:fin), fin-deb+1,j,forest_managed(i,j))
                       ELSE IF (k==1) THEN
                          deb=1
                          fin=int(real(k)*ntrees1/real(ncircclass-1)-1)
-                         circ(i,j,k)=vol_circ(circ_ij1(deb:fin), fin-deb+1,j)
+                         circ(i,j,k)=vol_circ(circ_ij1(deb:fin), fin-deb+1,j,forest_managed(i,j))
                       ELSE
                          deb=int(real(k-1)*ntrees1/real(ncircclass-1))
                          fin=int(real(k)*ntrees1/real(ncircclass-1)-1)
-                         circ(i,j,k)=vol_circ(circ_ij1(deb:fin), fin-deb+1,j)
+                         circ(i,j,k)=vol_circ(circ_ij1(deb:fin), fin-deb+1,j,forest_managed(i,j))
                       ENDIF

                    ENDDO
@@ -1759,7 +1906,7 @@

 ! In SRC, LAI is limited only the first 2 years (cf. Pontailler 1999, Liberloo 2006)
 ! probably due to good conditions and higher stem density after first coppicing
-                   IF (forest_managed(i,j) == 4) lai_max_calc(i,j)=min(real(age_stand(i,j)+1)**2/4.,1.)*lai_max(j)
+                   IF (forest_managed(i,j) == 4 .AND. three3 == .TRUE.)
   lai_max_calc(i,j)=min(real(age_stand(i,j)+1)**2/4.,1.)*lai_max(j)

                    ! If too few trees, lai_max_calc is limited by the total crown area
 !!$                   IF (ntrees1 .LE. 500) THEN
@@ -1783,7 +1930,7 @@
                    q3_height(i,j)=height_ij1(nint(real(3*ntrees1)/4))
                    standing_ab_bm(i,j)=biomass(i,j,isapabove)+biomass(i,j,iheartabove)+biomass(i,j,icarbres)*frac_shoot
                    standing_ab_vol(i,j)=sum(vol_ij)
-                   harvestable_vol(i,j)=sum(vol_ij, vol_ij > vol_circ(harv_circ,1,j))*(1-branch_ratio(j))
+                   harvestable_vol(i,j)=sum(vol_ij, vol_ij > vol_circ(harv_circ,1,j,forest_managed(i,j)))*(1-
   branch_ratio(j))
                    IF (last_cut(i,j)==0) exported_vol(i,j)=vol_thinned
                    circ_min(i,j)=minval(circ_ij1) !minval(height_ij1) !
                    circ_max(i,j)=maxval(circ_ij1) !maxval(height_ij1) !
@@ -1804,23 +1951,23 @@
                           & count((circ_ij1 >= 0.705 .AND. circ_ij1 < 1.175)),&
                           & count((circ_ij1 >= 1.175))/)
                    ELSE
-                      circ_class_n(i,j,:)= (/sum(vol_ij,(vol_ij > 0 .AND. vol_ij < vol_circ(class_width,1,j))),&
-                          & sum(vol_ij,(vol_ij > vol_circ(class_width,1,j) .AND. vol_ij < vol_circ(2*class_width,1,j))),&
-                          & sum(vol_ij,(vol_ij > vol_circ(2*class_width,1,j) .AND. vol_ij < vol_circ(3*class_width,1,j))),&
-                          & sum(vol_ij,(vol_ij > vol_circ(3*class_width,1,j) .AND. vol_ij < vol_circ(4*class_width,1,j))),&
-                          & sum(vol_ij,(vol_ij > vol_circ(4*class_width,1,j) .AND. vol_ij < vol_circ(5*class_width,1,j))),&
-                          & sum(vol_ij,(vol_ij > vol_circ(5*class_width,1,j) .AND. vol_ij < vol_circ(6*class_width,1,j))),&
-                          & sum(vol_ij,(vol_ij > vol_circ(6*class_width,1,j) .AND. vol_ij < vol_circ(7*class_width,1,j))),&
-                          & sum(vol_ij,(vol_ij > vol_circ(7*class_width,1,j) .AND. vol_ij < vol_circ(8*class_width,1,j))),&
-                          & sum(vol_ij,(vol_ij > vol_circ(8*class_width,1,j) .AND. vol_ij < vol_circ(9*class_width,1,j))),&
-                          & sum(vol_ij,(vol_ij > vol_circ(9*class_width,1,j) .AND. vol_ij < vol_circ(10*class_width,1,j))),&
-                          & sum(vol_ij,(vol_ij > vol_circ(10*class_width,1,j))),&
-                          & sum(vol_ij,(vol_ij > 0 .AND. vol_ij < vol_circ((/ class_max(1) /),1,j))),&
-                          & sum(vol_ij,(vol_ij > vol_circ((/ class_max(1) /),1,j) .AND. &
-                          &             vol_ij < vol_circ((/ class_max(2) /),1,j))),&
-                          & sum(vol_ij,(vol_ij > vol_circ((/ class_max(2) /),1,j) .AND. &
-                          &             vol_ij < vol_circ((/ class_max(3) /),1,j))),&
-                          & sum(vol_ij,(vol_ij > vol_circ((/ class_max(3) /),1,j)))/)
+                      circ_class_n(i,j,:)= (/sum(vol_ij,(vol_ij > 0 .AND. vol_ij <
   vol_circ(class_width,1,j,forest_managed(i,j)))),&
+                          & sum(vol_ij,(vol_ij > vol_circ(class_width,1,j,forest_managed(i,j)) .AND. vol_ij <
   vol_circ(2*class_width,1,j,forest_managed(i,j)))),&
+                          & sum(vol_ij,(vol_ij > vol_circ(2*class_width,1,j,forest_managed(i,j)) .AND. vol_ij <
   vol_circ(3*class_width,1,j,forest_managed(i,j)))),&
+                          & sum(vol_ij,(vol_ij > vol_circ(3*class_width,1,j,forest_managed(i,j)) .AND. vol_ij <
   vol_circ(4*class_width,1,j,forest_managed(i,j)))),&
+                          & sum(vol_ij,(vol_ij > vol_circ(4*class_width,1,j,forest_managed(i,j)) .AND. vol_ij <
   vol_circ(5*class_width,1,j,forest_managed(i,j)))),&
+                          & sum(vol_ij,(vol_ij > vol_circ(5*class_width,1,j,forest_managed(i,j)) .AND. vol_ij <
   vol_circ(6*class_width,1,j,forest_managed(i,j)))),&
```

```
+                          & sum(vol_ij,(vol_ij > vol_circ(6*class_width,1,j,forest_managed(i,j)) .AND. vol_ij <
  vol_circ(7*class_width,1,j,forest_managed(i,j)))),&
+                          & sum(vol_ij,(vol_ij > vol_circ(7*class_width,1,j,forest_managed(i,j)) .AND. vol_ij <
  vol_circ(8*class_width,1,j,forest_managed(i,j)))),&
+                          & sum(vol_ij,(vol_ij > vol_circ(8*class_width,1,j,forest_managed(i,j)) .AND. vol_ij <
  vol_circ(9*class_width,1,j,forest_managed(i,j)))),&
+                          & sum(vol_ij,(vol_ij > vol_circ(9*class_width,1,j,forest_managed(i,j)) .AND. vol_ij <
  vol_circ(10*class_width,1,j,forest_managed(i,j)))),&
+                          & sum(vol_ij,(vol_ij > vol_circ(10*class_width,1,j,forest_managed(i,j)))),&
+                          & sum(vol_ij,(vol_ij > 0 .AND. vol_ij < vol_circ((/ class_max(1) /),1,j,forest_managed(i,j)))),&
+                          & sum(vol_ij,(vol_ij > vol_circ((/ class_max(1) /),1,j,forest_managed(i,j)) .AND. &
+                          &             vol_ij < vol_circ((/ class_max(2) /),1,j,forest_managed(i,j)))),&
+                          & sum(vol_ij,(vol_ij > vol_circ((/ class_max(2) /),1,j,forest_managed(i,j)) .AND. &
+                          &             vol_ij < vol_circ((/ class_max(3) /),1,j,forest_managed(i,j)))),&
+                          & sum(vol_ij,(vol_ij > vol_circ((/ class_max(3) /),1,j,forest_managed(i,j))))/)
                    ENDIF

                    IF (bavard_f >=1) print *,'standing_ab_vol',standing_ab_vol(i,j),'exported_bm',exported_bm(i,j),&
@@ -2018,14 +2165,14 @@
     IF (bavard_f >=1) THEN
        print *,'Initial distribution, method 2'
        print *,'Min circ',circ_ij0(1),'Max circ',circ_ij0(nmaxtrees(j)),'Mean circ',sum(circ_ij0)/nmaxtrees(j),&
-            &'Initial volume',vol_circ(circ_ij0,nmaxtrees(j),j),'lambda',lambda,'nmaxtrees(j)',nmaxtrees(j)
+            &'Initial
  volume',vol_circ(circ_ij0,nmaxtrees(j),j,forest_managed(i,j)),'lambda',lambda,'nmaxtrees(j)',nmaxtrees(j)
     ENDIF
     !On définit le root:shoot ratio
     frac_shoot=sum(biomass(i,j,(/isapabove,iheartabove/)))/sum(biomass(i,j,(/isapabove,isapbelow,iheartabove,iheartbelow/)))

