```
cool = ems*beta*x_j = h\nu*beta*A_ij*x_j

But A_ij*x_j is really (n_j/nmol)Aij so

cool = (n_j/nmol)h\nu*Aij*beta_ij

F = E18 (Nmol*c/Dv) * cool/nu  so that

           F = E18 (1/Delta\nu) h\nu*Aij*beta_ij*N_j
```

```
!
!     SCALE RATES WITH WEIGHT FACTORS AND DEFINE THE REST OF CONSTANTS
!        A and C are scaled by WE; note -
!          n(i) A(i,j) = nmol x(i) (we(i) A(i,j))
!          n(i) C(i,j) = nmol x(i) (we(i) C(i,j))
!
!     The code uses line center optical depth
      AUX    = (NMOL/V)*CL**3/EITPI/ROOTPI
      DO I = 2,N
         DO J = 1,I-1
            A(I,J)     =  A(I,J)*WE(I)
            C(I,J)     =  C(I,J)*WE(I)*NH2
            TAUX(I,J)  =  AUX*A(I,J)/FREQ(I,J)**3
            EMS(I,J)   =  HPL*FREQ(I,J)*A(I,J)
         END DO
      END DO

      do i = 1, n
          coolev(i) = 0.0
          do j = 1, i
            cool(i,j) = 0.0
            if(a(i,j) .ne. 0.0) then
              cool(i,j) = ems(i,j)*esc(i,j)*x(i)
              if(tau(i,j) .lt. 0.0) then
!             this is an inverted transition:
                  nmaser = nmaser + 1
                  imaser(nmaser) = i
                  jmaser(nmaser) = j
!                 forget the emission from maser transitions
                  cool(i,j) = 0.
              end if
              coolev(i) = coolev(i) + cool(i,j)
              tcool = tcool + cool(i,j)
            end if
          end do
      end do

!     Molecular column per bandwith in kms:
      mcol = nmol*r*1.d5/V

!     for final summary printing
      aux = 1.D18*mcol*CL

!  Printing elements for every selected transition:
!     1 - Molecular column per velocity bandwidth
!     2 - Excitation temperature
!     3 - tau
!     4 - flux density in Jy, obtained from COOL(i,j) which is in erg/s/mol
!     5 - velocity-integrated line brightness temperature (at angle mu) in K*km/s
!     6 - line intensity (at angle mu for slab)
!     7 - line brightness temperature against CMB, Tbr
!     8 - RJ equivalent of Tbr
!     When dust absorption is on, next element is
!     9 - fractional contribution of dust to tau when there's dust absorption
!
```

```
      do k = 1, n_tr
         m(2) = itr(k)
         m(1) = jtr(k)
         nu    = freq(m(2),m(1))
         Tl    = TIJ(m(2),m(1))
         Tex   = Tl/DLOG(POP(m(1))/POP(m(2)))
         depth = tau(m(2),m(1))
!        Integrate (1 - exp(-tau_v)) over the line
         call simpson(100,1,100,freq_axis,1.0 -
         dexp(-(depth/mu_output)*exp(-freq_axis**2)),integral)
         call Tbr4Tx(Tl,Tex,depth,Tbr,TRJ)

         fin_tr(k,1,nprint) = mcol
         fin_tr(k,2,nprint) = Tex
         fin_tr(k,3,nprint) = depth
         fin_tr(k,4,nprint) = aux*cool(m(2),m(1))/freq(m(2),m(1))
         fin_tr(k,5,nprint) = Tex*(vt/1.d5)*integral
         fin_tr(k,6,nprint) = BB(nu,Tex)*(1. - dexp(-depth/mu_output))
         fin_tr(k,7,nprint) = Tbr
         fin_tr(k,8,nprint) = TRJ
```