Crystal Structure Evidence for the Directionality of Lone Pair – π interactions – Fact or Fiction?

Chuandong Jia,[#1] Haohao Miao,[#1] and Benjamin P. Hay[1,2,*]

[1] Key Laboratory of Synthetic and Natural Functional Molecule Chemistry of the Ministry of Education, College of Chemistry and Materials Science,, Northwest University, Xi'an 710127, China

[2] Supramolecular Design Institute, Oak Ridge, TN 37839, USA

[#]These authors contributed equally.

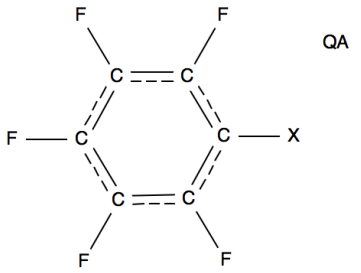
SUPPORTING INFORMATION

1. Search Criteria

1.1 Intermolecular lp atom contacts with C_6F_5X

Conquest software provided with the CSD subscription was used to query the CSD (database version 5.40 with updates through May 2019) and generate geometric parameters for 520,957 lone pair atoms that are in nonbonded contact with the centroid of a C_6F_5X group at distances ≤ 8.0 Å. These data were written to a file to create a master spreadsheet. Different subsets of data were then extracted from this master spreadsheet to produce the graphics and statistical analyses reported in the paper. This section gives step-by-step instructions needed to generate this master spreadsheet.

(1) After starting Conquest, go to Build Queries and select the Draw option. Draw the following molecule on the screen, where X indicates any atom and QA is a user-defined set of N, O, F, S, Cl, Br, and I lone pair atoms created using the More…/Other Elements tool.



(2) Select ADD 3D and specify the following items:

(a) Select QA then Define…/Atom Properties/Atomic Number

(b) Select X then Define…/Atom Properties/Atomic Number, Number of connected atoms, and Van der Waals radius

(c) Select all six ring C atoms and Plane: Define

(d) Select all six ring C atoms and Centroid: Define

(e) Select QA and Centroid and Vector: Define

(f) Select QA and Centroid then Distance: Define/Contact (to specify intermolecular interactions between these fragments)/OK/Define/From 0 to 8.0 Å range/OK/OK

(g) Select QA and Plane then Distance: Define/Any/OK

(h) Select Plane and Vector then Angle: Define

(i) Select QA and C atom attached to X then Distance: Define/Any/OK

(j) Repeat process (i) for the next five ring C atoms

(k) Select QA and X substituent atom then Distance: Define/Any/OK

(l) Repeat process (k) for the five F substituent atoms

(m) Select C atom attached to X and Centroid then Distance: Define

(n) Done

(3) Select Search and specify the following settings:

(a) 3D coordinates determined

(b) R factor <= 0.1

(c) Only non-disordered

(d) No errors

(e) Only single crystal structures

(f) Advanced Options/Normalize terminal H positions

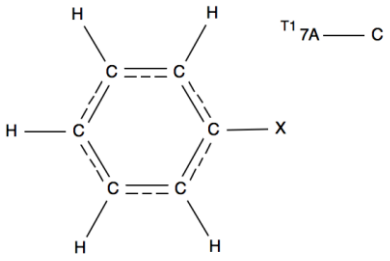
(4) Start Search

(5) After the search has completed (around 5 min), it should have found 6471 crystals in which one or more lone pair atoms meet the search criteria. The 3D data is exported to a spreadsheet by using Select File/Export Parameters and Data…/File type: Spreadsheet/Save/Save As: to create a .csv file.

(6) Use Microsoft® Excel (or an equivalent spreadsheet program) to edit this file. Delete the first column and the first row and save the rest as space-delimited text in a file. This file, which should contain 21 columns and 520,957 rows of data, is the master spreadsheet. Each row contains the following variables in this serial order: CSD reference code, atomic number of QA, atomic number of X, coordination number of X, van der Waals radius of X, distance of QA to arene centroid, distance of QA to arene plane, QA–centroid–plane angle, distances from QA to C1 (bound to X), C2, C3, C4, C5, C6, X, F1, F2, F3, F4, F5, and distance from C1 to arene centroid.