 !!$    bm_init(:)=(/0., 0., 0., 0., 0., 0., 0., 0./)
     bm_init(:)=bm_sapl(j,:)
-    bm_init(iheartabove)=max(bm_vol(real(vol_circ(circ_ij0,nmaxtrees(j),j)),j)-&
+    bm_init(iheartabove)=max(bm_vol(real(vol_circ(circ_ij0,nmaxtrees(j),j,forest_managed(i,j))),j,forest_managed(i,j))-&
             &bm_init(isapabove)-bm_init(icarbres)*frac_shoot_init,0.)
     bm_init(iheartbelow)=bm_init(iheartabove)*(1-frac_shoot_init)/frac_shoot_init
     IF (j == 9) THEN
@@ -2033,6 +2180,11 @@
        bm_init(iheartabove)=0.
        bm_init(iheartbelow)=0.
     ENDIF
+! saplings don't have heartwood
+    IF ((j==6).AND.(forest_managed(i,j)==4)) THEN
+       bm_init(:) = bm_sapl(j,:)
+    ENDIF
```

```
     bm_export(:)=(/0., 0., 0., 0., 0., 0., 0., 0./)
 !!$    bm_init(iheartabove)=0.
@@ -2114,14 +2266,14 @@
 !!$    biomass(i,j,icarbres) = biomass(i,j,icarbres)*bm_init
 !!$    biomass(i,j,iroot) = biomass(i,j,iroot)*bm_init !0.0

-end subroutine clearcut
+END SUBROUTINE clearcut

 ! Selects which trees are thinned in order to respect self-thinning rules or rdi_objective
-subroutine thinning(human_th,circ_ij0,ntrees0,psi_rdi,rdi_inf,rdi,circ_lim,npts,i,j,ind,vol_thinned &
+SUBROUTINE thinning(human_th,circ_ij0,ntrees0,psi_rdi,rdi_inf,rdi,circ_lim,npts,i,j,ind,vol_thinned &
     &,height_ij,vol_ij,biomass,frac_shoot,bm_to_litter &
 !!$  ,exported_bm_lastyear
     &,exported_bm &
-    &,ba,av_height,dia,bavard_f,density_rec,circ_class_n_bt,forest_managed,scale_dens,class_width,class_max)
+    &,ba,av_height,av_height_bt,dia,bavard_f,density_rec,circ_class_n_bt,forest_managed,scale_dens,class_width,class_max)
   ! Only trees with circumference lower than circ_lim can be thinned.
   ! 10 harvest classes with the same number of trees in growing order.
   ! The probability of any given tree to be thinned depends on its harvest class.
@@ -2176,7 +2328,7 @@
     ! Stand basal area (m2/ha)
     REAL(r_std),DIMENSION(npts,nvm), INTENT(INOUT)       :: ba
     ! Average height of vegetation (m)
-    REAL(r_std),DIMENSION(npts,nvm), INTENT(INOUT)         :: av_height
+    REAL(r_std),DIMENSION(npts,nvm), INTENT(INOUT)         :: av_height,av_height_bt
     ! mean stem diameter (m)
     REAL(r_std),DIMENSION(npts,nvm), INTENT(INOUT)                :: dia
 !!$    ! exported_bm_lastyear
@@ -2241,21 +2393,21 @@
             count((circ_ij0 >= 0.705 .AND. circ_ij0 < 1.175)),&
             count((circ_ij0 >= 1.175))/)
     ELSE
-       circ_class_n_bt(i,j,:)= (/sum(vol_ij,(vol_ij > 0 .AND. vol_ij < vol_circ(class_width,1,j))),&
-            sum(vol_ij,(vol_ij > vol_circ(class_width,1,j) .AND. vol_ij < vol_circ(2*class_width,1,j))),&
-            sum(vol_ij,(vol_ij > vol_circ(2*class_width,1,j) .AND. vol_ij < vol_circ(3*class_width,1,j))),&
-            sum(vol_ij,(vol_ij > vol_circ(3*class_width,1,j) .AND. vol_ij < vol_circ(4*class_width,1,j))),&
-            sum(vol_ij,(vol_ij > vol_circ(4*class_width,1,j) .AND. vol_ij < vol_circ(5*class_width,1,j))),&
-            sum(vol_ij,(vol_ij > vol_circ(5*class_width,1,j) .AND. vol_ij < vol_circ(6*class_width,1,j))),&
-            sum(vol_ij,(vol_ij > vol_circ(6*class_width,1,j) .AND. vol_ij < vol_circ(7*class_width,1,j))),&
-            sum(vol_ij,(vol_ij > vol_circ(7*class_width,1,j) .AND. vol_ij < vol_circ(8*class_width,1,j))),&
-            sum(vol_ij,(vol_ij > vol_circ(8*class_width,1,j) .AND. vol_ij < vol_circ(9*class_width,1,j))),&
-            sum(vol_ij,(vol_ij > vol_circ(9*class_width,1,j) .AND. vol_ij < vol_circ(10*class_width,1,j))),&
-            sum(vol_ij,(vol_ij > vol_circ(10*class_width,1,j))),&
-            sum(vol_ij,(vol_ij > 0 .AND. vol_ij < vol_circ((/ class_max(1) /),1,j))),&
-            sum(vol_ij,(vol_ij > vol_circ((/ class_max(1) /),1,j) .AND. vol_ij < vol_circ((/ class_max(2) /),1,j))),&
-            sum(vol_ij,(vol_ij > vol_circ((/ class_max(2) /),1,j) .AND. vol_ij < vol_circ((/ class_max(3) /),1,j))),&
-            sum(vol_ij,(vol_ij > vol_circ((/ class_max(3) /),1,j)))/)
+       circ_class_n_bt(i,j,:)= (/sum(vol_ij,(vol_ij > 0 .AND. vol_ij < vol_circ(class_width,1,j,forest_managed(i,j)))),&
+            sum(vol_ij,(vol_ij > vol_circ(class_width,1,j,forest_managed(i,j)) .AND. vol_ij <
   vol_circ(2*class_width,1,j,forest_managed(i,j)))),&
+            sum(vol_ij,(vol_ij > vol_circ(2*class_width,1,j,forest_managed(i,j)) .AND. vol_ij <
   vol_circ(3*class_width,1,j,forest_managed(i,j)))),&
+            sum(vol_ij,(vol_ij > vol_circ(3*class_width,1,j,forest_managed(i,j)) .AND. vol_ij <
   vol_circ(4*class_width,1,j,forest_managed(i,j)))),&
+            sum(vol_ij,(vol_ij > vol_circ(4*class_width,1,j,forest_managed(i,j)) .AND. vol_ij <
   vol_circ(5*class_width,1,j,forest_managed(i,j)))),&
+            sum(vol_ij,(vol_ij > vol_circ(5*class_width,1,j,forest_managed(i,j)) .AND. vol_ij <
   vol_circ(6*class_width,1,j,forest_managed(i,j)))),&
+            sum(vol_ij,(vol_ij > vol_circ(6*class_width,1,j,forest_managed(i,j)) .AND. vol_ij <
   vol_circ(7*class_width,1,j,forest_managed(i,j)))),&
+            sum(vol_ij,(vol_ij > vol_circ(7*class_width,1,j,forest_managed(i,j)) .AND. vol_ij <
   vol_circ(8*class_width,1,j,forest_managed(i,j)))),&
+            sum(vol_ij,(vol_ij > vol_circ(8*class_width,1,j,forest_managed(i,j)) .AND. vol_ij <
   vol_circ(9*class_width,1,j,forest_managed(i,j)))),&
+            sum(vol_ij,(vol_ij > vol_circ(9*class_width,1,j,forest_managed(i,j)) .AND. vol_ij <
   vol_circ(10*class_width,1,j,forest_managed(i,j)))),&
+            sum(vol_ij,(vol_ij > vol_circ(10*class_width,1,j,forest_managed(i,j)))),&
+            sum(vol_ij,(vol_ij > 0 .AND. vol_ij < vol_circ((/ class_max(1) /),1,j,forest_managed(i,j)))),&
+            sum(vol_ij,(vol_ij > vol_circ((/ class_max(1) /),1,j,forest_managed(i,j)) .AND. vol_ij < vol_circ((/
   class_max(2) /),1,j,forest_managed(i,j)))),&
+            sum(vol_ij,(vol_ij > vol_circ((/ class_max(2) /),1,j,forest_managed(i,j)) .AND. vol_ij < vol_circ((/
   class_max(3) /),1,j,forest_managed(i,j)))),&
+            sum(vol_ij,(vol_ij > vol_circ((/ class_max(3) /),1,j,forest_managed(i,j))))/)
     ENDIF

     ! Determination of circ_lim
@@ -2396,7 +2550,7 @@
           DO WHILE (k <= ntrees0)

