BITS AND PIECES, SAMPLE ROUTINES

Note: this document is updated during the course!

GENERATING FCC LATTICE

```
      SUBROUTINE FCC(N,NC,BOXL,RX,RY,RZ)

      IMPLICIT NONE

!  SETS UP THE FCC LATTICE FOR N ATOMS
!  THE SIMULATION BOX IS A UNIT CUBE CENTRED AT THE ORIGIN.
!  N SHOULD BE AN INTEGER OF THE FORM ( 4 * ( NC ** 3 ) ),
!  WHERE NC IS THE NUMBER OF FCC UNIT CELLS IN EACH DIRECTION.
!  PRINCIPAL VARIABLES:
!  INTEGER N                    NUMBER OF PARTICLES
!  REAL*8    RX(N),RY(N),RZ(N)    PARTICLE POSITIONS

      INTEGER     N, NC
      REAL*8        RX(N), RY(N), RZ(N)
      REAL*8        CELL, CELL2, BOXL
      INTEGER     I, IX, IY, IZ, IREF, M

!*********************************************************************

C     ** CALCULATE THE SIDE OF THE UNIT CELL **
      CELL  = 1.0 / REAL ( NC )
      CELL2 = 0.5 * CELL

C     ** BUILD THE UNIT CELL **

C     ** SUBLATTICE A **
      RX(1) =  0.0
      RY(1) =  0.0
      RZ(1) =  0.0

C     ** SUBLATTICE B **
      RX(2) =  CELL2
      RY(2) =  CELL2
      RZ(2) =  0.0

C     ** SUBLATTICE C **
      RX(3) =  0.0
      RY(3) =  CELL2
      RZ(3) =  CELL2

C     ** SUBLATTICE D **
      RX(4) =  CELL2
      RY(4) =  0.0
      RZ(4) =  CELL2
```

```
C     ** CONSTRUCT THE LATTICE FROM THE UNIT CELL **
         M = 0
         DO IZ = 1, NC
            DO IY = 1, NC
               DO IX = 1, NC
                  DO IREF = 1, 4
                     RX(IREF+M) = RX(IREF) + CELL * REAL ( IX - 1 )
                     RY(IREF+M) = RY(IREF) + CELL * REAL ( IY - 1 )
                     RZ(IREF+M) = RZ(IREF) + CELL * REAL ( IZ - 1 )
                  ENDDO
                  M = M + 4
               ENDDO
            ENDDO
         ENDDO

C  SHIFT CENTRE OF BOX TO THE ORIGIN AND CONVERT TO SIMULATION UNITS
(IN TERMS OF BOXL)
         RX(:) = (RX(:) - 0.5) * BOXL
         RY(:) = (RY(:) - 0.5) * BOXL
         RZ(:) = (RZ(:) - 0.5) * BOXL

         RETURN
         END
```

A NOTE ABOUT VISUALIZATION WITH VMD: THE FORMAT OF XYZ FILE

VMD-program needs the input coordinates in Angstroms, so we can use e.g. the Lennard-Jones sigma parameter for Argon: σ = 3.40 A.
One record of the coordinates in the xyz format has N+2 lines (N = number of atoms):

N
(comment line, can be empty)
Ar x1 y1 z1
Ar x2 y2 z2
…
Ar xN yN zN

You can have more than 1 of these records in the same file, VMD reads the full file and can show the data as an animation.

