```
**********************************************************************
       SUBROUTINE PERMA_LOC
ckc:   Purpose: Routine to calculate permafrost fraction using Frost Index
ckc:            from monthly surface temperature. From Nelson and Outcalt
ckc:            1987 for Frost index.
ckc:
ckc:   Initially no snow correction
ckc:
ckc:   K. CRICHTON - 13/06/2012
**********************************************************************
        use declar_mod
        use svat_mod
        use params_mod
        use buffer_mod
        use bio_mod


        INTEGER :: month

c reset indices

        freeze(lat,lon)=0.
        thaw(lat,lon)=0.

ckc sum degree-months above zero for thaw index and below zero for freeze
index
c corrected for snow fraction and height, linear model based on Taras et al
c 2002 measurement data

ckc Version: No snow correction
c *************************
c
c        DO month=1,12
c         IF (TATMSMON2(lat,lon,month).LT.0) THEN
c          freeze(lat,lon)=freeze(lat,lon)+TATMSMON2(lat,lon,month)
c         ELSE IF (TATMSMON2(lat,lon,month).GT.0) THEN
c          thaw(lat,lon)=thaw(lat,lon)+TATMSMON2(lat,lon,month)
c         ENDIF
c        ENDDO
c
c **************************
ckc end of no snow correction


ckc Version: Snow corrected
c *************************

ckc        IF (NYR.eq.1) THEN
ckc        p_frac(lat,lon)=FRPRM(lat,lon)
ckc        ELSE
        DO month=1,12

ckc        print *, "temp_month : ", TATMSMON2(lat,lon,month)
           IF (TATMSMON2(lat,lon,month).LT.0) THEN
            IF (TATMSMON2(lat,lon,month).GT.-180.) THEN !-6*30 is -180
ckc             IF (TATMSMON2(lat,lon,month).GT.-0.) THEN
             cold(lat,lon)=TATMSMON2(lat,lon,month)/30.
             ELSE
             cold(lat,lon)=FRSNWMON2(lat,lon,month)*((0.01*(-6.
     >         -(TATMSMON2(lat,lon,month)/30.))*HSNWMON2(lat,lon,month))
     >            +(TATMSMON2(lat,lon,month)/30.))
```

```
ckc              IF (cold(lat,lon).GT.-6.) THEN
ckc                 cold(lat,lon)=-6.
ckc               ENDIF
             ENDIF
ckc         print *, "original temp ", TATMSMON2(lat,lon,month)/30.
ckc         print *, "corrected temp ", cold(lat,lon)
             freeze(lat,lon)=freeze(lat,lon)+cold(lat,lon)
           ELSE IF (TATMSMON2(lat,lon,month).GT.0) THEN
             thaw(lat,lon)=thaw(lat,lon)+(TATMSMON2(lat,lon,month)/30.)
           ENDIF
        ENDDO
ckc        ENDIF

c ****************************
ckc end of snow corrected


ckc for magnitude of freeze
        freeze(lat,lon) = -1*(freeze(lat,lon))

ckc Calculate frost index, snow corrected

        fr_ndx(lat,lon)=SQRT(freeze(lat,lon))/(SQRT(freeze(lat,lon))
     >    +SQRT(thaw(lat,lon)))

c        print *,"frost index : ", fr_ndx(lat,lon)

ckc use frost index to permafrost fraction relationship developed by k
crichton
c for CLIMBER-2

ckc If using permafrost lag function, uncomment the next line
c       p_frac_old(lat,lon)=p_frac(lat,lon)