              IF (ran(idum) < (tau(k)*adjust_tau) .AND. circ_ij0(k) > 0) THEN
-                call harvest(npts,i,j,k,ind,vol_thinned,circ_ij0,ntrees0,rdi,scale_dens)
+                call harvest(npts,i,j,k,ind,vol_thinned,circ_ij0,ntrees0,rdi,scale_dens,forest_managed)
             ENDIF

             IF (rdi .LE. rdi_inf) exit
@@ -2460,8 +2614,9 @@

     l=k
     DO l=k,ntrees0
```

```
-       vol_ij(l)=vol_circ((/ circ_ij0(l) /),1,j)
-       height_ij(l)=height_circ(circ_ij0(l),ba(i,j),nint(ntrees0*10.**scale_dens),j)
+       vol_ij(l)=vol_circ((/ circ_ij0(l) /),1,j,forest_managed(i,j))
+       height_ij(l)=height_circ(circ_ij0(l),ba(i,j),nint(ntrees0*10.**scale_dens),j,forest_managed(i,j))
+print *,"thinning: height_ij(l)",height_ij(l)
     ENDDO


@@ -2503,27 +2658,31 @@
           ENDIF

           IF (forest_managed(i,j)==4) THEN
-
-             bm_init(icarbres)=50.*frac_shoot_init+biomass(i,j,icarbres)*(1-frac_shoot)
-             bm_init(isapabove)=bm_vol(real(vol_circ(circ_ij0,size(circ_ij0),j)),j)*10.**scale_dens-50.*frac_shoot_init
+!             bm_init(icarbres)=50.*frac_shoot_init+biomass(i,j,icarbres)*(1-frac_shoot)
+!             bm_init(isapabove)=bm_vol(real(vol_circ(circ_ij0,size(circ_ij0),j,forest_managed(i,j))),j,
   forest_managed(i,j))*10.**scale_dens-50.*frac_shoot_init
+!
+!             ! If carbres or sapwood above is not enough for regrowth, fill with heartabove
+!             IF (biomass(i,j,icarbres)*frac_shoot<bm_init(icarbres)*frac_shoot_init) THEN
+!            biomass(i,j,iheartabove)=max(biomass(i,j,iheartabove)-bm_init(icarbres)&
+!               &*frac_shoot_init+biomass(i,j,icarbres)*frac_shoot,0.)
+!            biomass(i,j,icarbres)=biomass(i,j,icarbres)+bm_init(icarbres)&
+!               &*frac_shoot_init-biomass(i,j,icarbres)*frac_shoot
+!             ENDIF
+!
+!             IF (biomass(i,j,isapabove)<bm_init(isapabove)) THEN
+!                biomass(i,j,iheartabove)=max(biomass(i,j,iheartabove)-bm_init(isapabove)+biomass(i,j,isapabove),0.)
+!                biomass(i,j,isapabove)=bm_init(isapabove)
+!             ENDIF

-             ! If carbres or sapwood above is not enough for regrowth, fill with heartabove
-             IF (biomass(i,j,icarbres)*frac_shoot<bm_init(icarbres)*frac_shoot_init) THEN
-            biomass(i,j,iheartabove)=max(biomass(i,j,iheartabove)-bm_init(icarbres)&
-               &*frac_shoot_init+biomass(i,j,icarbres)*frac_shoot,0.)
-            biomass(i,j,icarbres)=biomass(i,j,icarbres)+bm_init(icarbres)&
-               &*frac_shoot_init-biomass(i,j,icarbres)*frac_shoot
-             ENDIF
+             bm_init(icarbres)=biomass(i,j,icarbres)*(1-frac_shoot)
+             bm_init(isapabove)=bm_vol(sum((0.1*circ_ij0**2)/(4*pi)),j,forest_managed(i,j))

-             IF (biomass(i,j,isapabove)<bm_init(isapabove)) THEN
-                biomass(i,j,iheartabove)=max(biomass(i,j,iheartabove)-bm_init(isapabove)+biomass(i,j,isapabove),0.)
-                biomass(i,j,isapabove)=bm_init(isapabove)
-             ENDIF
-
              ! All above-ground biomass is exported
              bm_export((/iheartabove,ileaf,ifruit/))=biomass(i,j,(/iheartabove,ileaf,ifruit/))
              bm_export(isapabove)=(biomass(i,j,isapabove)-bm_init(isapabove))
-             bm_export(icarbres)=biomass(i,j,icarbres)*frac_shoot-bm_init(icarbres)*frac_shoot_init
+!             bm_export(icarbres)=biomass(i,j,icarbres)*frac_shoot-bm_init(icarbres)*frac_shoot_init
+             bm_export(icarbres)=biomass(i,j,icarbres)*frac_shoot
 !!$             exported_bm(i,j)=exported_bm(i,j)+sum(bm_export)
              exported_bm(i,j)=sum(bm_export)
 !!$             ! All above-ground biomass is exported
@@ -2576,25 +2735,33 @@
     ENDIF

     ba(i,j)=sum((circ_ij0**2)/(4*pi))*10.**scale_dens !new ba, calculated for next year m²/ha
+!                   IF (forest_managed(i,j)==4 .AND. j==6) THEN
+!                      lai_max(j)=ba(i,j)*0.5
+!                   ENDIF

     !new average height (m)
+    av_height_bt(i,j) = av_height(i,j)
     av_height(i,j)=sum(height_ij, height_ij > 0)/count(height_ij > 0)

     !new mean diameter (m)
     dia(i,j)=sum(circ_ij0/pi)/(count(circ_ij0 .GT. 0.0))

     IF (bavard_f >=1) THEN
-       print *,'vol_ij after thinning',sum(vol_ij),vol_circ(circ_ij0,ntrees0,j),&
+       print *,'vol_ij after thinning',sum(vol_ij),vol_circ(circ_ij0,ntrees0,j,forest_managed(i,j)),&
              &'vol_thinned',vol_thinned,'circ_ij<=0',count(circ_ij0<=0.)
        print *,"bm_init",bm_init
        print *,"bm_export",bm_export
        print *,"biomass(i,j,:)",biomass(i,j,:)
     ENDIF

-end subroutine thinning
+END SUBROUTINE thinning

 ! Harvests one tree (its circumference gets to 0, its biomass is exported, ...)
-subroutine harvest(npts,i,j,k,ind,vol_thinned,circ_ij,ntrees,rdi,scale_dens)
+SUBROUTINE harvest(npts,i,j,k,ind,vol_thinned,circ_ij,ntrees,rdi,scale_dens,fm)
     ! Domain size
     INTEGER(i_std), INTENT(in)                                 :: npts
     ! counters
@@ -2605,6 +2772,8 @@
```

```
     REAL(r_std), INTENT(in)                                       :: scale_dens
     ! density of individuals (1/(m**2 of nat/agri ground))
     REAL(r_std),DIMENSION(npts,nvm),INTENT(inout)                 :: ind
+    ! forest management type
+    INTEGER(i_std),DIMENSION(npts,nvm),INTENT(in)                 :: fm
     ! volume of thinned trees including waste wood (m3/ha)
     REAL(r_std), INTENT(inout)                                      :: vol_thinned
     ! circumference of all individual trees on 1 ha pts i and pft j (m)
@@ -2619,7 +2788,7 @@
     REAL(r_std)                                                   :: Dg

     ind(i,j)=nint(ind(i,j)*10000-10.**scale_dens)/10000.
-    vol_thinned=vol_thinned+vol_circ((/ circ_ij(k) /),1,j)*10.**scale_dens
+    vol_thinned=vol_thinned+vol_circ((/ circ_ij(k) /),1,j,fm(i,j))*10.**scale_dens
     circ_ij(k)=0.
 !!$    height_ij(k)=0.
 !!$    vol_ij(k)=0.
@@ -2627,9 +2796,9 @@

     rdi=ind(i,j)*10.**(4.-scale_dens)/Nmax(Dg,j,scale_dens)

-end subroutine harvest
+END SUBROUTINE harvest

-subroutine force_load(forcing_vol_inc,nforc_years,nforc_pts,bavard_f,circ_init,dens_max,force_inc,force_init)
+SUBROUTINE force_load(forcing_vol_inc,nforc_years,nforc_pts,bavard_f,circ_init,dens_max,force_inc,force_init)
     ! Number of years of forcing years
     INTEGER(i_std)                    :: nforc_years
     ! Number of years of forcing points
Index: src_stomate/stomate_prescribe.f90
===================================================================
--- src_stomate/stomate_prescribe.f90    (revision 561)
+++ src_stomate/stomate_prescribe.f90    (working copy)
@@ -39,7 +39,7 @@
 !VBMODIF
  SUBROUTINE prescribe (npts, &
                         veget_max, PFTpresent, everywhere, when_growthinit, &
-                        biomass, leaf_frac, ind, cn_ind, &
+                        biomass, leaf_frac, root_frac, ind, cn_ind, &
                         forest_managed)
 !ENDVBMODIF

