

REBOIII Manual

Travis Kemper
University of Florida

August 21, 2015

1 Formalism

The Reactive empirical bond order potential was first developed by Brenner et. al. in 1990 [?]. It was developed to model CVD deposition of diamond. It was later modified into the REBOII potential [?].

1.1 Energy function

$$E_b = \sum_i \sum_{j>i} [V_R(r_{ij}) - \bar{b}_{ij} V_A(r_{ij})] \quad (1)$$

The pair terms are:

$$V^R(r_{ij}) = f^C(r_{ij}) (1 + Q/r_{ij}) A e^{-\alpha r_{ij}} \quad (2)$$

$$V^A(r_{ij}) = f^C(r_{ij}) \sum_{n=1,3} B e^{-\beta r_{ij}} \quad (3)$$

Where $f^C(r_{ij})$ is the cut off function

$$f^c(r_{ij}) = \begin{cases} 1 & r_{ij} < D^{min} \\ [1 + \cos[(r_{ij} - D^{min})/(D^{max} - D^{min})]] & D^{min} < r_{ij} < D^{max} \\ 0 & r_{ij} > D^{max} \end{cases} \quad (4)$$

The bond order term is:

$$\bar{b}_{ij}(r_{ij}, r_{ik}, r_{jl}, r_{kl}) = \frac{1}{2} [b_{ij}^{\sigma-\pi} + b_{ji}^{\sigma-\pi}] + b_{ij}^{\pi} \quad (5)$$

where the σ and π bonding is accounted for by averaging the bond order terms of the i-j and j-i interactions. This term contains the three body terms dependant on bond angle θ_{ijk} and the coordination function P_{ij} .

$$b_{ij}^{\sigma-\pi} = \left[1 + \sum_{k \neq i,j} f_{ik}^C(r_{ik}) G(\cos(\theta_{ijk})) e^{\lambda_{ijk}} + P_{ij}(N_i^C, N_i^H) \right]^{-\frac{1}{2}} \quad (6)$$

For the hydrocarbon centered angular function is a bicubic spline G_C and G_H . For the oxygen based interactions [?]

$$G(\cos(\theta_{ijk})) = a_0 + a_1 [a_2 - \cos(\theta)]^2 \quad (7)$$

where the cosine of θ can be calculated using the law of cosines:

$$\cos(\theta_{ijk}) = \frac{r_{ij}^2 + r_{ik}^2 - r_{kj}^2}{2r_{ij}r_{ik}} \quad (8)$$

and the function e^λ is used to remove spurious wells in the H angle function, otherwise $\lambda = 0$. Referring to the code it seems e^λ is complex that it seems.

$$e_{ijk}^\lambda = e^{\lambda_{ijk}(BL_{ik}-BL_{ij})} e^{\lambda_{ijk}(r_{ij}-r_{jk})} \quad (9)$$

where $BL = 0$ except for $ijk = HCH$ and $ijk = HHC$ when $BL_{HH} = 0.7415886997$ and $BL_{CH} = 1.09$. The coordination function is a bicubic spline dependant on the number of neighbors of each element type. Where the total number of elemental neighbors is the sum of the cut off function f^c

$$N_i^{elm} = \sum_{k \neq i,j}^{atomsofelm} f_{ik}^c(r_{ik}) \quad (10)$$

π bonding is taken into account with four body term:

$$b_{ij}^\pi = \Pi^{RC} + b_{ij}^{DH} \quad (11)$$

The radical term Π_{ij}^{RC} describes the radical nature of carbon bond as well as conjugation effects.

$$\Pi_{ij}^{RC} = F_{ij}(N_i^t, N_j^t, N_{ij}^{conj}) \quad (12)$$

where F_{ij} is a tricubic spline which depends on the total numbers of C and it conjugation N_{ij}^{conj} .

$$N_i^t = \sum_{elm}^{neighborsofi} N_i^{elm} \quad (13)$$

$$N_{ij}^{conj} = 1 + \left[\sum_{k \neq i,j}^{carbon} f_{ik}^C(r_{ik}) F(X_{ik}) \right]^2 + \left[\sum_{l \neq i,j}^{carbon} f_{jl}^C(r_{jl}) F(X_{jl}) \right]^2 \quad (14)$$

$$F(x_{ik}) = \begin{cases} 1 & x_{ik} = 2 \\ [1 + \cos(\pi(x_{ik} - 2))] / 2 & 2 < x_{ik} < 3 \\ 0 & x_{ik} > 3 \end{cases} \quad (15)$$

$$x_{ik} = N_k^t - f_{ik}^c(r_{ik}) \quad (16)$$

the dihedral term b_{ij}^{DH} is a four body torsional interaction

$$b_{ij}^{DH} = T_{ij}(N_i^t, N_j^t, N_{ij}^{conj}) \left[\sum_{(k \neq i,j)} \sum_{(l \neq i,j)} (1 - \cos^2(\Theta_{ijkl})) f_{ik}^c(r_{ik}) f_{jl}^c(r_{jl}) \right] \quad (17)$$

where T_{ij} is a tricubic spline, and

$$\cos(\Theta_{ijkl}) = e_{ijk} e_{ijl} \quad (18)$$

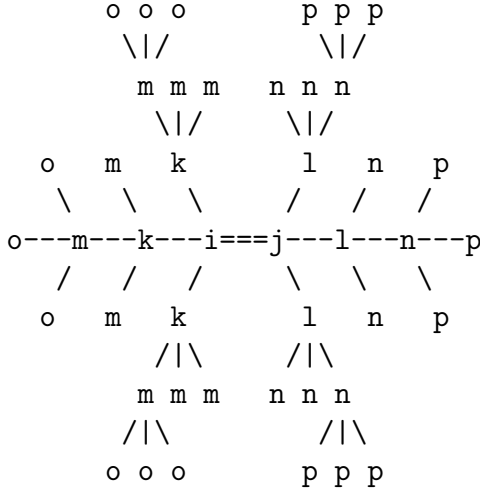
where

$$e_{ijk} = \hat{r}_{ji} \times \hat{r}_{ik} \quad (19)$$

$$e_{ijk} = \frac{r_{ji} \times r_{ik}}{|r_{ji}| |r_{ik}| \sin \theta_{jik}} \quad (20)$$

1.2 Derivatives

The energy function above is dependant on firts, second or third neighbors if screaning is implemented.