ckc permafrost function "LOW"
c       b(lat,lon) = 20*(fr_ndx(lat,lon)-0.6)
c       p_frac_pred(lat,lon) = 0.53*(0.976+(b(lat,lon)/
c     >       (SQRT(1+(b(lat,lon)**2)))))-0.015

ckc permafrost function "LOW-MED" !for deglaciation paper
       b(lat,lon) = 20.5*(fr_ndx(lat,lon)-0.595)
       p_frac_pred(lat,lon) = 0.54*(0.976+(b(lat,lon)/
     >       (SQRT(1+(b(lat,lon)**2)))))-0.015

ckc permafrost function "MED"
c       b(lat,lon) = 21*(fr_ndx(lat,lon)-0.59)
c       p_frac_pred(lat,lon) = 0.555*(0.976+(b(lat,lon)/
c     >       (SQRT(1+(b(lat,lon)**2)))))-0.015

ckc permafrost function "HIGH"
ckc    function "3" 25 oct 2012
c        b(lat,lon) = 22*(fr_ndx(lat,lon)-0.58)
ckc        print *,"B(lat,lon) : ", b(lat,lon)
c        p_frac_pred(lat,lon) = 0.58*(0.976+(b(lat,lon)/
c     >       (SQRT(1+(b(lat,lon)**2)))))-0.015


        IF(p_frac_pred(lat,lon).GT.1) THEN
            p_frac_pred(lat,lon) = 1
        ELSE IF(p_frac_pred(lat,lon).LT.0) THEN
            p_frac_pred(lat,lon) = 0
```

```
        ENDIF

ckc for no permafrost lag
        p_frac(lat,lon)=p_frac_pred(lat,lon) !no lag
ckc for permafrost lag function
c        p_frac(lat,lon)=0.5*(p_frac_old(lat,lon)+p_frac_pred(lat,lon))

        w_frac(lat,lon) = 1-p_frac(lat,lon)

c        print *,"perma frac PERM i : ", p_frac(lat,lon), lat
        FRPRM(lat,lon)=p_frac(lat,lon)
        FRWRM(lat,lon)=w_frac(lat,lon)
        FROST(lat,lon)=fr_ndx(lat,lon)
c        print *, "FRPRM : ", FRPRM(lat,lon)
ckc calculate the area of permafrost in each cell
c       parea(lat,lon)=carea(lat,lon)*p_frac(lat,lon)*FRGLC(lat,lon)

       return

       END SUBROUTINE PERMA_LOC
```

```
c*******************************************************************
              SUBROUTINE CCDYN
c*******************************************************************
       use declar_mod
       use params_mod
       use bio_mod
       use buffer_mod

cscript Les declarations suivantes ont ete faite par un script
      REAL G4D
csript ---
c*******************************************************************
c temporal var*/
        REAL tempor1,tempor2,tempor3,tempor4,db2,fd,dst,dd,nld,
     >        dstime,tempor5,tempor6,dsg,dsd,temp_sg,temp_st,
     >         b4t_hold, b4g_hold, b4t14_hold, b4g14_hold,
     >          b4t13_hold, b4g13_hold, b3t_hold, b3g_hold,
     >           b3t14_hold, b3g14_hold, b3t13_hold, b3g13_hold


c calculation of current carbon cycle parameters

        call CCPARAM

cnb        print*, 'npp',lat, lon, npp
cnb        npp_tot=npp_tot+npp
cnb        print*,'npp_tot',npp_tot

cnb - Test sur le carbone pris par la veget
cnb - S il est trop important on divise npp par 2
cnb - et on recalcul tous les reservoirs

cnb - Initialisation pour entrer dans la boucle
      anup(lat,lon)=co2_max+10
cnb      print *, "initialisation reussie, anup =",anup(lat,lon)
      test_veget=0

      do while (anup(lat,lon) .gt. co2_max)