@@ -66,6 +66,8 @@
     REAL(r_std), DIMENSION(npts,nvm,nparts), INTENT(inout)     :: biomass
     ! fraction of leaves in leaf age class
     REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)  :: leaf_frac
+    ! fraction of roots in root age class
+    REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)  :: root_frac
     ! density of individuals (1/(m**2 of ground))
     REAL(r_std), DIMENSION(npts,nvm), INTENT(inout)            :: ind
     ! crown area of individuals (m**2)
@@ -256,6 +258,11 @@
               ! set leaf age classes
               leaf_frac(i,j,:) = zero
               leaf_frac(i,j,1) = un
+
+              root_frac(i,j,:) = zero
+              root_frac(i,j,1) = un

               ! set time since last beginning of growing season
               when_growthinit(i,j) = large_value
@@ -266,7 +273,10 @@

                 biomass(i,j,ileaf) = 0.0
                 leaf_frac(i,j,1) = 0.0
-
+                biomass(i,j,iroot) = 0.0
+                root_frac(i,j,1) = 0.0
               ENDIF

             ENDIF
@@ -284,6 +294,10 @@
               ! set leaf age classes
               leaf_frac(i,j,:) = 0.0
               leaf_frac(i,j,1) = 1.0
+              root_frac(i,j,:) = 0.0
+              root_frac(i,j,1) = 1.0

               ! set time since last beginning of growing season
               when_growthinit(i,j) = large_value
Index: src_stomate/stomate.f90
===================================================================
--- src_stomate/stomate.f90  (revision 561)
+++ src_stomate/stomate.f90  (working copy)
@@ -191,6 +191,12 @@
   REAL(r_std),ALLOCATABLE,SAVE,DIMENSION(:,:,:)  :: leaf_age
   ! fraction of leaves in leaf age class
   REAL(r_std),ALLOCATABLE,SAVE,DIMENSION(:,:,:)  :: leaf_frac
+  ! root age (d)
+  REAL(r_std),ALLOCATABLE,SAVE,DIMENSION(:,:,:)  :: root_age
+  ! fraction of roots in leaf age class
+  REAL(r_std),ALLOCATABLE,SAVE,DIMENSION(:,:,:)  :: root_frac
   ! How much time ago was the PFT eliminated for the last time (y)
   REAL(r_std),ALLOCATABLE,SAVE,DIMENSION(:,:)    :: RIP_time
   ! duration of dormance (d)
@@ -733,6 +741,9 @@
             &         maxfpc_lastyear, maxfpc_thisyear, &
             &         turnover_longterm, gpp_week, biomass, resp_maint_part, &
             &         leaf_age, leaf_frac, &
+            &         root_age, root_frac, &
             &         senescence, when_growthinit, age, &
             &         resp_hetero_d, resp_maint_d, resp_growth_d, co2_fire, co2_to_bm_dgvm, &
             &         veget_lastlight, everywhere, need_adjacent, RIP_time, time_lowgpp, &
@@ -1083,6 +1094,9 @@
        IF (control%ok_stomate) THEN
           CALL stomate_var_init &
                &         (kjpindex, veget_cov, veget_cov_max, leaf_age, leaf_frac, &
+               &          root_age, root_frac, &
               &          tlong_ref, t2m_month, dead_leaves, &
 !VBADD
                    forest_managed,age_stand, &
@@ -1132,6 +1146,9 @@
             &          maxfpc_lastyear, maxfpc_thisyear, &
             &          turnover_longterm, gpp_week, biomass, resp_maint_part, &
             &          leaf_age, leaf_frac, &
+            &          root_age, root_frac, &
            &          senescence, when_growthinit, age, &
            &          resp_hetero_d, resp_maint_d, resp_growth_d, co2_fire, co2_to_bm_dgvm, &
            &          veget_lastlight, everywhere, need_adjacent, &
@@ -1468,7 +1485,11 @@
                &             turnover_longterm, gpp_daily, time_lowgpp, &
                &             time_hum_min, maxfpc_lastyear, resp_maint_part,&
                &             PFTpresent, age, fireindex, firelitter, &
-               &             leaf_age, leaf_frac, biomass, ind, adapted, regenerate, &
+               &             leaf_age, leaf_frac, &
+               &             root_age, root_frac, &
+               &              biomass, ind, adapted, regenerate, &
               &             senescence, when_growthinit, litterpart, litter, &
              &             dead_leaves, carbon, black_carbon, lignin_struc, &
 ! SZ ADD
@@ -1911,7 +1932,44 @@
     !
    bavard = 1
    CALL getin_p('BAVARD', bavard)
+    CALL getin_p('one1', one1)
+    CALL getin_p('one2', one2)
+    CALL getin_p('one3', one3 )
+
+    CALL getin_p('two1', two1 )
+    CALL getin_p('two2', two2 )
+    CALL getin_p('two3', two3 )
+
+    CALL getin_p('three1', three1 )
+    CALL getin_p('three2', three2 )
+    CALL getin_p('three3', three3 )
+    CALL getin_p('three4', three4 )
+
+    CALL getin_p('four1', four1 )
+    CALL getin_p('four2', four2 )
+    CALL getin_p('four3', four3 )
+    CALL getin_p('four4', four4 )

+write(*,*) "one1=",one1
+write(*,*) "one2=",one2
+write(*,*) "one3=",one3
+
+write(*,*) "two1=",two1
+write(*,*) "two2=",two2
+write(*,*) "two3=",two3
+
+write(*,*) "three1=",three1
+write(*,*) "three2=",three2
+write(*,*) "three3=",three3
+write(*,*) "three4=",three4
+
+write(*,*) "four1=",four1
+write(*,*) "four2=",four2
+write(*,*) "four3=",four3
+write(*,*) "four4=",four4
+
     IF ( kjpindex > 0 ) THEN
        !
        !Config  Key  = STOMATE_DIAGPT
@@ -2090,6 +2148,12 @@
     l_error = l_error .OR. (ier /= 0)
     ALLOCATE(leaf_frac(kjpindex,nvm,nleafages),stat=ier)
     l_error = l_error .OR. (ier /= 0)
+    ALLOCATE(root_age(kjpindex,nvm,nleafages),stat=ier)
+    l_error = l_error .OR. (ier /= 0)
+    ALLOCATE(root_frac(kjpindex,nvm,nleafages),stat=ier)
+    l_error = l_error .OR. (ier /= 0)
ALLOCATE(RIP_time(kjpindex,nvm),stat=ier)
l_error = l_error .OR. (ier /= 0)
ALLOCATE(time_lowgpp(kjpindex,nvm),stat=ier)
@@ -2367,6 +2431,10 @@
IF (ALLOCATED(need_adjacent)) DEALLOCATE(need_adjacent)
IF (ALLOCATED(leaf_age)) DEALLOCATE(leaf_age)
IF (ALLOCATED(leaf_frac)) DEALLOCATE(leaf_frac)
+    IF (ALLOCATED(root_age)) DEALLOCATE(root_age)
+    IF (ALLOCATED(root_frac)) DEALLOCATE(root_frac)
IF (ALLOCATED(RIP_time)) DEALLOCATE(RIP_time)
IF (ALLOCATED(time_lowgpp)) DEALLOCATE(time_lowgpp)
IF (ALLOCATED(time_hum_min)) DEALLOCATE(time_hum_min)
@@ -2503,6 +2571,9 @@
   !
SUBROUTINE stomate_var_init &
&  (kjpindex, veget_cov, veget_cov_max, leaf_age, leaf_frac, &
+       &   root_age, root_frac, &
&   tlong_ref, t2m_month, dead_leaves, &
!VBADD
&   forest_managed,age_stand, &
@@ -2546,6 +2617,12 @@
REAL(r_std),DIMENSION(kjpindex,nvm,nleafages),INTENT(inout) :: leaf_age
! fraction of leaves in leaf age class
REAL(r_std),DIMENSION(kjpindex,nvm,nleafages),INTENT(inout) :: leaf_frac
+    ! root age (d)
+    REAL(r_std),DIMENSION(kjpindex,nvm,nleafages),INTENT(inout) :: root_age
+    ! fraction of roots in leaf age class
+    REAL(r_std),DIMENSION(kjpindex,nvm,nleafages),INTENT(inout) :: root_frac
! output scalar
! output fields
! Maximum water on vegetation for interception
@@ -2581,6 +2658,9 @@
!     only if STOMATE is activated
!
CALL vmax (kjpindex, dt_0, leaf_age, leaf_frac,&
+                 root_age, root_frac, &
!VBADD
                 forest_managed,age_stand, &
!ENDVBADD
Index: src_stomate/lpj_pftinout.f90
===================================================================
--- src_stomate/lpj_pftinout.f90   (revision 561)
+++ src_stomate/lpj_pftinout.f90   (working copy)
@@ -31,7 +31,9 @@
```