Therefore, the force $\frac{dE}{dr}$ on these atoms must be calculated.

$$E_b(r_{ij}, r_{ik}, r_{jl}) = \sum_i \sum_{j>i} [V_R(r_{ij}) - \bar{b}_{ij} V_A(r_{ij})] \quad (21)$$

$$\frac{dE_b}{dr_{ij}} = \frac{dV_R(r_{ij})}{dr_{ij}} - \bar{b}_{ij} \frac{dV_A(r_{ij})}{dr_{ij}} - \frac{d\bar{b}_{ij}}{dr_{ij}} V_A(r_{ij}) \quad (22)$$

$$\frac{dE_b}{dr_{ik}} = \frac{dV_R(r_{ij})}{dr_{ij}} - \bar{b}_{ij} \frac{dV_A(r_{ij})}{dr_{ij}} - \frac{d\bar{b}_{ij}}{dr_{ij}} V_A(r_{ij}) \quad (23)$$

$$\frac{dV^R(r_{ij})}{dr_{ij}} = \frac{df^c(r_{ij})}{dr_{ij}} \left(1 + \frac{Q}{r_{ij}}\right) A e^{-\alpha r_{ij}} + f^c(r_{ij}) \left[\frac{Q}{r_{ij}^2} A e^{-\alpha r_{ij}} - \alpha r_{ij} \left(1 + \frac{Q}{r_{ij}}\right) A e^{-\alpha r_{ij}} \right] \quad (24)$$

$$\frac{dV^A(r_{ij})}{dr_{ij}} = \frac{df^c(r_{ij})}{dr_{ij}} \sum_{n=1,3} B e^{-\beta_n r_{ij}} - f^c(r_{ij}) \sum_{n=1,3} \beta_n B e^{-\beta_n r_{ij}} \quad (25)$$

$$\frac{df^c(r_{ij})}{dr_{ij}} = \begin{cases} 0 & r_{ij} < D^{min} \\ -\frac{1}{2(D^{max}-D^{min})} \sin[(r_{ij} - D^{min})/(D^{max} - D^{min})] & D^{min} < r_{ij} < D^{max} \\ 0 & r_{ij} > D^{max} \end{cases} \quad (26)$$

the multibody bond order term is dependant on r_{ij} , r_{ik} , r_{jl} and r_{kl} . Taking the derivative of b_{ij} with respect to r_{ij}

$$\frac{d\bar{b}_{ij}}{dr_{ij}} = \frac{1}{2} \left[\frac{db_{ij}^{\sigma-\pi}}{dr_{ij}} + \frac{db_{ji}^{\sigma-\pi}}{dr_{ij}} \right] + \frac{db_{ij}^{\pi}}{dr_{ij}} \quad (27)$$

where

$$\frac{db_{ij}^\pi}{dr_{ij}} = \frac{d\Pi^{RC}}{dr_{ij}} + \frac{db_{ij}^{DH}}{dr_{ij}} \quad (28)$$

if

$$b_{ij}^{\sigma-\pi} = [d_{ij}]^{-\frac{1}{2}} \quad (29)$$

where

$$d_{ij} = 1 + \sum_{k \neq i} f_{ik}^C(r_{ik})G(\cos(\theta_{ijk}))e + P_{ij}(N_i^C, N_i^H) \quad (30)$$

then

$$\frac{db_{ij}}{dr_{ij}} = \frac{db_{ij}}{dd_{ij}} \frac{dd_{ij}}{dr_{ij}} \quad (31)$$

$$\frac{db_{ij}}{dd_{ij}} = -\frac{b_{ij}}{2d_{ij}} \quad (32)$$

$$\frac{dd_{ij}}{dr_{ij}} = \sum_{k \neq i,j} f_{ik}^C(r_{ik})e^{\lambda_{ijk}} \frac{dG(\cos(\theta_{ijk}))}{dr_{ij}} + f_{ik}^C(r_{ik})G(\cos(\theta_{ijk})) \frac{de^{\lambda_{ijk}}}{dr_{ij}} + \frac{dP_{ij}(N_i^C, N_i^H)}{dr_{ij}} \quad (33)$$

$$\frac{dG(\cos(\theta_{ijk}))}{dr_{ij}} = \frac{dG(\cos(\theta_{ijk}))}{dcos(\theta_{ijk})} \frac{dcos(\theta_{ijk})}{dr_{ij}} \quad (34)$$

if $G(\cos(\theta_{ijk}))$ is for oxygen or sulfur

$$\frac{dG(\cos(\theta_{ijk}))}{dcos(\theta_{ijk})} = 2a_1 [\cos(\theta) - a_2] \quad (35)$$

$\cos(\theta_{ijk})$ is calculated with the law of cosines ??

$$\frac{dcos(\theta_{ijk})}{dr_{ij}} = \frac{r_{ij}^2 - r_{ik}^2 + r_{kj}^2}{2r_{ij}r_{ik}r_{ij}} \quad (36)$$

$$\frac{de^{\lambda_{ijk}}}{dr_{ij}} = \lambda_{ijk}e^{\lambda_{ijk}} \quad (37)$$

$$\frac{dP_{ij}(N_i^C, N_i^H)}{dr_{ij}} = \sum_{elm} \frac{P_{ij}}{dN_{elm}} \frac{dN_{elm}}{dr_{ij}} \quad (38)$$

$$\frac{dN^{elm}}{dr_{ij}} = 0??? \quad (39)$$

$$\frac{db_{ij}^{DH}}{dr_{ij}} = \frac{dT_{ij}}{dr_{ij}} DH_{sum} + T_{ij} \frac{dDH_{sum}}{dr_{ij}} \quad (40)$$

where

$$DH_{sum} = \left[\sum_{(k \neq i, j)} \sum_{(l \neq i, j)} (1 - \cos^2(\Theta_{ijkl})) f_{ik}^c(r_{ik}) f_{jl}^c(r_{jl}) \right] \quad (41)$$

$$\frac{dDH_{sum}}{dr_{ij}} = \left[\sum_{(k \neq i, j)} \sum_{(l \neq i, j)} \frac{d(1 - \cos^2(\Theta_{ijkl}))}{dr_{ij}} f_{ik}^c(r_{ik}) f_{jl}^c(r_{jl}) \right] \quad (42)$$

$$\frac{d(1 - \cos^2(\Theta_{ijkl}))}{dr_{ij}} = 2 \frac{d\cos(\Theta_{ijkl})}{dr_{ij}} \quad (43)$$

$$\frac{d}{dr_{ij}} \left[\frac{(r_{ji} \times r_{ik})(r_{ij} \times r_{jl})}{|r_{ik}| |r_{jl}| |r_{ij}|^2 \sin\theta_{ik} \sin\theta_{jl}} \right] = \frac{d}{dr_{ij}} \left[\frac{rcross}{rmag} \right] \quad (44)$$

$$\frac{d}{dr_{ij}} \left[\frac{rcross}{rmag} \right] = \frac{\frac{drcross}{r_{ij}} rmag + rcross \frac{drmag}{dr_{ij}}}{rmag^2} \quad (45)$$

$$\frac{drmag}{dr_{ij}} = 2r_{ik}r_{jl}r_{ij}\sin\theta_{ik}\sin\theta_{jl} + r_{ik}r_{jl}r_{ij}^2\sin\theta_{ik}\frac{d\sin\theta_{jl}}{r_{ij}} + r_{ik}r_{jl}r_{ij}^2\sin\theta_{jl}\frac{d\sin\theta_{ik}}{r_{ij}} \quad (46)$$

$$\frac{d\sin\theta_{ik}}{dr_{ij}} = \frac{d(1 - \cos^2\theta_{ik})^{\frac{1}{2}}}{dr_{ij}} = -\frac{\cos\theta_{ik}}{\sin\theta_{ik}} \frac{d\cos\theta_{ik}}{dr_{ij}} \quad (47)$$

$$\frac{d\sin\theta_{jl}}{dr_{ij}} = \frac{d(1 - \cos^2\theta_{jl})^{\frac{1}{2}}}{dr_{ij}} = -\frac{\cos\theta_{jl}}{\sin\theta_{jl}} \frac{d\cos\theta_{jl}}{dr_{ij}} \quad (48)$$