1.2 Intermolecular halogen atom contacts with C_6H_5X

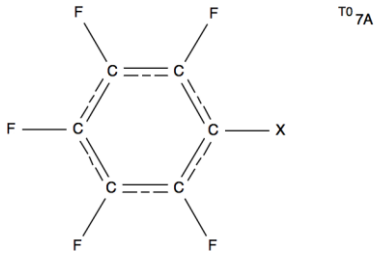
A strictly analogous procedure to that described in **1.1** was used to query the CSD and generate a master spreadsheet for lone pair atoms from R-Y (alkyl and aryl halides, Y = F, Cl, Br, I) that are in intermolecular contact with the centroid of a C_6H_5X group at distances ≤ 6.0 Å. In this case, the search fragment is defined as shown. Here X is any atom and the contacting atom, 7A, is any halogen atom. To specify that the search includes only terminal halogen atoms, the number of atoms bonded to 7A is constrained to be 1. Data produced by this search were used to generate the graphics shown in Figure 13.



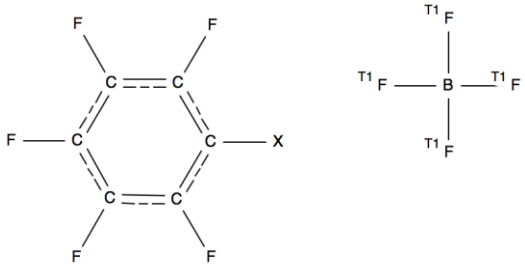
1.3 Intermolecular anion contacts with C_6F_5X

A strictly analogous procedure to that described in **1.1** was used to query the CSD and generate master spreadsheets for halide anions (F^-, Cl^-, Br^-, I^-), B atoms of BF_4^- anions, and Cl atoms of ClO_4^- anions that are in intermolecular contact with the centroid of a C_6F_5X group at distances ≤ 8.0 Å. Search fragments were defined as described below.

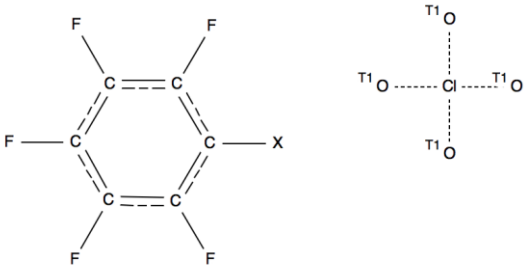
Halide anion search fragment: X is any atom and the contacting atom, 7A, is any halogen atom. To specify that the search includes only halide anions, the number of atoms bonded to 7A is constrained to be 0. The spreadsheet produced by this search was used to generate the graphics shown in Figure 15(a).



BF_4^- search fragment: X is any atom and the number of atoms bonded to each anionic F atom is constrained to be 1. The 3D data is defined using the center of the anion, in other words, the B atom. The spreadsheet produced by this search was used to generate the graphics shown in Figure 15(b).



ClO_4^- search fragment: X is any atom, the Cl–O bonds are designated with the type *any*, and the number of atoms bonded to each anionic O atom is constrained to be 1. The 3D data is defined using the center of the anion, in other words, the Cl atom. The spreadsheet produced by this search was used to generate the graphics shown in Figure 15(c).



2. Extracting data subsets from master spreadsheet

A FORTRAN code named SUBSET, which is included as part of this Supporting Information, was used to read a master spreadsheet and generate subsets of contact data. Distances between the contacting atom and each of the 12 arene atoms are compared with distances calculated using reference van der Waals radii to determine whether a contacting atom should be included within a given data subset. The default van der Waals radii given in Tables 1 and 2 of the paper are used for every atom except for the B atom in BF_4^- and the Cl atom in ClO_4^-. Effective van der Waals radii for these tetrahedral anions were estimated by taking the average of the center to vertex distance and the center to face distance and adding the radius of the substituent atom to give values of 2.38 Å for BF_4^- and 2.46 Å for ClO_4^-. To determine the location of a contacting atom, the code computes Cartesian coordinates as described in the paper (see Figure 2). Finally, a density is computed for each contact. This density is defined as the number of other contacts located within a 0.20 Å radius sphere about the contacting atom.