```
   SUBROUTINE pftinout (npts, dt, adapted, regenerate, &
        neighbours, veget, veget_max, &
-       biomass, ind, age, leaf_frac, npp_longterm, lm_lastyearmax, senescence, &
+       biomass, ind, age, leaf_frac, root_frac, npp_longterm, lm_lastyearmax, senescence, &
        PFTpresent, everywhere, when_growthinit, need_adjacent, RIP_time, &
        co2_to_bm, &
        avail_tree, avail_grass)
@@ -68,6 +70,8 @@
     REAL(r_std), DIMENSION(npts,nvm), INTENT(inout)           :: age
     ! fraction of leaves in leaf age class
     REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout) :: leaf_frac
+    ! fraction of roots in root age class
+    REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout) :: root_frac
     ! "long term" net primary productivity (gC/(m**2 of ground)/year)
     REAL(r_std), DIMENSION(npts,nvm), INTENT(inout)           :: npp_longterm
     ! last year's maximum leaf mass, for each PFT (gC/(m**2 of ground))
@@ -381,6 +385,9 @@

              ! all leaves are young
              leaf_frac(:,j,1) = un
+             root_frac(:,j,1) = un

              ! non-zero "long term" npp and last year's leaf mass for saplings -
              !   so they won't be killed off by gap or kill
Index: src_stomate/stomate_turnover.f90
===================================================================
--- src_stomate/stomate_turnover.f90     (revision 561)
+++ src_stomate/stomate_turnover.f90     (working copy)
@@ -36,10 +36,14 @@
        herbivores, &
        maxmoiavail_lastyear, minmoiavail_lastyear, &
        moiavail_week, moiavail_month, tlong_ref, t2m_month, t2m_week, veget_max, &
-       leaf_age, leaf_frac, age, lai, biomass, &
+       leaf_age, leaf_frac, &
+       root_age, root_frac, &
+       age, lai, biomass, &
                    turnover, senescence,turnover_time, &
 !VBADD
-                   forest_managed,av_height)
+                   forest_managed,av_height,last_cut)
 !ENDVBADD

    !
@@ -78,6 +82,8 @@
```

```
     INTEGER(i_std), DIMENSION (npts,nvm), INTENT(in)                  :: forest_managed
     ! Average stand height (m)
     REAL(r_std), DIMENSION(npts,nvm),INTENT(in)            :: av_height
+    ! Years since last thinning (years)
+    INTEGER(i_std), DIMENSION(npts,nvm),INTENT(in)  :: last_cut
 !ENDVBADD

     ! 0.2 modified fields
@@ -86,6 +92,12 @@
     REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)  :: leaf_age
     ! fraction of leaves in leaf age class
     REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)  :: leaf_frac
+    ! age of the roots (days)
+    REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)  :: root_age
+    ! fraction of roots in leaf age class
+    REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)  :: root_frac
     ! age (years)
     REAL(r_std), DIMENSION(npts,nvm), INTENT(inout)            :: age
     ! leaf area index
@@ -129,10 +141,20 @@
     REAL(r_std), DIMENSION(npts)                                :: lm_old
     ! leaf mass change for each age class
     REAL(r_std), DIMENSION(npts,nleafages)                      :: delta_lm
+    ! old root mass (gC/(m**2 of ground))
+    REAL(r_std), DIMENSION(npts)                                :: rm_old
+    ! root mass change for each age class
+    REAL(r_std), DIMENSION(npts,nleafages)                      :: delta_rm
     ! turnover rate
     REAL(r_std), DIMENSION(npts)                                :: turnover_rate
     ! critical leaf age (d)
     REAL(r_std), DIMENSION(npts,nvm)                           :: leaf_age_crit
+    ! critical root age (d)
+    REAL(r_std), DIMENSION(npts,nvm)                           :: root_age_crit
     ! instantaneous turnover time
     REAL(r_std), DIMENSION(npts,nvm)                           :: new_turnover_time
 !VBMODIF
@@ -339,15 +361,23 @@

           ! 3.2.1 trees

-          WHERE ( senescence(:,j) )
+          IF (two3 == .FALSE.) THEN
+             WHERE ( senescence(:,j) )

-             turnover(:,j,ileaf) = biomass(:,j,ileaf) * dt / pheno_crit%leaffall(j)
-             turnover(:,j,iroot) = biomass(:,j,iroot) * dt / pheno_crit%leaffall(j)
+                turnover(:,j,ileaf) = biomass(:,j,ileaf) * dt / pheno_crit%leaffall(j)
+                turnover(:,j,iroot) = biomass(:,j,iroot) * dt / pheno_crit%leaffall(j)

-             biomass(:,j,ileaf) = biomass(:,j,ileaf) - turnover(:,j,ileaf)
-             biomass(:,j,iroot) = biomass(:,j,iroot) - turnover(:,j,iroot)
+                biomass(:,j,ileaf) = biomass(:,j,ileaf) - turnover(:,j,ileaf)
+                biomass(:,j,iroot) = biomass(:,j,iroot) - turnover(:,j,iroot)

-          ENDWHERE
+             ENDWHERE
+          ELSE
+             WHERE ( senescence(:,j) )
+                turnover(:,j,ileaf) = biomass(:,j,ileaf) * dt / pheno_crit%leaffall(j)
+                biomass(:,j,ileaf) = biomass(:,j,ileaf) - turnover(:,j,ileaf)
+             ENDWHERE
+          ENDIF

       ELSE

@@ -399,34 +429,85 @@
          ! 4.1.1 critical age: prescribed for trees

          leaf_age_crit(:,j) = pheno_crit%leafagecrit(j)
-
+!IF (j==6) print *,'leaf_age_crit(:,6) = ',leaf_age_crit(:,j)
+          root_age_crit(:,j) = pheno_crit%rootagecrit(j)
+!IF (j==6) print *,'root_age_crit(:,6) = ',root_age_crit(:,j)
          ! 4.1.2 loop over leaf age classes

          DO m = 1, nleafages
             turnover_rate(:) =0
-             WHERE ( leaf_age(:,j,m) .GT. leaf_age_crit(:,j)/2. )
+             IF (two3 == .FALSE.) THEN
+                WHERE ( leaf_age(:,j,m) .GT. leaf_age_crit(:,j)/2. )

-                turnover_rate(:) =  &
-                     MIN( 0.99_r_std, dt / ( leaf_age_crit(:,j) * &
-                     ( leaf_age_crit(:,j) / leaf_age(:,j,m) )**4._r_std ) )
+                   turnover_rate(:) =  &
+                        MIN( 0.99_r_std, dt / ( leaf_age_crit(:,j) * &
+                        ( leaf_age_crit(:,j) / leaf_age(:,j,m) )**4._r_std ) )

-                dturnover(:) = biomass(:,j,ileaf) * leaf_frac(:,j,m) * turnover_rate(:)
-                turnover(:,j,ileaf) = turnover(:,j,ileaf) + dturnover(:)
-                biomass(:,j,ileaf) = biomass(:,j,ileaf) - dturnover(:)
+                   dturnover(:) = biomass(:,j,ileaf) * leaf_frac(:,j,m) * turnover_rate(:)
+                   turnover(:,j,ileaf) = turnover(:,j,ileaf) + dturnover(:)
+                   biomass(:,j,ileaf) = biomass(:,j,ileaf) - dturnover(:)

-                ! save leaf mass change
-                delta_lm(:,m) = - dturnover(:)
+                   ! save leaf mass change
+                   delta_lm(:,m) = - dturnover(:)

-                dturnover(:) = biomass(:,j,iroot) * leaf_frac(:,j,m) * turnover_rate(:)
-                turnover(:,j,iroot) = turnover(:,j,iroot) + dturnover(:)
-                biomass(:,j,iroot) = biomass(:,j,iroot) - dturnover(:)
+                   dturnover(:) = biomass(:,j,iroot) * leaf_frac(:,j,m) * turnover_rate(:)
+                   turnover(:,j,iroot) = turnover(:,j,iroot) + dturnover(:)
+                   biomass(:,j,iroot) = biomass(:,j,iroot) - dturnover(:)

-                dturnover(:) = biomass(:,j,ifruit) * leaf_frac(:,j,m) * turnover_rate(:)
-                turnover(:,j,ifruit) = turnover(:,j,ifruit) + dturnover(:)
-                biomass(:,j,ifruit) = biomass(:,j,ifruit) - dturnover(:)
+                   dturnover(:) = biomass(:,j,ifruit) * leaf_frac(:,j,m) * turnover_rate(:)
+                   turnover(:,j,ifruit) = turnover(:,j,ifruit) + dturnover(:)
+                   biomass(:,j,ifruit) = biomass(:,j,ifruit) - dturnover(:)

-             ENDWHERE
+                ENDWHERE
+             ELSE
+                WHERE ( leaf_age(:,j,m) .GT. leaf_age_crit(:,j)/2. )