Taking the derivative of b_{ij} with respect to r_{ik}

$$\frac{d\cos(\theta_{ijk})}{dr_{ik}} = \frac{-(r_{ij}^2 - r_{ik}^2) + r_{kj}^2}{2r_{ij}r_{ik}r_{ik}} \quad (49)$$

$$\frac{de^{\lambda_{ijk}}}{dr_{ij}} = -\lambda_{ijk}e^{\lambda_{ijk}} \quad (50)$$

$$\frac{dN^{elm}}{dr_{ij}} = \sum_{k \neq i,j}^{atomsofelm} \frac{df_{ik}^c(r_{ik})}{r_{ik}} \quad (51)$$

$$\frac{drmag}{dr_{ik}} = |r_{jl}||r_{ij}|^2 \sin\theta_{jl} \sin\theta_{ik} + |r_{ik}||r_{jl}||r_{ij}|^2 \sin\theta_{jl} \frac{d\sin\theta_{ik}}{dr_{ik}} \quad (52)$$

$$\frac{d\sin\theta_{ik}}{dr_{ik}} = -\frac{\cos\theta_{ik}}{\sin\theta_{ik}} \frac{d\cos\theta_{ik}}{dr_{ik}} \quad (53)$$

$$\frac{drcross}{dr_{ik}} = \left[\frac{d}{dr_{ij}} (\vec{r}_{ji} \times \vec{r}_{ik}) \right] \cdot (\vec{r}_{ij} \times \vec{r}_{jl}) \quad (54)$$

$$\frac{drcross}{dr_{ik}} = \left[\frac{d}{dr_{ij}} \left(\begin{vmatrix} r_{ji}^y & r_{ji}^z \\ r_{ik}^y & r_{ik}^z \end{vmatrix} \hat{x} - \begin{vmatrix} r_{ji}^x & r_{ji}^z \\ r_{ik}^x & r_{ik}^z \end{vmatrix} \hat{y} + \begin{vmatrix} r_{ji}^x & r_{ji}^y \\ r_{ik}^x & r_{ik}^y \end{vmatrix} \hat{z} \right) \right] \cdot \left(\begin{vmatrix} r_{ij}^y & r_{ij}^z \\ r_{jl}^y & r_{jl}^z \end{vmatrix} \hat{x} - \begin{vmatrix} r_{ij}^x & r_{ij}^z \\ r_{jl}^x & r_{jl}^z \end{vmatrix} \hat{y} + \begin{vmatrix} r_{ij}^x & r_{ij}^y \\ r_{jl}^x & r_{jl}^y \end{vmatrix} \hat{z} \right)$$

$$\frac{drcross}{dr_{ik}} = \left((r_{ji}^y - r_{ji}^z) \begin{vmatrix} r_{ij}^y & r_{ij}^z \\ r_{jl}^y & r_{jl}^z \end{vmatrix} \hat{x} + (r_{ji}^z - r_{ji}^x) \begin{vmatrix} r_{ij}^x & r_{ij}^z \\ r_{jl}^x & r_{jl}^z \end{vmatrix} \hat{y} + (r_{ji}^x - r_{ji}^y) \begin{vmatrix} r_{ij}^x & r_{ij}^y \\ r_{jl}^x & r_{jl}^y \end{vmatrix} \hat{z} \right) \quad (55)$$

Taking the derivative of b_{ij} with respect to r_{jl}

$$\frac{drmag}{dr_{jl}} = |r_{ik}||r_{ij}|^2 \sin\theta_{jl} \sin\theta_{ik} + |r_{ik}||r_{jl}||r_{ij}|^2 \sin\theta_{ik} \frac{d\sin\theta_{jl}}{dr_{jl}} \quad (56)$$

$$\frac{d\sin\theta_{jl}}{dr_{jl}} = -\frac{\cos\theta_{jl}}{\sin\theta_{jl}} \frac{d\cos\theta_{jl}}{dr_{jl}} \quad (57)$$

Taking the derivative of b_{ij} with respect to r_{il}

$$\frac{drmag}{dr_{il}} = |r_{ik}||r_{jl}||r_{ij}|^2 \sin\theta_{ik} \frac{d\sin\theta_{jl}}{dr_{il}} \quad (58)$$

$$\frac{d\sin\theta_{jl}}{dr_{il}} = -\frac{\cos\theta_{jl}}{\sin\theta_{il}} \frac{d\cos\theta_{jl}}{dr_{il}} \quad (59)$$

Taking the derivative of b_{ij} with respect to r_{kj}

$$\frac{d\cos(\theta_{ijk})}{dr_{kj}} = \frac{2r_{kj}}{2r_{ij}r_{ik}} \quad (60)$$

$$\frac{drmag}{dr_{kj}} = |r_{jl}||r_{ij}|^2 \sin\theta_{jl} \sin\theta_{ik} + |r_{ik}||r_{jl}||r_{ij}|^2 \sin\theta_{jl} \frac{d\sin\theta_{ik}}{dr_{ik}} \quad (61)$$

$$\frac{d\sin\theta_{ik}}{dr_{ik}} = -\frac{\cos\theta_{ik}}{\sin\theta_{ik}} \frac{d\cos\theta_{ik}}{dr_{ik}} \quad (62)$$

2 Parameters

2.1 sub.setpairs.f

Set pair parameters and angular function as well as potential cut off region.

1. $A = DD(I, J)$
2. $\alpha = DXL(I, J)$
3. $Q = ED(I, J)$
4. $B = AD(I, J)$
5. $\beta = AXL(I, J)$
6. $B_2 = BD(I, J)$
7. $\beta_2 = BXL(I, J)$
8. $B_3 = CD(I, J)$
9. $\beta_3 = CXL(I, J)$

where B_2, B_3 and β_2, β_3 are set to 1 and zero respectively for all elements except carbon. $RB1(I, J)$ and $RB2$ are the inner and outer points of the REBO cut off region used in f^C .

1. $RLIST(I, J) = (RB2(I, J) + RLL)^2$
2. $RMAX = RB1^2$

Where RLL is the neighbor itemizecut off.