INITIAL VELOCITIES FROM GAUSSIAN DISTRIBUTION

```
      SUBROUTINE VELDIST(N,TEMPER,MASS,VX,VY,VZ)

! RETURNS A NORMAL DISTRIBUTION FOR THE INITIAL VELOCITIES
! CORRESPONDING TO TEMPERATURE TEMPER
! REFERENCE: ALLEN & TILDESLEY, COMPUTER SIMULATION OF LIQUIDS P. 347

      IMPLICIT NONE

      REAL*8 BOLTZ  , PI
!     PARAMETER (BOLTZ=8.617E-5, PI=4.*ATAN(1.))
      PARAMETER (BOLTZ=1., PI=3.141592654)

      INTEGER N,I,J
      REAL*8 TEMPER,MASS,R1,R2,VXS,VYS,VZS,SCALE,VS
      REAL*8 VX(N),VY(N),VZ(N)

      VXS=0.
      VYS=0.
      VZS=0.

! USE F90 INTRINSIC ROUTINE RANDOM_NUMBER(x), WHICH RETURNS A UNIFORM
! DISTRIBUTION  WITH 0<x<1

      DO I=1,N
            CALL RANDOM_NUMBER(R1)
            CALL RANDOM_NUMBER(R2)
            VX(I)=SQRT(-2.*LOG(R1))*COS(2.*PI*R2)
            CALL RANDOM_NUMBER(R1)
            CALL RANDOM_NUMBER(R2)
            VY(I)=SQRT(-2.*LOG(R1))*COS(2.*PI*R2)
            CALL RANDOM_NUMBER(R1)
            CALL RANDOM_NUMBER(R2)
            VZ(I)=SQRT(-2.*LOG(R1))*COS(2.*PI*R2)
            VXS=VXS+VX(I)
            VYS=VYS+VY(I)
            VZS=VZS+VZ(I)
      ENDDO

! REMOVE CENTER OF MASS MOTION

      VXS=VXS/REAL(N)
      VYS=VYS/REAL(N)
      VZS=VZS/REAL(N)
      VS=0.
      DO I=1,N
            VX(I)=VX(I)-VXS
            VY(I)=VY(I)-VYS
            VZ(I)=VZ(I)-VZS
            VS=VS+VX(I)**2+VY(I)**2+VZ(I)**2
      ENDDO
```

```
! SCALE TO DESIRED TEMPERATURE

      SCALE=SQRT(3.*REAL(N)*BOLTZ*TEMPER/MASS/VS)

      DO I=1,N
            VX(I)=SCALE*VX(I)
            VY(I)=SCALE*VY(I)
            VZ(I)=SCALE*VZ(I)
      ENDDO

      RETURN
      END
```

THE LENNARD - JONES POTENTIAL AND SIMULATIONS UNITS

Potential: only two parameters, σ and ε. σ sets the length scale, ε the energy scale

$$V(r) = 4 \varepsilon [(\sigma / r)^{12} - (\sigma / r)^{6}]$$

Bottom of the well: $r_0 = (2)^{1/6} \sigma$
Depth of the well: $V_0 = \varepsilon$

The Lennard-Jones parameters are natural parameters for the simulation units in the MD program. In addition one needs a third unit, it can be either for time or mass. Selecting one will define the other. Since we are going to do simulations with only one type of particle ("Lennard-Jones atom") I suggest that everybody uses the mass of this particle as the unit of mass – meaning mass = 1. This defines the unit of time as

```
CC   --> UNIT OF TIME = SIGMA * SQRT( 1. / EPSILON )
```