+                   turnover_rate(:) =  &
+                        MIN( 0.99_r_std, dt / ( leaf_age_crit(:,j) * &
+                        ( leaf_age_crit(:,j) / leaf_age(:,j,m) )**4._r_std ) )
+
+                   dturnover(:) = biomass(:,j,ileaf) * leaf_frac(:,j,m) * turnover_rate(:)
+                   turnover(:,j,ileaf) = turnover(:,j,ileaf) + dturnover(:)
+                   biomass(:,j,ileaf) = biomass(:,j,ileaf) - dturnover(:)
+
+                   ! save leaf mass change
+                   delta_lm(:,m) = - dturnover(:)
+
+                   dturnover(:) = biomass(:,j,ifruit) * leaf_frac(:,j,m) * turnover_rate(:)
+                   turnover(:,j,ifruit) = turnover(:,j,ifruit) + dturnover(:)
+                   biomass(:,j,ifruit) = biomass(:,j,ifruit) - dturnover(:)
+
+                ENDWHERE
+             ENDIF
+             IF (two3 == .TRUE.) THEN
+                WHERE ( root_age(:,j,m) .GT. root_age_crit(:,j)/2. )
+
+                   turnover_rate(:) =  &
+                        MIN( 0.99_r_std, dt / ( root_age_crit(:,j) * &
+                       ( root_age_crit(:,j) / root_age(:,j,m) )**4._r_std ) )
+
+                   dturnover(:) = biomass(:,j,iroot) * root_frac(:,j,m) * turnover_rate(:)
+                   turnover(:,j,iroot) = turnover(:,j,iroot) + dturnover(:)
+                   biomass(:,j,iroot) = biomass(:,j,iroot) - dturnover(:)
+
+                   ! save root mass change
+                   delta_rm(:,m) = - dturnover(:)
+
+                ENDWHERE
+             ENDIF
           ENDDO

        ELSE
@@ -497,7 +578,25 @@
           ENDWHERE

        ENDDO
+       !
+       ! 4.4 recalculate fraction in each root age class
+       !     new fraction = new root mass of that fraction / new total root mass
+       !                  = ( old fraction*old total root mass + biomass change of that fraction ) /
+       !                    new total root mass
+       !

+       DO m = 1, nleafages
+
+          WHERE ( biomass(:,j,iroot) .GT. 0.0 )
+             root_frac(:,j,m) = ( root_frac(:,j,m)*rm_old(:) + delta_rm(:,m) ) / biomass(:,j,iroot)
+          ELSEWHERE
+             root_frac(:,j,m) = 0.0
+          ENDWHERE
+
+       ENDDO
+
     ENDDO         ! loop over PFTs

     !
@@ -530,27 +629,43 @@
```

```
        IF ( tree(j) .AND. ( pheno_crit%senescence_type(j) .NE. 'none' ) ) THEN

           ! check whether we shed the remaining leaves
+          IF (two3 == .FALSE.) THEN
+             WHERE ( ( biomass(:,j,ileaf) .GT. 0.0 ) .AND. senescence(:,j) .AND. &
+                  ( biomass(:,j,ileaf) .LT. (pheno_crit%lai_initmin(j) / 2.)/sla(j) )               )
+
+                shed_rest(:) = .TRUE.
+
+                turnover(:,j,ileaf) = turnover(:,j,ileaf) + biomass(:,j,ileaf)
+                turnover(:,j,iroot) = turnover(:,j,iroot) + biomass(:,j,iroot)
+                turnover(:,j,ifruit) = turnover(:,j,ifruit) + biomass(:,j,ifruit)

-          WHERE ( ( biomass(:,j,ileaf) .GT. 0.0 ) .AND. senescence(:,j) .AND. &
-               ( biomass(:,j,ileaf) .LT. (pheno_crit%lai_initmin(j) / 2.)/sla(j) )               )
+                biomass(:,j,ileaf) = 0.0
+                biomass(:,j,iroot) = 0.0
+                biomass(:,j,ifruit) = 0.0

-             shed_rest(:) = .TRUE.
+                ! reset leaf age
+                leaf_meanage(:,j) = 0.0

-             turnover(:,j,ileaf) = turnover(:,j,ileaf) + biomass(:,j,ileaf)
-             turnover(:,j,iroot) = turnover(:,j,iroot) + biomass(:,j,iroot)
-             turnover(:,j,ifruit) = turnover(:,j,ifruit) + biomass(:,j,ifruit)
+             ENDWHERE
+          ELSE
+             WHERE ( ( biomass(:,j,ileaf) .GT. 0.0 ) .AND. senescence(:,j) .AND. &
+                  ( biomass(:,j,ileaf) .LT. (pheno_crit%lai_initmin(j) / 2.)/sla(j) )               )
+
+                shed_rest(:) = .TRUE.
+
+                turnover(:,j,ileaf) = turnover(:,j,ileaf) + biomass(:,j,ileaf)
+                turnover(:,j,ifruit) = turnover(:,j,ifruit) + biomass(:,j,ifruit)

-             biomass(:,j,ileaf) = 0.0
-             biomass(:,j,iroot) = 0.0
-             biomass(:,j,ifruit) = 0.0
+                biomass(:,j,ileaf) = 0.0
+                biomass(:,j,ifruit) = 0.0

+                ! reset leaf age
+                leaf_meanage(:,j) = 0.0
+             ENDWHERE
+          ENDIF

-             ! reset leaf age
-             leaf_meanage(:,j) = 0.0
-
-          ENDWHERE
-
       ENDIF

       !
@@ -736,7 +851,14 @@

          ! above the ground

-          sapconv(:) = biomass(:,j,isapabove) * dt / tau_sap(j)
+          IF ((last_cut(i,j)<5) .AND. (forest_managed(i,j)==4) .AND. (three2 == .TRUE.)) THEN
+             sapconv(:) = 0
+          ELSE
+             sapconv(:) = biomass(:,j,isapabove) * dt / tau_sap(j)
+          ENDIF
+!           sapconv(:) = biomass(:,j,isapabove) * dt / tau_sap(j)
          biomass(:,j,isapabove) = biomass(:,j,isapabove) - sapconv(:)
          biomass(:,j,iheartabove) =  biomass(:,j,iheartabove) + sapconv(:)

Index: src_stomate/stomate_lpj.f90
===================================================================
--- src_stomate/stomate_lpj.f90    (revision 561)
+++ src_stomate/stomate_lpj.f90    (working copy)
@@ -84,7 +84,11 @@
        turnover_longterm, gpp_daily, time_lowgpp, &
        time_hum_min, maxfpc_lastyear, resp_maint_part, &
        PFTpresent, age, fireindex, firelitter, &
-       leaf_age, leaf_frac, biomass, ind, adapted, regenerate, &
+       leaf_age, leaf_frac, &
+       root_age, root_frac, &
+       biomass, ind, adapted, regenerate, &
        senescence, when_growthinit, &
        litterpart, litter, dead_leaves, carbon, black_carbon, lignin_struc, &
 ! SZ ADD
@@ -199,6 +203,12 @@
     REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)     :: leaf_age
     ! fraction of leaves in leaf age class
     REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)     :: leaf_frac
+    ! root age (days)
+    REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)     :: root_age
+    ! fraction of roots in leaf age class
+    REAL(r_std), DIMENSION(npts,nvm,nleafages), INTENT(inout)     :: root_frac
     ! biomass (gC/(m**2 of ground))
     REAL(r_std), DIMENSION(npts,nvm,nparts), INTENT(inout)        :: biomass
     ! density of individuals (1/(m**2 of ground))
@@ -346,6 +356,8 @@
     REAL(r_std),DIMENSION(npts,nvm)            :: dom_height
     ! Average height in the stand (m)
     REAL(r_std),DIMENSION(npts,nvm),INTENT(inout)          :: av_height
+    ! Average height in the stand before thinning (m)
+    REAL(r_std),DIMENSION(npts,nvm)            :: av_height_bt
     ! Standing aboveground biomass (gC/m2)
     REAL(r_std),DIMENSION(npts,nvm)            :: standing_ab_bm
     ! Annual aboveground volume increment (m3/ha/yr)
@@ -532,7 +544,9 @@

     CALL prescribe (npts, &
          veget_max, PFTpresent, everywhere, when_growthinit, &
-         biomass, leaf_frac, ind, cn_ind, &
+         biomass, leaf_frac, root_frac, ind, cn_ind, &
         forest_managed)
     !
     ! 2 climatic constraints for PFT presence and regenerativeness
@@ -556,7 +570,9 @@

        CALL pftinout (npts, dt_days, adapted, regenerate, &
             neighbours, veget, veget_max, &
-            biomass, ind, age, leaf_frac, npp_longterm, lm_lastyearmax, senescence, &
+            biomass, ind, age, leaf_frac, root_frac, npp_longterm, lm_lastyearmax, senescence, &
             PFTpresent, everywhere, when_growthinit, need_adjacent, RIP_time, &
             co2_to_bm, &
             avail_tree, avail_grass)
@@ -570,6 +586,9 @@
        CALL kill (npts, 'pftinout  ', lm_lastyearmax, &
             ind, PFTpresent, cn_ind, biomass, senescence, RIP_time, &
             lai, age, leaf_age, leaf_frac, &
+            root_age, root_frac, &
             when_growthinit, everywhere, veget, veget_max, bm_to_litter)

        !
@@ -619,6 +638,9 @@
          gdd_m5_dormance, gdd_midwinter, ncd_dormance, ngd_minus5, &
         senescence, time_lowgpp, time_hum_min, &
         biomass, leaf_frac, leaf_age, &
```