cnb ---

c calculation of fraction dynamic variables

        fd=forshare_st-st(lat,lon)
        dd=desshare_st-sd(lat,lon)
        nld=nlshare_st-snlt(lat,lon)
        g4d=g4share_st-sg4(lat,lon)
        temp_st=st(lat,lon)
        temp_sg=sg(lat,lon)

c calculation of forest dynamics; exponential filtre
        dst=forshare_st-fd*exp(-1./t2t)-st(lat,lon)
        st(lat,lon)=st(lat,lon)+dst
        snlt(lat,lon)=nlshare_st-nld*exp(-1./t2t)

c desert dynamics; exponential filtre
        dsd=desshare_st-dd*exp(-1./t2g)-sd(lat,lon)
        tempor1=sd(lat,lon)+dsd+st(lat,lon)

c calculation of characteristic time of desert propagation
        if (tempor1.gt.0.9) then
             dstime=t2g*(1-tempor1)*10.+t2t*(tempor1-0.9)*10
```

```
            dsd=desshare_st-dd*exp(-1./dstime)-sd(lat,lon)
        endif

        sd(lat,lon)=sd(lat,lon)+dsd
        dsg=-dst-dsd

        sg(lat,lon)=1.-st(lat,lon)-sd(lat,lon)
        sg4(lat,lon)=g4share_st-g4d*exp(-1./t2g)

        if (sg(lat,lon).lt.0) sg(lat,lon)=0
        if (st(lat,lon).lt.0) st(lat,lon)=0
        if (sd(lat,lon).lt.0) sd(lat,lon)=0


c calculation of dynamics of storages

c calculation of changes of storages due to conservation law

c correction for trees

        tempor1=b4t(lat,lon)
        tempor2=b3t(lat,lon)
        tempor3=b4t14(lat,lon)
        tempor4=b3t14(lat,lon)
        tempor5=b4t13(lat,lon)
        tempor6=b3t13(lat,lon)


        if(st(lat,lon).gt.0) then
          if(dst.gt.0) then

            b4t(lat,lon)=(b4t(lat,lon)*temp_st
     >         +b4g(lat,lon)*dst)/st(lat,lon)
            b3t(lat,lon)=(b3t(lat,lon)*temp_st
     >         +b3g(lat,lon)*dst)/st(lat,lon)

            b4t14(lat,lon)=(b4t14(lat,lon)*temp_st
     >         +b4g14(lat,lon)*dst)/st(lat,lon)
            b3t14(lat,lon)=(b3t14(lat,lon)*temp_st
     >         +b3g14(lat,lon)*dst)/st(lat,lon)

            b4t13(lat,lon)=(b4t13(lat,lon)*temp_st
     >         +b4g13(lat,lon)*dst)/st(lat,lon)
            b3t13(lat,lon)=(b3t13(lat,lon)*temp_st
     >         +b3g13(lat,lon)*dst)/st(lat,lon)


          endif

          b2t(lat,lon)=b2t(lat,lon)*temp_st/st(lat,lon)
          b1t(lat,lon)=b1t(lat,lon)*temp_st/st(lat,lon)

          b2t14(lat,lon)=b2t14(lat,lon)*temp_st/st(lat,lon)
          b1t14(lat,lon)=b1t14(lat,lon)*temp_st/st(lat,lon)

          b2t13(lat,lon)=b2t13(lat,lon)*temp_st/st(lat,lon)
          b1t13(lat,lon)=b1t13(lat,lon)*temp_st/st(lat,lon)

        endif

c correction for grass
```

```
        if(sg(lat,lon).gt.0) then
                if(dst.gt.0) then

                 b4g(lat,lon)=b4g(lat,lon)*(temp_sg-dst)
     >           /sg(lat,lon)
                 b3g(lat,lon)=b3g(lat,lon)*(temp_sg-dst)
     >           /sg(lat,lon)

                 b4g14(lat,lon)=b4g14(lat,lon)*(temp_sg-dst)
     >           /sg(lat,lon)
                 b3g14(lat,lon)=b3g14(lat,lon)*(temp_sg-dst)
     >           /sg(lat,lon)