2.2 sub.setsplines.f

1. $XHO = P_O(N^H + N^C, N^O)$ knot values
2. $CLMOX = P_{HC,O}$ coefficients
3. $XHS = P_S(N^H + N^C, N^S)$ knot values
4. $CLMSX = P_{HC,S}$ coefficients
5. $SPGC = G_C$
6. $SPGH = G_H$
7. $XH(CC = 1/CH = 2, N^H, N^C), = P_{CC/CH}(N^H, N^C)$ knot values
8. $CLM() = P_{HC}$ coefficients

9. $IN2$ = bicubic spline exponents
10. $IN3$ = tricubic spline exponents
11. $CLMN()$ = F_{HC} coefficients
12. $TLMN()$ = F_{ij} torsional coefficients

2.3 mtable.f

Loop over all interactions

```
do ki=1,ntypes
  do kj = ki,ntype
```

$ntab$ the number of data points taken to define the pair potential between 0 – R_{max}

```
ddtab(ki,kj) = rb2(ki,kj)/float(ntab - 2)
ddtab(kj,ki) = ddtab(ki,kj)
```

$ddtab(ki, kj) = dx$ the distance between two points when defining the potential table

Loop over $\sum_{r=dx}^{(r_{cut}^{outer})}$

2.3.1 Calculate the cut off function and its derivative

$f^c(r)$

```
IF(RC.LT.RB2(KI,KJ)) THEN
  DTEMP=PID(KI,KJ)*(RC-RB1(KI,KJ))
  FC=(1.0d0+COS(DTEMP))/2.0d0
  DFC=-PID(KI,KJ)/2.0d0*SIN(DTEMP)
ENDIF
```

C

```
IF(RC.LE.RB1(KI,KJ)) THEN
  FC=1.0d0
  DFC=0.0d0
ENDIF
```

If $r_{min}^C < r < r_{max}^C$

$$PID = frac{\pi}{2} (D^{max} - D^{min}) \quad (63)$$

$$DTEMP = PID * (r - D^{min}) \quad (64)$$

$$f^c(r) = \frac{1 + \cos(DTEMP)}{2}$$

which leads to the cut off function ?? and its derivative.

$$\frac{df^c}{dr} = -\frac{1}{2(D^{MAX} - D^{min})} \sin\left(\frac{r - D^{min}}{D^{Max} - D^{min}}\right) \quad (65)$$

Where if $f^c(D^{min}) = 1$ and $f^c(D^{max}) = 0$
 f^c and DFC are stored in `tabfc` and `tabdfc`

```
tabfc(ki,kj,i) = fc
tabfc(kj,ki,i) = tabfc(ki,kj,i)
tabdfc(ki,kj,i) = dfc
tabdfc(kj,ki,i) = tabdfc(ki,kj,i)
```

2.3.2 Calculate pair terms

Attractive pair terms ??

```
VA=AD(KI,KJ)*EXP(-AXL(KI,KJ)*RC)
DVA=-AXL(KI,KJ)*VA
VB=BD(KI,KJ)*EXP(-BXL(KI,KJ)*RC)
DVB=-BXL(KI,KJ)*VB
VC=CD(KI,KJ)*EXP(-CXL(KI,KJ)*RC)
DVC=-CXL(KI,KJ)*VC
```

$$\frac{dV^A}{dr} = \frac{df^c}{dr} \left[\sum_{n=1}^3 B_n e^{-\beta_n r} \right] + f^c \left[\sum_{n=1}^3 -\beta_n B_n e^{-\beta_n r} \right] \quad (66)$$

Where

$$\sum_{n=1}^3 B_n e^{-\beta_n r} = VA + VB + VC \quad (67)$$

and

$$\sum_{n=1}^3 -\beta_n B_n e^{-\beta_n r} = DVA + DVB + DVC \quad (68)$$

Values at RC are stored in `atable` and `datable`

```
VV=(VA+VB+VC)/2.0D0
DVV=(DVA+DVB+DVC)/2.0D0
atable(ki,kj,i) = FC*VV
atable(kj,ki,i) = atable(ki,kj,i)
datable(ki,kj,i) = (FC*DVV+DFC*VV)/RC
datable(kj,ki,i) = datable(ki,kj,i)
```

not sure why there is a 1/2 in there ???

$$atable(KI, KJ, I) = \frac{1}{2} \sum_{n=1}^3 B_n e^{-\beta_n r}$$

$$datable(KI, KJ, I) = -\frac{1}{2r} \frac{dV^A}{dr}$$

2.3.3 Repulsive pair terms

```
FF1=DD(KI,KJ)*EXP(-DXL(KI,KJ)*RC)
DF1=-DXL(KI,KJ)*FF1
FF2=(1.0D0+ED(KI,KJ)/RC)
DF2=-ED(KI,KJ)/RSQ
```

$$\begin{aligned} FF1 &= Ae^{-\alpha r} \\ DF1 &= -\alpha FF1 \\ FF2 &= \left(1 + \frac{Q}{r}\right) \\ DF2 &= -\frac{Q}{r} \end{aligned}$$

```
VV=FF1*FF2
DVM=(DF1*FF2 + FF1*DF2)
```

$$\begin{aligned} VV &= \left(1 + \frac{Q}{r}\right) Ae^{-\alpha r} \\ DVM &= \left(-\frac{Q}{r} Ae^{-\alpha r}\right) + (-\alpha) \left(1 + \frac{Q}{r}\right) Ae^{-\alpha r} \end{aligned}$$

store repulsive values in *rtable* and *drtable*

```
rtable(ki,kj,i) = vv*fc
rtable(kj,ki,i) = rtable(ki,kj,i)
drtable(ki,kj,i) = -(FC*DVM+DFC*VV)/RC
drtable(kj,ki,i) = drtable(ki,kj,i)
```

2.4 caguts.f

Check to see if neighbor list should be updated

```
IF(LCHK.EQ.1) THEN
```

where *LCHK* is set in the predictor corrector. If set to 1 then update neighbor list.

```
      K=0
      DO 302 I=1,NP
        NABORS(I)=K+1
        DO 299 L=1,3
          RI(L)=R0(I,L)
299      CONTINUE
        KI=KTYPE(I)
C
c cuts out all but C,H,Si, and Ge
C
          if(ki.ge.5) go to 302
C
        DO 301 J=1,NP
C
          IF(I.EQ.J) GO TO 301
C
          KJ=KTYPE(J)
C
c cuts out all but C,H,Si, and Ge
C
          if(kj.ge.5) go to 301
          RLIS=RLIST(KI,KJ)
C
          RSQ=0.0D0
          DO 298 L=1,3
            RR(L)=RI(L)-R0(J,L)
            RR(L)=RR(L) -
&              CUBE(L)*ANINT(RR(L)/CUBE(L))
            RSQ=RSQ+RR(L)*RR(L)
            IF(RSQ.GT.RLIS) GO TO 301
298      CONTINUE
C
405      CONTINUE
        K=K+1
        LIST(K)=J
        IVCT2B(K)=I
        JVCT2B(K)=J
C
```

```

301             CONTINUE
302 CONTINUE
C
      NABORS(NP+1)=K+1
      KEND=K
      if(kend.gt.nlmax) then
        write(*,*) 'kend exceeds nlmax'
        write(*,*) 'kend,nlmax = ',kend,nlmax
        write(*,*) 'increase nlmax and recompile'
        include 'close.inc'
        stop
      endif
C      write(*,*) 'kend= ',kend
      ENDIF

```

Loop over all neighbors and grab atom number and atom ID.

```

DO 320 K=1,KEND
  I=IVCT2B(K)
  J=JVCT2B(K)
  KI=KTYPE(I)
  KJ=KTYPE(J)