```
+          root_age, root_frac, &
           when_growthinit, co2_to_bm, lai)

     !
@@ -629,7 +651,11 @@
     CALL alloc (npts, dt_days, &
           lai, veget_max, senescence, when_growthinit, &
           moiavail_week, tsoil_month, soilhum_month, &
-          biomass, age, leaf_age, leaf_frac, rprof, f_alloc, &
+          biomass, age, leaf_age, leaf_frac, &
+          root_age, root_frac, &
+          rprof, f_alloc, &
           PFTpresent, height, bm_alloc, delta_bm, sapwood_age, &
           ls_longterm, EndOfYear, sum_sapconv, sum_sapalloc, &
           biomass_lastyear,lr_longterm,sum_rootm,lai_max_calc, &
@@ -645,7 +671,11 @@
           PFTpresent, &
           tlong_ref, t2m_daily, tsoil_daily, lai, rprof, &
           gpp_daily, f_alloc, bm_alloc, resp_maint_part,&
-          biomass, leaf_age, leaf_frac, age, &
+          biomass, leaf_age, leaf_frac, &
+          root_age, root_frac, &
+          age, &
           resp_maint, resp_growth, npp_daily)

     ! VBADD
@@ -658,7 +688,7 @@
                            , when_growthinit, leaf_frac, &
 !gpp_lastyear, gpp_sofar, &
     veget_max, &
-                           sigma, gamma, age_stand, rotation_n, last_cut, av_circ, ba, dom_height, av_height, &
+                           sigma, gamma, age_stand, rotation_n, last_cut, av_circ, ba, dom_height, av_height, av_height_bt, &
                            med_height,q1_height,q3_height, &
                            standing_ab_bm, deltavol, standing_ab_vol, harvestable_vol, harvestable_vol_inc, exported_bm, &
                            exported_vol, exported_vol_th, circ_lim, circ_min, circ_max, &
@@ -698,6 +728,9 @@
        CALL kill (npts, 'fire      ', lm_lastyearmax, &
             ind, PFTpresent, cn_ind, biomass, senescence, RIP_time, &
             lai, age, leaf_age, leaf_frac, &
+            root_age, root_frac, &
             when_growthinit, everywhere, veget, veget_max, bm_to_litter)

     ENDIF
@@ -723,6 +756,9 @@
        CALL kill (npts, 'gap       ', lm_lastyearmax, &
             ind, PFTpresent, cn_ind, biomass, senescence, RIP_time, &
             lai, age, leaf_age, leaf_frac, &
+            root_age, root_frac, &
             when_growthinit, everywhere, veget, veget_max, bm_to_litter)

     ENDIF
@@ -733,6 +769,9 @@

     CALL vmax (npts, dt_days, &
          leaf_age, leaf_frac, &
+         root_age, root_frac, &
!VBADD
          forest_managed,age_stand, &
 !ENDVBADD
@@ -749,11 +788,15 @@
          herbivores, &
          maxmoiavail_lastyear, minmoiavail_lastyear, &
          moiavail_week,  moiavail_month,tlong_ref, t2m_month, t2m_week, veget_max, &
-         leaf_age, leaf_frac, age, lai, biomass, &
+         leaf_age, leaf_frac, &
+         root_age, root_frac, &
+         age, lai, biomass, &
                turnover_daily, senescence,turnover_time, &
 !VBADD
 !!$                    delta_bm, bm_alloc,
-              forest_managed,av_height)
+              forest_managed,av_height,last_cut)
 !ENDVBADD

     !
@@ -777,6 +820,9 @@
        CALL kill (npts, 'light     ', lm_lastyearmax, &
             ind, PFTpresent, cn_ind, biomass, senescence, RIP_time, &
             lai, age, leaf_age, leaf_frac, &
+             root_age, root_frac, &
             when_growthinit, everywhere, veget, veget_max, bm_to_litter)

     ENDIF
@@ -796,6 +842,9 @@
             precip_lastyear, gdd0_lastyear, lm_lastyearmax, &
             cn_ind, lai, avail_tree, avail_grass, &
             leaf_age, leaf_frac, &
+            root_age, root_frac, &
             ind, biomass, age, everywhere, co2_to_bm, veget_max)
        !
@@ -840,7 +889,9 @@
                co2_to_bm, bm_to_litter, turnover_daily, bm_sapl, tree, cn_ind,flux10,flux100, &
 !!$               prod10,prod100,prod10_total,prod100_total,&
 !!$               convflux,cflux_prod_total,cflux_prod10,cflux_prod100,leaf_frac,&
-               prod10,prod100,convflux,cflux_prod10,cflux_prod100,leaf_frac,&
+               prod10,prod100,convflux,cflux_prod10,cflux_prod100,leaf_frac,root_frac,&
                npp_longterm, lm_lastyearmax, litter, carbon)

        ENDIF
@@ -1133,6 +1184,10 @@
             dom_height, npts*nvm, horipft_index) !modifTL
        CALL histwrite (hist_id_stomate, 'AV_HEIGHT', itime, &
             av_height, npts*nvm, horipft_index)
+       CALL histwrite (hist_id_stomate, 'AV_HEIGHT_BT', itime, &
+            av_height_bt, npts*nvm, horipft_index)
        CALL histwrite (hist_id_stomate, 'MED_HEIGHT', itime, &
          !           365*med_height, npts*nvm, horipft_index)
             med_height, npts*nvm, horipft_index) !modifTL
Index: src_sechiba/hydrolc.f90
===================================================================
--- src_sechiba/hydrolc.f90  (revision 512)
+++ src_sechiba/hydrolc.f90  (working copy)
@@ -2150,13 +2157,19 @@
          IF ( gqsb(ji,jv)+bqsb(ji,jv) .GT. zero ) THEN
             !
             IF (dsg(ji,jv).EQ. zero .OR. gqsb(ji,jv).EQ.zero) THEN
-               humrel(ji,jv) = EXP( - humcste(jv) * dpu_cste * (dsp(ji,jv)/dpu_cste) )
+!               humrel(ji,jv) = EXP( - humcste(jv) * dpu_cste * (dsp(ji,jv)/dpu_cste) )
+!               humrel(ji,jv) = EXP( - humcste(jv) * dsp(ji,jv) / (1.-qwilt/mx_eau_eau) )
+               humrel(ji,jv) = EXP( - humcste(jv) * dpu_cste * dsp(ji,jv) / ( dpu_cste-qwilt/(mx_eau_eau*dpu_cste) ) )
                dsg(ji,jv) = zero
                !
                ! if the dry soil height is larger than the one corresponding
                ! to the wilting point, or negative lower soil moisture : humrel is 0.0
                !
-               IF (dsp(ji,jv).GT.(dpu_cste - (qwilt / ruu_ch(ji))) .OR. bqsb(ji,jv).LT.zero) THEN
+! now qwilt is recalculated for the soil depth (qwilt is in kg/m3 instead of kg/m2
+               IF (dsp(ji,jv).GT.(dpu_cste - (qwilt*dpu_cste / ruu_ch(ji))) .OR. bqsb(ji,jv).LT.zero) THEN
                   humrel(ji,jv) = zero
                ENDIF
                !
@@ -2174,8 +2187,14 @@
                ! As we need a slower variable for vegetation growth the stress is computed
                ! differently than in humrel.
                !
-               zhumrel_lo(ji) = EXP( - humcste(jv) * dsp(ji,jv))
-               zhumrel_up(ji) = EXP( - humcste(jv) * dss(ji,jv))
+!               zhumrel_lo(ji) = EXP( - humcste(jv) * dsp(ji,jv))
+!               zhumrel_up(ji) = EXP( - humcste(jv) * dss(ji,jv))
+!               zhumrel_lo(ji) = EXP( - humcste(jv) * dsp(ji,jv) / (1.-qwilt/mx_eau_eau) )
+!               zhumrel_up(ji) = EXP( - humcste(jv) * dss(ji,jv) / (1.-qwilt/mx_eau_eau) )
+               zhumrel_lo(ji) = EXP( - humcste(jv) * dpu_cste * dsp(ji,jv) / ( dpu_cste-qwilt/(mx_eau_eau*dpu_cste) ) )
+               zhumrel_up(ji) = EXP( - humcste(jv) * dpu_cste * dss(ji,jv) / ( dpu_cste-qwilt/(mx_eau_eau*dpu_cste) ) )
                ! Ajouts Nathalie - Fred - le 28 Mars 2006
                a_subgrd(ji,jv)=MIN(MAX(dsg(ji,jv)-dss(ji,jv),0.)/dsg_min,1.)
                humrel(ji,jv)=a_subgrd(ji,jv)*zhumrel_up(ji)+(1.-a_subgrd(ji,jv))*zhumrel_lo(ji)
Index: src_sechiba/slowproc.f90
===================================================================
--- src_sechiba/slowproc.f90 (revision 512)
+++ src_sechiba/slowproc.f90 (working copy)
@@ -760,6 +760,11 @@
     humcste(:)= &
          & (/5., .8, .8, 1., .8, .8, 1., 1., .8, 4., 4., 4., 4./)
     CALL getin_p ("HYDROL_HUMCSTE", humcste)
+  IF (four4 == .TRUE.) THEN
+    CALL getin_p ('humcste_src',humcste(6))
+  ENDIF