                 b4g13(lat,lon)=b4g13(lat,lon)*(temp_sg-dst)
     >           /sg(lat,lon)
                 b3g13(lat,lon)=b3g13(lat,lon)*(temp_sg-dst)
     >           /sg(lat,lon)


                else

           b4g(lat,lon)=(b4g(lat,lon)*temp_sg-tempor1*dst)
     >     /sg(lat,lon)
           b3g(lat,lon)=(b3g(lat,lon)*temp_sg-tempor2*dst)
     >     /sg(lat,lon)

           b4g14(lat,lon)=(b4g14(lat,lon)*temp_sg
     >     -tempor3*dst)/sg(lat,lon)
           b3g14(lat,lon)=(b3g14(lat,lon)*temp_sg
     >     -tempor4*dst)/sg(lat,lon)

           b4g13(lat,lon)=(b4g13(lat,lon)*temp_sg
     >     -tempor5*dst)/sg(lat,lon)
           b3g13(lat,lon)=(b3g13(lat,lon)*temp_sg
     >     -tempor6*dst)/sg(lat,lon)

                endif

         b2g(lat,lon)=b2g(lat,lon)*temp_sg/sg(lat,lon)
         b1g(lat,lon)=b1g(lat,lon)*temp_sg/sg(lat,lon)

         b2g14(lat,lon)=b2g14(lat,lon)*temp_sg/sg(lat,lon)
         b1g14(lat,lon)=b1g14(lat,lon)*temp_sg/sg(lat,lon)

         b2g13(lat,lon)=b2g13(lat,lon)*temp_sg/sg(lat,lon)
         b1g13(lat,lon)=b1g13(lat,lon)*temp_sg/sg(lat,lon)

        endif


c slow soil organic matter

ckc place holder for b4 terms

        b4t_hold=b4t(lat,lon)
        b4g_hold=b4g(lat,lon)
        b4t14_hold=b4t14(lat,lon)
        b4g14_hold=b4g14(lat,lon)
        b4t13_hold=b4t13(lat,lon)
        b4g13_hold=b4g13(lat,lon)
```

```
ckc for non permafrost affected fraction

        b4t(lat,lon)=w_frac(lat,lon)*(b4t_hold
     *               +k3t/t3t*b3t(lat,lon)-b4t_hold/t4t)
        b4g(lat,lon)=w_frac(lat,lon)*(b4g_hold+k4g/t2g*b2g(lat,lon)
     *               +k3g/t3g*b3g(lat,lon)-b4g_hold/t4g)

ckc for permafrost fraction of cell for b4 and c14 and c13

        b5t(lat,lon)=p_frac(lat,lon)*(b4t_hold+k3t/t3t*b3t(lat,lon)
     >               -b4t_hold/t5t)
        b5g(lat,lon)=p_frac(lat,lon)*(b4g_hold+k4g/t2g*b2g(lat,lon)
     >               +k3g/t3g*b3g(lat,lon)-b4g_hold/t5g)


        b4t14(lat,lon)=w_frac(lat,lon)*(b4t14_hold+k3t/t3t*
     >                 b3t14(lat,lon)-b4t14_hold/t4t)
        b4g14(lat,lon)=w_frac(lat,lon)*(b4g14_hold+k4g/t2g*
     >                 b2g14(lat,lon)+k3g/t3g*b3g14(lat,lon)-
     >                  b4g14_hold/t4g)


        b5t14(lat,lon)=p_frac(lat,lon)*(b4t14_hold+k3t/t3t*
     >                 b3t14(lat,lon)-b4t14_hold/t5t)
        b5g14(lat,lon)=p_frac(lat,lon)*(b4g14_hold+k4g/t2g*
     >                 b2g14(lat,lon)+k3g/t3g*b3g14(lat,lon)-
     >                  b4g14_hold/t5g)



        b4t13(lat,lon)=w_frac(lat,lon)*(b4t13_hold+k3t/t3t*
     >                 b3t13(lat,lon)-b4t13_hold/t4t)
        b4g13(lat,lon)=w_frac(lat,lon)*(b4g13_hold+k4g/t2g*
     >                 b2g13(lat,lon)+k3g/t3g*b3g13(lat,lon)-
     >                  b4g13_hold/t4g)



        b5t13(lat,lon)=p_frac(lat,lon)*(b4t13_hold+k3t/t3t*
     >                 b3t13(lat,lon)-b4t13_hold/t5t)
        b5g13(lat,lon)=p_frac(lat,lon)*(b4g13_hold+k4g/t2g*