```

LCHECK determine if part of REBO parameter set
Calculate inter neighbor distance and store r^2 in *COR*

```

  RSQ=0.0D0
  DO L=1,3
    RR(L)=R0(I,L)-R0(J,L)
    RR(L)=RR(L) - CUBE(L)*ANINT(RR(L)/CUBE(L))
    RSQ=RSQ+RR(L)*RR(L)
    COR(K,L)=RR(L)
  ENDDO

```

If distance is greater than cut off leave *LCHECK* = 0 and skip interaction.

```

IF(RSQ.GT.RMAX(KI,KJ)) GOTO 320

```

Otherwise set REBO atoms 1-10 to *LCHECK* = 1 and other atoms to *LCHECK* = 2
which was originally tight binding

```

  if((kj.le.10).and.(ki.le.10)) LCHECK(K)=1
  if((kj.ge.10).and.(ki.ge.10)) LCHECK(K)=2

```

Calculate table value of r and store r in *RCOR*

```

RC=SQRT(RSQ)
rt = rc/ddtab(ki,kj)
it = min(int(rt) + 1,ntab-1)
RCOR(K)=RC

```

$$rt = \frac{r_{ij}}{dr_{TAB}}$$

$it = \text{minimum}(\text{INT}(rt) \text{ or } \text{total number of elements in lookup table})$

Interpolate table values of pair potential by finding the table value and adding the difference of the current value and the next value times the difference between r and its truncated integer value it used for the look up table

```

      WW(K)=TABFC(ki,kj,it)
&      +(TABFC(ki,kj,it+1)-TABFC(ki,kj,it))*(rt-it+1)
      DWW(K)=TABDFC(ki,kj,it)
&      +(TABDFC(ki,kj,it+1)-TABDFC(ki,kj,it))*(rt-it+1)
      EXX1(K) = atable(ki,kj,it)
&      +(atable(ki,kj,it+1)-atable(ki,kj,it))*(rt-it+1)
      DEXX1(K) = datable(ki,kj,it) +
&      (datable(ki,kj,it+1)-datable(ki,kj,it))*(rt - it +1)
      IF(I.GE.J) GO TO 320
      vv = rtable(ki,kj,it)
&      +(rtable(ki,kj,it+1)-rtable(ki,kj,it))*(rt-it+1)
      rp = drtable(ki,kj,it)
&      +(drtable(ki,kj,it+1)-drtable(ki,kj,it))*(rt-it+1)

```

The cut off function interpolation is written explicitly the other values are calculated in the same manner.

$$\begin{aligned}
WW &= f^C(it) + (f^C(it + dr_{TAB}) - f^C(it))(r - it) = f^C(r_{TAB}) \\
DWW &= \frac{df^C(r_{TAB})}{dr} \\
EXX1 &= V^A(r_{TAB}) \\
DEXX1 &= \frac{V^A(r_{TAB})}{dr} \\
vv &= V^R(r_{TAB}) \\
rp &= \frac{V^R(r_{TAB})}{dr}
\end{aligned}$$

The values for f^C and V^A are stored to be later multiplied by b_{ij} , while V^R is added to the total energy.

```

tote = tote + vv
eatom(i) = eatom(i) + vv/2.0d0
eatom(j) = eatom(j) + vv/2.0d0

```

$$E_{total} = \sum_{pair=1}^K E_{pair}$$

$$E_{atom} = \frac{1}{2} \sum E_{pair}$$

Store repulsive force for pair

```

DO 318 L=1,3
      RPP(K,L)=RP*RR(L)
318   CONTINUE

```

$$F_{pair}(k, x/y/z) = \frac{dV^R}{dr} r$$

Add repulsive force to total force. Loop over all pairs and if *lcheck* $\neq 0$ ($r \notin D^{MAX}$) and $I < J$

```

DO 321 K=1,KEND
  if(lcheck(k).eq.0) go to 321
  I=IVCT2B(K)
  J=JVCT2B(K)
  IF(I.GE.J) GO TO 321
  DO 322 L=1,3
    RNP(I,L)=RNP(I,L) + RPP(K,L)
    RNP(J,L)=RNP(J,L) - RPP(K,L)
322   CONTINUE
321   CONTINUE

```

$$RNP(I, :) = \sum F_{pair}$$

2.5 pibond.f

Loop over all atoms and determine connectivity

```

      DO 500 I=1,NP
        JBEGIN=NABORS(I)
        JEND=NABORS(I+1)-1
        DO NN = 1,NTYPES
          XHC(I,NN)=1.0d0
        ENDDO
        IF(JBEGIN.GT.JEND) GO TO 500
        DO 490 J=JBEGIN,JEND
          IF(LCHECK(J).ne.1) GO TO 490
          JN=LIST(J)
          XHC(I,KTYPE(JN))=XHC(I,KTYPE(JN))+WW(J)
490      CONTINUE
500    CONTINUE

```

If *LCHECK* is equal to 1 (REBO pair under D^{MAX} cut off distance) and atom number *JN* from *LIST*. Sum the cut off function of *J* ?? and store in *XHC* array.

$$XHC(I, KJ) = N_J^{KJ}$$

Loop over all atoms and its neighbors

```

C*****
      DO 40 I=1,NP
        JBEGIN=NABORS(I)
        JEND=NABORS(I+1)-1
        IF(JBEGIN.GT.JEND) GO TO 40
        KI=KTYPE(I)
        DO 30 J=JBEGIN,JEND
          IF(LCHECK(J).ne.1) GO TO 30
          JN=LIST(J)
          IF(I.GE.JN) GO TO 30
30      CONTINUE
C*****

```

Get \hat{r} values from *COR* array

```

      DO 401 MM=1,3
        CJ(MM)=COR(J,MM)
401    CONTINUE

```

Set $SIJ = ||r||$ [Var:??] and $RSQIJ = r \cdot r$ and $KIKIJ = KI + KJ$. Intial three body variables (NK,XSIJ,SSUMK,CONK) and local arrays.

```

NK=0
XSIJ=0.0d0
SSUMK=0.0d0
CONK=0.0D0
XNI(:)=XHC(I,:)
XNI(KJ) = XNI(KJ) - WW(J)
QI =-1.0d0* FLOAT(NTYPES)
DO NN = 1,NTYPES
    QI = QI + XNI(NN)
ENDDO
SDALIK=0.0D0

```

Loop over all other neighbors other than J and keep track of how many there are with ??.

```

DO 20 K=JBEGIN,JEND
    IF(K.EQ.J) GO TO 20

```

Calculate the angle between $K - I - J$ using the law of cosines.

$$\cos(\theta) = \frac{a^2 + b^2 - c^2}{2ab} \quad (69)$$

```

DO 402 MM=1,3
    XK(NK,MM)=COR(K,MM)-CJ(MM)
    RSQ2=RSQ2+XK(NK,MM)*XK(NK,MM)
402 CONTINUE
SS=2.0d0*SIJ*S3
RR=RSQIJ-RSQ3
COSTH=(RSQIJ+RSQ3-RSQ2)/SS