Twelve output files are written using the following format. The first line contains an integer giving the number of atoms, *natom*, in the file. Then line 2 to line *natom*+1 each contains the atom label x, y, z, density, distance to centroid, distance to plane, and the offset value R for the contacting atom, as well as the atomic number and coordination number for the substituent X atom. As described below, these files contain the different subsets of contact data that are discussed in the paper. Cases where one of these output files was used to generate a Figure or Table in the paper are indicated.

(a) *vdw+X.X*, where X.X = 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, and 1.0: This group of seven output files contain subsets of atoms that contact at least one of the 12 arene atoms at a distance ≤ Σr_{vdw} + X.X Å. If any one of the distances between the atom and one of the 12 arene atoms is ≤ Σr_{vdw} + X.X Å, then data for the atom is included in the corresponding output file. These data subsets were used to create the Tables 1 and 2 (*vdw+0.0*), Table 3 (*vdw+0.0*, *vdw+0.1*, *vdw+0.2*, *vdw+0.3*, *vdw+0.4*, *vdw+0.5*), Figures 3 and 4 (*vdw+0.0*), Figures 5, 6, and 8 (*vdw+0.0*, *vdw+0.1*, *vdw+0.2*), Figure 9 (b) and (c) (*vdw+0.0*, *vdw+1.0*), Figure 11(b) (*vdw+0.0*), and Figure 12(a) and 13(a) (*vdw+1.0*).

(b) *2002_data*: Subset of atoms that simultaneously contact all six ring carbon atoms at distances ≤ Σr_{vdw}. This is the search criterion reported in Quinoñero, D.; Garau, C.; Rotger, C.; Frontera, A.; Ballester, P.; Costa, A.; Deyà, P. M. Anion-π Interactions: Do They Exist? *Angew. Chem. Int. Ed.* **2002**, *41*, 3389 -3392.

(c) *actual_2002_data*: Subset of atoms that simultaneously contact all six ring carbon atoms at distances ≤ Σr_{vdw} + 1.0 Å. This is the search criterion actually used to obtain the results reported in Quinoñero, D.; Garau, C.; Rotger, C.; Frontera, A.; Ballester, P.; Costa, A.; Deyà, P. M. Anion-π Interactions: Do They Exist? *Angew. Chem. Int. Ed.* **2002**, *41*, 3389 -3392. This data was used to create Figures 9(d), 10 and 11(a).

(d) *2008_data*: Subset of atoms that simultaneously contact all six ring carbon atoms at distances ≤ 4.0 Å. This search criterion was used to obtain the results reported in Mooibroek, T. J.; Gamez, P.; Reedijk, J. Lone Pair – π Interactions: A New Supramolecular Bond? *CrystEngComm* **2008**, *10*, 1501-1515. This data was used to create Figures 12(b), 12(c), 13(b), and 13(c).

(e) *2011_data*: Subset of atoms in contact with at least one of the six ring C atoms at a distance ≤ Σr_{vdw}. This search criterion was used to obtain the results reported in Frontera, A.; Gamez, P.; Mascal, M.; Mooibroek, T. J.; Reedijk, J. Putting Anion-π Interactions into Perspective. *Angew. Chem. Int. Ed.* **2011**, *50*, 9564-9583. This data was used to create Figures 14 and 15.

(f) *2012_data*: Subset of atoms that are ≤ 5.0 Å from the C_6 centroid. This search criterion was used to obtain the results reported in Mooibroek, T. J.; Gamez, P. Anion-Arene and Lone Pair-

Arene Interactions are Directional. *CrystEngComm* **2012**, *14*, 1027-1030. This data was used to create Figures 16(a), 17(a), and 18(a).