     >                 b2g13(lat,lon)+k3g/t3g*b3g13(lat,lon)-
     >                  b4g13_hold/t5g)



ckc add permafrost carbon back in to b4 components

        b4t(lat,lon)=b4t(lat,lon)+b5t(lat,lon)
        b4g(lat,lon)=b4g(lat,lon)+b5g(lat,lon)

        b4t14(lat,lon)=b4t14(lat,lon)+b5t14(lat,lon)
        b4g14(lat,lon)=b4g14(lat,lon)+b5g14(lat,lon)

        b4t13(lat,lon)=b4t13(lat,lon)+b5t13(lat,lon)
        b4g13(lat,lon)=b4g13(lat,lon)+b5g13(lat,lon)


*        b4t14(lat,lon)=b4t14(lat,lon)+k3t/t3t
*     *                  *b3t14(lat,lon)
```

```
*     *                 -b4t14(lat,lon)/t4t
*        b4g14(lat,lon)=b4g14(lat,lon)+k4g/t2g
*     *                 *b2g14(lat,lon)+k3g/t3g*
*     *  b3g14(lat,lon)-b4g14(lat,lon)/t4g

*        b4t13(lat,lon)=b4t13(lat,lon)+k3t/t3t
*     *                 *b3t13(lat,lon)
*     *                 -b4t13(lat,lon)/t4t
*        b4g13(lat,lon)=b4g13(lat,lon)+k4g/t2g
*     *                 *b2g13(lat,lon)+k3g/t3g*
*     *  b3g13(lat,lon)-b4g13(lat,lon)/t4g


c   fast soil organic matter
ckc place holder for fast soil terms
        b3t_hold=b3t(lat,lon)
        b3g_hold=b3g(lat,lon)
        b3t14_hold=b3t14(lat,lon)
        b3g14_hold=b3g14(lat,lon)
        b3t13_hold=b3t13(lat,lon)
        b3g13_hold=b3g13(lat,lon)

ckc non-permafrost fraction of cell, b3, permafrost affected b6

        b3t(lat,lon)=w_frac(lat,lon)*(b3t_hold+b1t(lat,lon)
     >          /t1t*k0t+k2t/t2t*b2t(lat,lon)-b3t_hold/t3t)
        b3g(lat,lon)=w_frac(lat,lon)*(b3g_hold+b1g(lat,lon)
     >          /t1g*k0g+k2g/t2g*b2g(lat,lon)-b3g_hold/t3g)


        b6t(lat,lon)=p_frac(lat,lon)*(b3t_hold+b1t(lat,lon)
     >          /t1t*k0t+k2t/t2t*b2t(lat,lon)-b3t_hold/t6t)
        b6g(lat,lon)=p_frac(lat,lon)*(b3g_hold+b1g(lat,lon)
     >          /t1g*k0g+k2g/t2g*b2g(lat,lon)-b3g_hold/t6g)


        b3t14(lat,lon)=w_frac(lat,lon)*(b3t14_hold+b1t14(lat,lon)
     >        /t1t*k0t+k2t/t2t*b2t14(lat,lon)-b3t14_hold/t3t)
        b3g14(lat,lon)=w_frac(lat,lon)*(b3g14_hold+b1g14(lat,lon)
     >        /t1g*k0g+k2g/t2g*b2g14(lat,lon)-b3g14_hold/t3g)
        b6t14(lat,lon)=p_frac(lat,lon)*(b3t14_hold+b1t14(lat,lon)
     >        /t1t*k0t+k2t/t2t*b2t14(lat,lon)-b3t14_hold/t6t)
        b6g14(lat,lon)=p_frac(lat,lon)*(b3g14_hold+b1g14(lat,lon)
     >        /t1g*k0g+k2g/t2g*b2g14(lat,lon)-b3g14_hold/t6g)

        b3t13(lat,lon)=w_frac(lat,lon)*(b3t13_hold+b1t13(lat,lon)
     >        /t1t*k0t+k2t/t2t*b2t13(lat,lon)-b3t13_hold/t3t)
        b3g13(lat,lon)=w_frac(lat,lon)*(b3g13_hold+b1g13(lat,lon)
     >        /t1g*k0g+k2g/t2g*b2g13(lat,lon)-b3g13_hold/t3g)
        b6t13(lat,lon)=p_frac(lat,lon)*(b3t13_hold+b1t13(lat,lon)
     >        /t1t*k0t+k2t/t2t*b2t13(lat,lon)-b3t13_hold/t6t)
        b6g13(lat,lon)=p_frac(lat,lon)*(b3g13_hold+b1g13(lat,lon)
     >        /t1g*k0g+k2g/t2g*b2g13(lat,lon)-b3g13_hold/t6g)

ckc sum b3 components back together