 !MM, T. d'O. : before in constantes_soil :
 !          diaglev = &
Index: src_sechiba/enerbil.f90
===================================================================
--- src_sechiba/enerbil.f90  (revision 512)
+++ src_sechiba/enerbil.f90  (working copy)
@@ -239,12 +239,24 @@
        CALL histwrite(hist_id, 'evapot_corr', kjit, evapot_corr, kjpindex, index)
       CALL histwrite(hist_id, 'lwdown', kjit, lwabs,  kjpindex, index)
       CALL histwrite(hist_id, 'lwnet',  kjit, lwnet,  kjpindex, index)
+  CALL histwrite(hist_id, 'rau',  kjit, rau,  kjpindex, index)
+  CALL histwrite(hist_id, 'q_cdrag',  kjit, q_cdrag,  kjpindex, index)
+  CALL histwrite(hist_id, 'psnew',  kjit, psnew,  kjpindex, index)
+  CALL histwrite(hist_id, 'epot_air',  kjit, epot_air_new,  kjpindex, index)
        IF ( hist2_id > 0 ) THEN
          CALL histwrite(hist2_id, 'netrad', kjit, netrad, kjpindex, index)
          CALL histwrite(hist2_id, 'evapot', kjit, evapot, kjpindex, index)
          CALL histwrite(hist2_id, 'evapot_corr', kjit, evapot_corr, kjpindex, index)
          CALL histwrite(hist2_id, 'lwdown', kjit, lwabs,  kjpindex, index)
          CALL histwrite(hist2_id, 'lwnet',  kjit, lwnet,  kjpindex, index)
+     CALL histwrite(hist_id, 'rau',  kjit, rau,  kjpindex, index)
+     CALL histwrite(hist_id, 'q_cdrag',  kjit, q_cdrag,  kjpindex, index)
+     CALL histwrite(hist_id, 'psnew',  kjit, psnew,  kjpindex, index)
+     CALL histwrite(hist_id, 'epot_air',  kjit, epot_air_new,  kjpindex, index)
        ENDIF
     ELSE
        CALL histwrite(hist_id, 'LWnet', kjit, lwnet, kjpindex, index)
Index: src_sechiba/intersurf.f90
===================================================================
--- src_sechiba/intersurf.f90      (revision 512)
+++ src_sechiba/intersurf.f90      (working copy)
@@ -3317,6 +3317,20 @@
             & iim,jjm, hori_id, 1,1,1, -99, 32, avescatter(1), dt,dw)
        CALL histdef(hist_id, 'lai', 'Leaf Area Index', '1', &
             & iim,jjm, hori_id, nvm, 1, nvm, vegax_id, 32, avescatter(1), dt,dw)
+
+       CALL histdef(hist_id, 'rau', 'Density', '1', &
+            & iim,jjm, hori_id, 1, 1, 1, -99, 32, fluxop(1), dt,dw)
+       CALL histdef(hist_id, 'q_cdrag', 'Surface drag', '1', &
+            & iim,jjm, hori_id, 1, 1, 1, -99, 32, fluxop(1), dt,dw)
+       CALL histdef(hist_id, 'psnew', 'New surface static energy', '1', &
+            & iim,jjm, hori_id, 1, 1, 1, -99, 32, fluxop(1), dt,dw)
+       CALL histdef(hist_id, 'epot_air', 'Air potential energy', '1', &
+            & iim,jjm, hori_id, 1, 1, 1, -99, 32, fluxop(1), dt,dw)
+!       CALL histdef(hist_id, 'zri', 'stable or unstable', '1', &
+!            & iim,jjm, hori_id, 1, 1, 1, -99, 32, fluxop(1), dt,dw)
+
        IF ( control_flags%river_routing ) THEN
           CALL histdef(hist_id, 'basinmap', 'Aproximate map of the river basins', ' ', &
                & iim,jjm, hori_id, 1,1,1, -99, 32, once(1), dt,dw)
@@ -5487,6 +5501,10 @@
         & iim, jjm, hist_hori_id, nvm,1,nvm, hist_PFTaxis_id, 32, 'once', dt,hist_dt)
     CALL histdef(hist_id_stom, 'AV_HEIGHT', 'Average height of stand', 'm', &
         & iim, jjm, hist_hori_id, nvm,1,nvm, hist_PFTaxis_id, 32, 'once', dt,hist_dt)
+    CALL histdef(hist_id_stom, 'AV_HEIGHT_BT', 'Average height of stand before thinning', 'm', &
+        & iim, jjm, hist_hori_id, nvm,1,nvm, hist_PFTaxis_id, 32, 'once', dt,hist_dt)
! added by TANKUN 2010/11/30
    CALL histdef(hist_id_stom, 'MED_HEIGHT', 'Median height of stand', 'm', &
        & iim, jjm, hist_hori_id, nvm,1,nvm, hist_PFTaxis_id, 32, 'once', dt,hist_dt)
Index: src_parameters/constantes_veg.f90
===================================================================
--- src_parameters/constantes_veg.f90    (revision 512)
+++ src_parameters/constantes_veg.f90    (working copy)
@@ -154,18 +156,25 @@
  & (/0.0, 12.E-5, 12.E-5, 12.e-5, 12.e-5, 25.e-5, 12.e-5,&
        & 25.e-5, 25.e-5, 30.e-5, 30.e-5, 30.e-5, 30.e-5 /)
```

```
 !-
 ! Maximum field capacity for each of the vegetations (Temporary).
 ! Value of wmax_veg : max quantity of water :
 ! one for each vegetation type en Kg/M3
   REAL(r_std),DIMENSION(nvm),SAVE :: wmax_veg = &
  & (/ 150., 150., 150., 150., 150., 150., 150.,&
  &    150., 150., 150., 150., 150., 150. /)
+! & (/ 100., 100., 100., 100., 100., 100., 100.,&
+! &    100., 100., 100., 100., 100., 100. /)
!-
 ! Root profile description for the different vegetation types.
 ! These are the factor in the exponential which gets
 ! the root density as a function of depth
   REAL(r_std),DIMENSION(nvm), SAVE :: humcste = &
  & (/5., .8, .8, 1., .8, .8, 1., 1., .8, 4., 4., 4., 4./)
+!TDG
+! & (/5., .8, .8, 1., .8, 10., 1., 1., .8, 4., 4., 4., 4./)
+!endTDG
 !-
 ! Type of behaviour of the LAI evolution algorithm
 ! for each vegetation type.
Index: src_parameters/constantes_soil.f90
===================================================================
--- src_parameters/constantes_soil.f90   (revision 512)
+++ src_parameters/constantes_soil.f90   (working copy)
@@ -64,10 +64,12 @@
 !-
 ! Constantes from the Choisnel hydrology
 !-
-! Wilting point (Has a numerical role for the moment)
-  REAL(r_std),PARAMETER :: qwilt = 5.0
+! Wilting point (Has a numerical role for the moment) (kg/m3)
+  REAL(r_std),PARAMETER :: qwilt = 50.0
! Total depth of soil reservoir (for hydrolc)
  REAL(r_std),SAVE :: dpu_cste = 2.0_r_std
 ! The minimal size we allow for the upper reservoir (m)
   REAL(r_std),PARAMETER :: min_resdis = 2.e-5
 ! Diffusion constant for the slow regime
@@ -79,8 +81,10 @@
   REAL(r_std),PARAMETER :: exp_drain = 1.5
 ! Transforms leaf area index into size of interception reservoir
   REAL(r_std),SAVE      :: qsintcst = 0.1
! Maximum quantity of water (Kg/M3)
  REAL(r_std),PARAMETER :: mx_eau_eau = 150.
!-
```

```
 ! Constant in the computation of resistance for bare  soil evaporation
   REAL(r_std),PARAMETER :: rsol_cste = 33.E3
Index: src_parameters/constantes.f90
===================================================================
--- src_parameters/constantes.f90  (revision 512)
+++ src_parameters/constantes.f90  (working copy)
@@ -15,6 +15,26 @@
 !-
 ! Unit for output messages
   INTEGER(i_std), SAVE :: numout = 6
+  LOGICAL,SAVE :: one1
+  LOGICAL,SAVE :: one2
+  LOGICAL,SAVE :: one3
+
+  LOGICAL,SAVE :: two1
+  LOGICAL,SAVE :: two2
+  LOGICAL,SAVE :: two3
+
+  LOGICAL,SAVE :: three1
+  LOGICAL,SAVE :: three2
+  LOGICAL,SAVE :: three3
+  LOGICAL,SAVE :: three4
+
+  LOGICAL,SAVE :: four1
+  LOGICAL,SAVE :: four2
+  LOGICAL,SAVE :: four3
+  LOGICAL,SAVE :: four4
+
!-
 ! To set for more printing
   LOGICAL,SAVE :: long_print = .FALSE.
```