```

$$\begin{aligned}
 XK(NK, D) &= r_{KI} \hat{D} - r_{JI} \hat{D} \\
 RSQ2 &= (\hat{r}_K - \hat{r}_J)^2 = r_{KJ}^2 = c^2 \\
 SS &= 2r_{KI}r_{JI} = 2ab \\
 RR &= r_{JI}^2 - r_{KI}^2 = a^2 - b^2 \\
 COSTH &= \frac{r_{JI}^2 + r_{KI}^2 - r_{KJ}^2}{2r_{KI}r_{JI}}
 \end{aligned}$$

Round in case $||\cos|| > 1$

```

IF(COSTH.GT.1.0D0) COSTH=1.0D0
IF(COSTH.LT.-1.0D0) COSTH=-1.0D0

```

Store $\cos(\theta_{kij})$ in *COSK* and calculate *sin*

```
SINK(NK)=SQRT(1.0D0-COSTH*COSTH)
IF(ACOS(COSTH).GT.PI) SINK(NK)=-SINK(NK)
```

$$\sin = \sqrt{1 - \cos^2}$$

2.5.1 carbon anglur function

If KI is carbon Calculate spline index *IG* and spline values *GANGLE* and *DGDTHET*

```
IG=IGC(INT(-COSTH*12.0D0)+13)
IF(KI.EQ.1) THEN
  IF(IG.NE.4) THEN
    GANGLE=SPGC(1,IG)+SPGC(2,IG)*COSTH
    DGDTHET=SPGC(2,IG)
    DO 45 JJ=3,6
      GANGLE=GANGL+SPGC(JJ,IG)
      &                                *(COSTH**(JJ-1))
      DGDTHET=DGDTHET+SPGC(JJ,IG)*
      &                                (JJ-1)*(COSTH**(JJ-2))
45    CONTINUE
  ELSE
    ALI=0.0D0
    DALI=0.0D0
    IF(QI.LT.XQM) THEN
C####
C      QQE=QI-2.0D0
C      QQM=QQE/(QI-XQM)
C      ALI=ATT*EXP(0.05D0*QQE*QQM)
C      DALI=ALI*0.05D0*QQM*(2.0D0-QQM)
C####

      ALI=1.0D0
      IF(QI.GT.ATT) THEN
        DTEMP=PQ*(QI-ATT)
        ALI=(1.0D0+COS(DTEMP))/2.0D0
        DALI=-PQ/2.0D0*SIN(DTEMP)
      ENDIF
C####

      ENDIF
      GANGLE=SPGC(1,IG)+SPGC(2,IG)*COSTH
      DGDTHET=SPGC(2,IG)
      IG1=IG+1
      GANGLE1=SPGC(1,IG1)+SPGC(2,IG1)*COSTH
```

```

DGDTHET1=SPGC(2,IG1)
DO 545 JJ=3,6
    GANGLE=GANGL1+SPGC(JJ,IG)
    &                                *(COSTH**(JJ-1))
    DGDTHET=DGDTHET+SPGC(JJ,IG)*
    &                                (JJ-1)*(COSTH**(JJ-2))
C
    GANGLE1=GANGL1+SPGC(JJ,IG1)
    &                                *(COSTH**(JJ-1))
    DGDTHET1=DGDTHET1+SPGC(JJ,IG1)*
    &                                (JJ-1)*(COSTH**(JJ-2))
545
    CONTINUE
    DALDIK=DALI*(GANGL1-GANGL)
    GANGLE=GANGL+ALI*(GANGL1-GANGL)
    DGDTHET=DGDTHET+ALI*(DGDTHET1-DGDTHET)
ENDIF

```

$$IG = \begin{array}{ll} 4 & 1 < \cos(\theta) < -\frac{1}{4} & 0 < \theta < 102.5 \\ 3 & -\frac{1}{4} < \cos(\theta) < -\frac{5}{12} & ? \\ 2 & -\frac{1}{2} < \cos(\theta) < -\frac{7}{12} & ? \\ 1 & -\frac{2}{3} < \cos(\theta) < -1 & ? \end{array} \quad (70)$$

$$\begin{aligned}
IG = 1, 2, 3 \text{ } GANGLE &= a + b\cos(\theta) + c\cos(\theta)^2 + d\cos(\theta)^3 + e\cos(\theta)^4 + f\cos(\theta)^5 \\
DGDTHET &= b + 2c\cos(\theta) + 3d\cos(\theta)^2 + 4e\cos(\theta)^3 + 5f\cos(\theta)^4
\end{aligned}$$

2.5.2 Oxygen angular function

```

ELSEIF (KI.eq.isulfur) THEN
    GANGLE = a_s0 + a_s1 * (a_s2 - costh)**2
    DGDTHET = 2.d0 * a_s1 * (costh - a_s2)

```

$$\begin{aligned}
G(\theta) &= a_s^0 + a_s^1(a_s^0 - \cos(\theta))^2 \\
\frac{dG(\theta)}{d\cos(\theta)} &= 2a_s^1(\cos(\theta) - a_s^0)
\end{aligned}$$

2.5.3 Hydrogen angular function

```

ELSE
    IG=IGH(INT(-COSTH*12.0D0)+13)

```

```

GANGLE=SPGH(1,IG)+SPGH(2,IG)*COSTH
DGDTHET=SPGH(2,IG)

DO 46 JJ=3,6
  GANGLE=GANGLE+SPGH(JJ,IG)*(COSTH**(JJ-1))
  DGDTHET=DGDTHET+SPGH(JJ,IG)*
    (JJ-1)*(COSTH**(JJ-2))
&
46 CONTINUE

```

2.5.4 Conjugate Evaluation

From Equ. ?? where X_{ik} is the degree of saturation

```

FC=WW(K)
DFC=DWW(K)
CFUNI(NK)=0.0d0
DCFUNI(NK)=0.0d0
IF(KK.EQ.icarb.and.KI.NE.isulfur.and.
&  KJ.NE.isulfur ) THEN
  FNTF = FLOAT(NTYPES)
  XX = - FC -1.0d0*FNTF
  DO NN = 1,NTYPES
    XX=XX + XHC(KN,NN)
  ENDDO
  IF(XX.LT.3.0d0) THEN
    IF(XX.LE.2.0d0) THEN
      CFUNI(NK)=1.0d0
    ELSE
      PX=PI*(XX- 2.0d0 )
      CFUNI(NK)=(1.0+COS(PX))/2.0d0
      DCFUNI(NK)=-FC*SIN(PX)*PI/2.0d0
    ENDIF
  ENDIF
ENDIF
CONK=CONK+FC*CFUNI(NK)

```

$$\begin{aligned}
FC &= f_{ik}^C \\
DFC &= \frac{df_{ik}^C}{d??} \\
XX(KJ) = X_{ik} &= \sum_{k \neq i, j}^{allatomtypes} f_{ik}^{KJ} \\
PX &= \pi(X_{ik} - 2) \\
CFUNI &= F(X_{ik}) \\
DCFUNI &= \frac{dF(X_{ik})}{dX_{ik}} \\
CONK &= \sum_{k \neq i, j}^{allatomtypes} f_{ik}^C * F(X_{ik})
\end{aligned}$$