Unit of pressure is

```
CC   --> UNIT OF PRESSURE = EPSILON / SIGMA ** 3
```

A sample routine to calculate LJ potential and forces, applying cutoff of the potential and periodic boundary conditions (PBC). Note: it is useful to make the application of PBC optional by introducing an input parameter (`PERIODIC`) that controls whether these effects take action or not.

```
      SUBROUTINE LJONES( BOX, PERIODIC, RCUT, V, VC, W,
     :                   N, RX, RY, RZ, FX, FY, FZ )

      IMPLICIT NONE
!*************************************************************
! LENNARD-JONES FORCE ROUTINE IN REDUCED UNITS
!
! THE POTENTIAL IS V(R) = 4*((1/R)**12-(1/R)**6)
! WE INCLUDE SPHERICAL CUTOFF AND MINIMUM IMAGING IN CUBIC BOX.
! TWO POTENTIAL ENERGIES ARE RETURNED.
! V IS CALCULATED USING THE UNSHIFTED POTENTIAL. WHEN LONG-
! RANGE TAIL CORRECTIONS ARE ADDED, THIS MAY BE USED TO
! CALCULATE THERMODYNAMIC INTERNAL ENERGY ETC.
! VC IS CALCULATED USING THE SHIFTED POTENTIAL WITH NO
! DISCONTINUITY AT THE CUTOFF.  THIS MAY BE USED TO CHECK THE
! CONSERVATION LAWS.
!
! PRINCIPAL VARIABLES:
!
! INTEGER N                 NUMBER OF MOLECULES
! REAL*8    RX(N),RY(N),RZ(N) MOLECULAR POSITIONS
! REAL*8    FX(N),FY(N),FZ(N) MOLECULAR FORCES
! REAL*8    BOX               SIMULATION BOX LENGTH
! REAL*8    RCUT              PAIR POTENTIAL CUTOFF
! REAL*8    V                 POTENTIAL ENERGY
! REAL*8    VC                SHIFTED POTENTIAL
! REAL*8    W                 VIRIAL FUNCTION
! REAL*8    VIJ               PAIR POTENTIAL BETWEEN I AND J
! REAL*8    WIJ               NEGATIVE OF PAIR VIRIAL FUNCTION W
! *************************************************************

      INTEGER N, PERIODIC
      REAL*8        RCUT, BOX, V, VC, W
      REAL*8        RX(N), RY(N), RZ(N), FX(N), FY(N), FZ(N)
      INTEGER     I, J, NCUT
      REAL*8        BOXINV, RCUTSQ
      REAL*8        RXI, RYI, RZI, FXI, FYI, FZI
      REAL*8        RXIJ, RYIJ, RZIJ, FXIJ, FYIJ, FZIJ
      REAL*8        RIJSQ, SR2, SR6, SR12, VIJ, WIJ, FIJ

C
*****************************************************************
C     ** USEFUL QUANTITIES **

      BOXINV = 1.0 / BOX
      RCUTSQ = RCUT ** 2

C     ** ZERO FORCES, POTENTIAL, VIRIAL **

      FX(:) = 0.0
```

```
         FY(:) = 0.0
         FZ(:) = 0.0

         NCUT = 0
         V    = 0.0
         W    = 0.0

C     ** OUTER LOOP BEGINS **

         DO I = 1, N - 1

            RXI = RX(I)
            RYI = RY(I)
            RZI = RZ(I)
            FXI = FX(I)
            FYI = FY(I)
            FZI = FZ(I)

C        ** INNER LOOP BEGINS **

            DO J = I + 1, N

               RXIJ = RXI - RX(J)
               RYIJ = RYI - RY(J)
               RZIJ = RZI - RZ(J)
               RXIJ = RXIJ - ANINT ( RXIJ * BOXINV ) * BOX * PERIODIC
               RYIJ = RYIJ - ANINT ( RYIJ * BOXINV ) * BOX * PERIODIC
               RZIJ = RZIJ - ANINT ( RZIJ * BOXINV ) * BOX * PERIODIC
               RIJSQ  = RXIJ ** 2 + RYIJ ** 2 + RZIJ ** 2

               IF ( RIJSQ .LT. RCUTSQ ) THEN

                  SR2   = 1.0 / RIJSQ
                  SR6   = SR2 * SR2 * SR2
                  SR12  = SR6 ** 2
                  VIJ   = SR12 - SR6
                  V     = V + VIJ
                  WIJ   = VIJ + SR12
                  W     = W + WIJ
                  FIJ   = WIJ * SR2
                  FXIJ  = FIJ * RXIJ
                  FYIJ  = FIJ * RYIJ
                  FZIJ  = FIJ * RZIJ
                  FXI   = FXI + FXIJ
                  FYI   = FYI + FYIJ
                  FZI   = FZI + FZIJ
                  FX(J) = FX(J) - FXIJ
                  FY(J) = FY(J) - FYIJ
                  FZ(J) = FZ(J) - FZIJ
                  NCUT  = NCUT + 1

               ENDIF

            ENDDO
```

```
C       ** INNER LOOP ENDS **

           FX(I) = FXI
           FY(I) = FYI
           FZ(I) = FZI

        ENDDO

C     ** OUTER LOOP ENDS **

C     ** CALCULATE SHIFTED POTENTIAL **

        SR2  = 1.0 / RCUTSQ
        SR6  = SR2 * SR2 * SR2
        SR12 = SR6 * SR6
        VIJ  = SR12 - SR6
        VC   = V - REAL ( NCUT ) * VIJ

C     ** MULTIPLY RESULTS BY NUMERICAL FACTORS **

           FX(:) = FX(:) * 24.0
           FY(:) = FY(:) * 24.0
           FZ(:) = FZ(:) * 24.0

        V  = V  * 4.0
        VC = VC * 4.0
        W  = W  * 24.0 / 3.0

        RETURN
        END
```