        b3t(lat,lon)=b3t(lat,lon)+b6t(lat,lon)
        b3g(lat,lon)=b3g(lat,lon)+b6g(lat,lon)
        b3t14(lat,lon)=b3t14(lat,lon)+b6t14(lat,lon)
        b3g14(lat,lon)=b3g14(lat,lon)+b6g14(lat,lon)
        b3t13(lat,lon)=b3t13(lat,lon)+b6t13(lat,lon)
```

```
        b3g13(lat,lon)=b3g13(lat,lon)+b6g13(lat,lon)

ckc Soil respiration

      rsoil_g(lat,lon)=w_frac(lat,lon)*((-b3g_hold/t3g)+(-b4g_hold/t4g))
     >                +p_frac(lat,lon)*((-b3g_hold/t6g)+(-b4g_hold/t5g))
      rsoil_t(lat,lon)=w_frac(lat,lon)*((-b3g_hold/t3t)+(-b4g_hold/t4t))
     >                +p_frac(lat,lon)*((-b3g_hold/t6t)+(-b4g_hold/t5t))

ckc Veg respiration

       rveg_g(lat,lon)=-b2g(lat,lon)/t2g
       rveg_t(lat,lon)=(-b1t(lat,lon)/t1t)+(-b2t(lat,lon)/t2t)


c leaves biomass


        b1t(lat,lon)=b1t(lat,lon)+k1t*npp-b1t(lat,lon)/t1t
        b1g(lat,lon)=k1g*npp*t1g


        b1t14(lat,lon)=b1t14(lat,lon)+k1t*npp*c14atm
     *                 -b1t14(lat,lon)/t1t

             b1g14(lat,lon)=k1g*npp*c14atm*t1g


        b1t13(lat,lon)=b1t13(lat,lon)+k1t*npp*c13atm*
     *                 c13frac-b1t13(lat,lon)/t1t


        b1g13(lat,lon)=k1g*npp*c13atm*(c13frac*(1-sg4(lat,lon))+
     *                 c13frac4*sg4(lat,lon))*t1g

c   stems and roots biomass

        b2t(lat,lon)=b2t(lat,lon)+(1-k1t)*npp-b2t(lat,lon)/t2t
        b2g(lat,lon)=b2g(lat,lon)+(1-k1g)*npp-b2g(lat,lon)/t2g


        b2t14(lat,lon)=b2t14(lat,lon)+(1-k1t)*npp*c14atm
     *                 -b2t14(lat,lon)/t2t

        b2g14(lat,lon)=b2g14(lat,lon)+(1-k1g)*npp*c14atm
     *                 -b2g14(lat,lon)/t2g

        b2t13(lat,lon)=b2t13(lat,lon)+(1-k1t)*npp*c13atm
     *                 *c13frac-b2t13(lat,lon)/t2t

        b2g13(lat,lon)=b2g13(lat,lon)+(1-k1g)*npp*c13atm
     *                 *(c13frac*(1-sg4(lat,lon))+
     *                 c13frac4*sg4(lat,lon))-b2g13(lat,lon)/t2g

c   c14 annual decay
        b1t14(lat,lon)=b1t14(lat,lon)*(1-c14tdec)
        b2t14(lat,lon)=b2t14(lat,lon)*(1-c14tdec)
        b3t14(lat,lon)=b3t14(lat,lon)*(1-c14tdec)
        b4t14(lat,lon)=b4t14(lat,lon)*(1-c14tdec)
        b1g14(lat,lon)=b1g14(lat,lon)*(1-c14tdec)
        b2g14(lat,lon)=b2g14(lat,lon)*(1-c14tdec)
```

```
        b3g14(lat,lon)=b3g14(lat,lon)*(1-c14tdec)
        b4g14(lat,lon)=b4g14(lat,lon)*(1-c14tdec)

        call CLIMPAR

cnb --- Ze Test sur le carbone pris par la veget
cnb --- S il est trop important on divise npp par 2
cnb --- et on recalcul tous les reservoirs

ckc  should also be for KLSR is 2 for seds restart?
cam: This test is not done if KLSR=1:
        IF (KLSR.EQ.1) anup(lat,lon)=co2_max
cnb        do while (anup(lat,lon) .gt. co2_max)
          if (anup(lat,lon) .gt. co2_max) then
cnb            print *,"lat,lon",lat,lon,"anup",anup(lat,lon)
            npp=npp/2
            test_veget=1
cnb          print*,'passe dans test veget'
          endif

      enddo !fin du test


        return
        end
```