(g) *2012_vdw*: Subset of atoms that are both ≤ 5.0 Å from the C_6 centroid and contacting least one of the 12 arene atoms at a distance ≤ Σr_{vdw}. This data was used to create Figures 17(b) and 18(b).

3. Generating random atom distributions

A FORTRAN code named RANDOM, which is included as part of this Supporting Information, was used to generate random distributions of F atoms within different volumes about the C_6F_6 arene. Three output files are produced with the names *random_shell*, *random_2012*, and *random_2012_vdw*. The RANDOM program asks for 2 input variables, the number of atoms, *natom*, and *X.X*, which is a value added to the Σr_{vdw} to specify the maximum contact distance to be used when creating a *random_shell* distribution. The algorithm uses the rand() function to generate random x, y, and z coordinates for a F atom within a 6.0 x 6.0 x 6.0 Å cube that is centered at the origin and orthogonal to the x, y, and z axes. Consistent with the reference geometry defined in Figure 2 of the paper, atomic coordinates are assigned for a C_6F_6 molecule that lies in the x,y plane with the centroid at the origin and one of the C–F bonds oriented along the +x axis. Distances from the randomly generated F atom to each C_6F_6 atom and to the C_6F_6 centroid are calculated. In all types of distribution, if any F atom---atom distance is < Σr_{vdw} – 0.2 Å, then the F atom is considered to be too close to the arene and is rejected. With the *random_shell* distribution, if all F atom---atom distances are > Σr_{vdw} + X.X Å, then the F atom is too far from the arene surface and is rejected. With the *random_2012* distribution, if the F atom---centroid distance is > 5.0 Å, then the F atom is outside the search volume and is rejected. With the random_2012_vdw distribution, if the F atom---centroid distance is either > 5.0 Å or if any F atom---atom distance > Σr_{vdw}, then the F atom is rejected. When a randomly generated F atom location passes all tests, then the atom is added to the output file. The process is completed until the requested number of atoms, *natoms*, has been reached.

Three output files are written. As described below, these files contain the different subsets of random contact data that are discussed in the paper. Cases where random data sets contained in these output files were used to generate Tables or Figures in the paper are indicated below.

(a) *random_shell*: File containing a set of *natom* F atoms randomly distributed about C_6F_5X such that all atom---atom distances > Σr_{vdw} – 0.2 Å and with at least one atom---atom distance ≤ Σr_{vdw} + *X.X* Å. This type of output file provided data presented in Table 3, Figure 4(b), and Figure 6 and was used by MAKEVIEW to generate the POVRay input file for Figure 3(b).

(b) *random_2012*: File containing a set of *natom* F atoms randomly distributed about C_6F_5X with all atom---atom distances > Σr_{vdw} – 0.2 Å and all atom---centroid distances ≤ 5.0 Å. This type of output file provided data presented in Figures 17(a) and 18(a) and was used by MAKEVIEW to generate the POVRay input file for Figure 16(b).

(c) *random_2012_vdw*: File containing a set of *natom* F atoms randomly distributed about C_6F_5X with all atom---centroid distances ≤ 5.0 Å, all atom---atom distances > Σr_{vdw} – 0.2 Å, and at least one atom---atom distance ≤ Σr_{vdw}. This type of output file provided data presented in Figures 17(b) and 18(b).

4. Generating 3D graphics input files

The 3D graphics shown in Figures 3, 5, 9, 12, 13, 14, and 16 of the paper were generated using the POVRay software available at no charge from the www.povray.org website. A FORTRAN code named MAKEVIEW, which is included as part of this Supporting Information, was used to generate POVRay input files needed to make these graphics. This code reads one of the data files generated by SUBSET or RANDOM and generates a number of different output files. These output files are described below. Cases where one of these POVRay input files was used to generate a graphic that appears in the paper are indicated.