Evaluate λ in b_{ij} if XDB not equal zero.

$$\begin{aligned}
&\text{IF}(XDB(KI, KJ, KK) .NE. 0.0D0) \text{ THEN} \\
&\quad \text{EXX} = \text{REG}(KI, KJ, KK) \\
&\& \quad * \text{EXP}(XDB(KI, KJ, KK) * (SIJ - S3)) \\
&\text{ELSE} \\
&\quad \text{EXX} = 1.0 \\
&\text{ENDIF} \\
\\
&\begin{aligned}
XDB(2, 2, 2) &= 4.0D0 \\
XDB(2, 1, 2) &= 4.0D0 \\
XDB(2, 2, 1) &= 4.0D0 \\
XDB(2, 1, 1) &= 4.0D0 \\
XDB(1, 2, 1) &= 0.0D0 \\
XDB(1, 2, 2) &= 0.0D0
\end{aligned}
\end{aligned} \tag{71}$$

$$XDB = \lambda(KI, KJ, KK)$$

2.5.5 Calculate dihedral angles

I think ??

$$\begin{aligned}
DCTDJK &= -2.0d0/SS \\
DCTDIJ &= (RR + RSQ2) / (SS * RSQIJ) \\
DCTDIK &= (-RR + RSQ2) / (SS * RSQ3) \\
DCTJK(NK) &= DCTDJK \\
DCTIJ(NK) &= DCTDIJ \\
DCTIK(NK) &= DCTDIK \\
GS &= GANGLE * EXX
\end{aligned}$$

```

SSUMK=SSUMK + FC*GS
XTEMP=FC*EXX*DGDTHET
GFX=GS*FC*XDB(KI,KJ,KK)
XSIJ=XSIJ + XTEMP*DCTDIJ+GFX/SIJ
XSIK(NK)=(GS*DFC-GFX)/S3+XTEMP*DCTDIK
SDALIK=SDALIK+EXX*FC*DALDIK
XSJK(NK) = XTEMP*DCTDJK

```

$DCTDJK =$

Repeat for JI side of the bond

2.6 Corrdination function

Initialize spline values

```

EXNIJ=0.0d0
DEXNI(:)=0.0d0
EXNI2J = 0.0d0
DEXNI2(:) = 0.0d0

```

Set the thresh hold value of N^S to use sulfur spline.

```

pno_min = 1.01d0
pno_max = 1.21d0
delt_po = pno_max - pno_min
pie0 = dacos(-1.d0)
pij_min =-0.39d0

```

Round the number of nieghbors to find knots for spline evaluation. And remember all these values have a factor of +1 to allow for fortran77 indexing, ie $N \neq 0$. Which leads to a lot of -1 's

```

NH=INT(XNI(ihyd)+1.0D-12)
NC=INT(XNI(icarb)+1.0D-12)
NSX=INT(XNI(isulfur)+1.0D-12)
NTCH = NH + NC

```

IF C-C bond use P_{CC} and the number of sulfur atoms is less then the threshold value and set the derivative with respect to sulfur the same as the derivative with respect to carbon.

```

      IF(KI.EQ.icarb.and.KJ.EQ.icarb) THEN
        IF( (ABS(FLOAT(NH)-XNI(ihyd)).GT.1.0d-8).OR.
&          (ABS( FLOAT(NC+NSX)-XNI(icarb)-XNI(isulfur))
&          .GT.1.0d-8)) THEN
          CALL BCUINT(KI,KJ,XNI(2),XNI(1)+XNI(3)-1.0d0
&                  ,NH,NC+NSX - 1,
&                  EXNIJ,DEXNI(2),DEXNI(1) )
        ELSE
          EXNIJ=XH(KJ,NH,NC)
          DEXNI(2)=XH1(KJ,NH,NC)
          DEXNI(1)=XH2(KJ,NH,NC)
        ENDIF
        DEXNI(isulfur)=DEXNI(icarb)

```

Use if statement to determine if the value of N_j^{KJ} is close enough to an integer to evaluate the spline at the knots. Add N^S value to N^C value as a small correction.

$$N^H = \sum_{k \neq i,j}^H f_{IK}^c(r_{IK}) N^C = \sum_{k \neq i,j}^C f_{IK}^c(r_{IK}) + \sum_{k \neq i,j}^S f_{IK}^c(r_{IK}) P_{CC} = (N^H, N^C,)$$

If C-H bond calculate spline values from P_{CH} if the number of sulfur neighbors is less than the threshold

```

      ELSEIF (KI.EQ.icarb.and.KJ.EQ.ihyd) THEN
        IF(XNI(isulfur).LE.pno_min ) THEN
          IF ( (ABS(FLOAT(NH) - XNI(ihyd)).gt.1.0d-8) .OR.
&            (ABS(FLOAT(NC) - XNI(icarb)).gt.1.0d-8) ) THEN
            CALL BCUINT(KI,KJ,XNI(2),XNI(1),NH,NC,
&                    EXNIJ,DEXNI(2),DEXNI(1) )
          ELSE
            EXNIJ=XH(KJ,NH,NC)
            DEXNI(2)=XH1(KJ,NH,NC)
            DEXNI(1)=XH2(KJ,NH,NC)
          ENDIF

```

If the number of sulfur neighbors is greater than the threshold then use switching function. Interpolate values from each spline P^{CH} and P^{CHS} .

```

      ELSEIF ( XNI(isulfur).GE.pno_max) THEN
        IF((ABS(FLOAT(NTCH)-XNI(icarb)-XNI(ihyd)).GT.1.0d-8)
&        .or.(ABS(FLOAT(NSX) - XNI(isulfur)).GT.1.0d-8)) THEN
          CALL SULBCUINT (KI,KJ,XNI(icarb)+XNI(ihyd)-1.0d0,
&                    XNI(isulfur),NTCH-1,NSX,EXNIJ
&                    ,DEXNI(icarb),DEXNI(isulfur) )

```



```

        DEXNI(ihyd) = DEXNI(icarb)
        IF ( EXNIJ.LT.pij_min) THEN
            EXNIJ = pij_min
        ENDIF
    ELSE
        EXNIJ = XHS(KI,KJ,NTCH-1,NSX)
        DEXNI(icarb) = XHS1(KI,KJ,NTCH-1,NSX)
        DEXNI(isulfur) = XHS2(KI,KJ,NTCH-1,NSX)
        DEXNI(ihyd) = DEXNI(icarb)
    ENDIF
ELSE !switching region
    IF ( (ABS(FLOAT(NH) - XNI(ihyd)).GT.1.0d-8).OR.
&      (ABS(FLOAT(NC) - XNI(icarb)).GT.1.0d-8)) THEN
        CALL BCUINT(KI,KJ,XNI(2),XNI(1),NH,NC,
&              EXNIJ,DEXNI(2),DEXNI(1) )
    ELSE
        EXNIJ=XH(KJ,NH,NC)
        DEXNI(2)=XH1(KJ,NH,NC)
        DEXNI(1)=XH2(KJ,NH,NC)
    ENDIF
    IF((ABS(FLOAT(NTCH)-XNI(icarb)-XNI(ihyd)).GT.1.0d-8)
& .or.(ABS(FLOAT(NSX) - XNI(isulfur)).GT.1.0d-8)) THEN
        CALL SULBCUINT (KI,KJ,XNI(icarb)+XNI(ihyd)-1.0d0,
&              XNI(isulfur),NTCH-1,NSX,EXNI2J
&              ,DEXNI2(icarb),DEXNI2(isulfur) )
        DEXNI2(ihyd) = DEXNI2(icarb)
        IF ( EXNIJ.LT.pij_min) THEN
            EXNI2J = pij_min
        ENDIF
    ELSE
        EXNI2J = XHS(KI,KJ,NTCH-1,NSX)
        DEXNI2(icarb) = XHS1(KI,KJ,NTCH-1,NSX)
        DEXNI2(isulfur) = XHS2(KI,KJ,NTCH-1,NSX)
        DEXNI2(ihyd) = DEXNI2(icarb)
    ENDIF

```

Then interpolate values

```

    fijmid = 0.5d0*(1.d0+cos(pie0*(XNI(isulfur)-pno_min)
&      /delt_po))
    dfijmid = - pie0 * 0.5d0 *sin(pie0*(XNI(isulfur)-pno_min)
&      /delt_po)/delt_po
    pijtmp = EXNIJ*fijmid + EXNI2J*(1-fijmid)
    DEXNI(icarb)=DEXNI(icarb)*fijmid+DEXNI2(icarb)*(1-fijmid)
    DEXNI(ihyd)=DEXNI(ihyd)*fijmid+DEXNI2(ihyd)*(1-fijmid)

```

```

      DEXNI(isulfur) = DEXNI(isulfur)*fijmid +
&                      DEXNI2(isulfur)*(1-fijmid) +
&                      (EXNIJ-EXNI2J)*dfijmid
      EXNIJ = pijtmp

```

fijmid = ??

If H- not to oxygen. set everything to zero

```

      ELSEIF (KI.EQ.ihyd.and.KJ.NE.isulfur) THEN
        EXNIJ = 0.0d0
        DEXNI(:) = 0.0d0

```

Otherwise H-O bond

```

      ELSE
        IF ( (ABS(FLOAT(NTCH)-XNI(icarb)-XNI(ihyd)).GT.1.0d-8)
&          .or.(ABS(FLOAT(NSX)-XNI(isulfur)).GT.1.0d-8)) THEN
          CALL SULBCUINT(KI,KJ,XNI(icarb)+XNI(ihyd)-1.0d0,
&                      XNI(isulfur),NTCH-1,NSX,EXNIJ
&                      ,DEXNI(icarb),DEXNI(isulfur) )

          DEXNI(ihyd) = DEXNI(icarb)
          IF ( EXNIJ.LT.pij_min) THEN
            EXNIJ = pij_min
          ENDIF
        ELSE
          EXNIJ = XHS(KI,KJ,NTCH-1,NSX)
          DEXNI(icarb) = XHS1(KI,KJ,NTCH-1,NSX)
          DEXNI(isulfur) = XHS2(KI,KJ,NTCH-1,NSX)
          DEXNI(ihyd) = DEXNI(icarb)
        ENDIF
      ENDIF

```

Calculate b_{ij} , and retrieve V^A from *EXX1*.

```

      DIJ=(1.0d0+EXNIJ+SSUMK)
      BIJ=DIJ**(-0.50D0)
      DJI=(1.0d0+EXNJI+SSUML)
      BJI=DJI**(-0.50D0)
      DBDZI=-0.50D0*BIJ/DIJ
      DBDZJ=-0.50D0*BJI/DJI
      VATT=EXX1(J)

```

$$\begin{aligned}
DIJ &= 1 + P_{IJ} + G(\theta) \\
BIJ &= \frac{1}{(1 + P_{IJ} + G(\theta))^{\frac{1}{2}}} \\
DJI &= 1 + P_{JI} + G(\theta) \\
BJI &= \frac{1}{(1 + P_{JI} + G(\theta))^{\frac{1}{2}}} \\
DBDZI &= \frac{1}{2} \frac{1}{(1 + P_{IJ} + G(\theta))^{\frac{3}{2}}} \\
DBDZJ &= \frac{1}{2} \frac{1}{(1 + P_{JI} + G(\theta))^{\frac{3}{2}}} \\
VATT &= V^A(J)
\end{aligned}$$

Calculate values for conjugate term.

```

DRADI=0.0d0
DRADJ=0.0d0
DRDC=0.0d0
CONJUG = 1.0D0 + (CONK**2) + (CONL**2)
XNT1 = 1.0d0 - 1.0d0*FLOAT(NTYPES)
XNT2 = 1.0d0 - 1.0d0*FLOAT(NTYPES)
DO NN = 1,NTYPES
  XNT1 = XNT1 + XNI(NN)
  XNT2 = XNT2 + XNJ(NN)
ENDDO

```

2.6.1 Energy

```
TOTE=TOTE-BTOT*VATT

eatom(i) = eatom(i) - btot*vatt/2.0d0
eatom(jn)= eatom(jn) - btot*vatt/2.0d0

VDBDI=VATT*DBDZI
VDBDJ=VATT*DBDZJ
VDRDC=VATT*DRDC
VDRDI=VATT*DRADI
VDRDJ=VATT*DRADJ
```

$TOTE =$

2.6.2 Forces

Start suming forces

```
RP= VDBDI*XSIIJ + VDBDJ*XSJII + BTOT*DEXX1(J)
DO 407 MM=1,3
    REP=RP*CJ(MM)
    RNP(I,MM)=RNP(I,MM)+REP
    RNP(JN,MM)=RNP(JN,MM)-REP
407 CONTINUE
```

$RP = VDBDI = VATT * DBDZIVDBDJ = VATT * DBDZJVDRDC = VATT * DRDCVDRDI =$

3 Predictor corrector

4 Thermostats

4.1 Langevin Equation

```
DO 30 II=1,NTA
    I=NLIST(II)
```

```

        BM=BET*XMASS(KTYPE(I))
        SM=SQRT(XMASS(KTYPE(I)))
        DO 29 J=1,3
            rrzp = RNP(I,J)
            RNP(I,J)=RNP(I,J)-BM*R1(I,J)-SM*GL(II+(J-1)*NTA)
29      CONTINUE
30    CONTINUE

```

5 Terminology

1. atom number - sequential number of atom with in list of all atoms in the simulation
2. atom ID - internal reference number 1-N total parameterized elements

6 Variables

1. I - atom number [I]
2. KI - atom ID [L]
3. J - nieghbor atom number [L]
4. KJ - niegghbor atom ID [L]
5. KTYPE(I) = KI - atom number to atom ID [G]
6. SIJ = $||r_{ij}||$ [L]
7. NK = number of K nieghbors [L]
8. D = spacial dimension x,y,z [L]