(a) *cloud_ball.pov*: Contact distribution about a ball-and-stick model of C_6F_5X. This POVRay input file was used to create the graphics in a number of Figures when MAKEVIEW is run using various data subsets generated by SUBSET including Figure 3(a) using the *vdw+0.0* file, Figure 9(d) using the *actual_2002_data* file, Figures 12(b) and 13(b) using the *2008_data files*, Figure

14(a) using the *2011_data* file, and Figure 16(a) using the *2012_data* file or generated by RANDOM including Figure 3(b) using XXXXXX and Figure 16(b) using XXXXXX file.

(b) *cloud_tube.pov*: Contact distribution about a tube model of C_6F_5X. Not used in paper.

(c) *cloud_space.pov*: Contact distribution about a space-filling model of C_6F_5X. Not used in paper.

(d) *dense_ball.pov*: Contact density about a ball-and-stick model of C_6F_5X. Not used in paper.

(e) *dense_tube.pov*: Contact density about a tube model of C_6F_5X. Figures 5(a), 5(b), and 5(c) are generated with this POVRay input file when MAKEVIEW is run with *vdw+0.0*, *vdw+0.1*, and *vdw+0.2 files* created by SUBSET, respectively.

(f) *dense_space.pov*: Contact density about a space-filling model of C_6F_5X. Figure 14(b) is generated with this POVRay input file when MAKEVIEW is run with the *2011_data* file created by SUBSET.

(g) *2002.pov*: Cut-away view of a representative bullet-shaped search volume within the contact distribution about a ball-and-stick model of C_6F_5X. This illustrates the search volume for F atom contacts reported in Quinoñero, D.; Garau, C.; Rotger, C.; Frontera, A.; Ballester, P.; Costa, A.; Deyà, P. M. Anion-π Interactions: Do They Exist? *Angew. Chem. Int. Ed.* **2002**, *41*, 3389-3392. Figure 9(b) is generated with this POVRay input file when MAKEVIEW is run with the *vdw+0.0* data file created by SUBSET.

(h) *actual_2002.pov*: Cut-away view of a representative bullet-shaped search volume within the contact distribution about a ball-and-stick model of C_6F_5X. This illustrates the search volume for F atom contacts actually used to obtain the results reported in Quinoñero, D.; Garau, C.; Rotger, C.; Frontera, A.; Ballester, P.; Costa, A.; Deyà, P. M. Anion-π Interactions: Do They Exist? *Angew. Chem. Int. Ed.* **2002**, *41*, 3389-3392. Figure 9(c) is generated with this POVRay input file when MAKEVIEW is run with the *vdw+1.0* data file created by SUBSET.

(i) *2008.pov*: Cut-away view of the bullet-shaped search volume within contact distribution about a ball-and-stick model of C_6F_5X. This illustrates the search volume used to obtain results reported in Mooibroek, T. J.; Gamez, P.; Reedijk, J. Lone Pair – π Interactions: A New Supramolecular Bond? *CrystEngComm* **2008**, *10*, 1501-1515. Figures 12(a) and 13(a) were

generated with this POVRay input file when MAKEVIEW is run with *vdw+1.0* data files created by SUBSET.

In addition to POVRay input files, the code MAKEVIEW writes an additional output file named *angles*. The *angles* file contains a list of all contacting atom – arene centroid – C atom angles. Using *actual_2002_data* and *vdw+0.0* files generated by SUBSET, the *angles* file was used to create Figure 11(a) and 11(b), respectively.

5. Fortran source codes

Fortran source code files needed to create executables for the SUBSET, MAKEVIEW, and RANDOM programs are given below. The GNU Fortran compiler, *gfortran*, which is available at no charge from https://gcc.gnu.org, was used to create the executables with the following command:

gfortran *name.f* -o *name*

5.1 SUBSET source code – subset.f

```
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
c                              SUBSET
c               Copyright (C) 2019 by Benjamin P Hay
c
c     This program is free software: you can redistribute it and/or
c     modify it under the terms of the GNU General Public License as
c     published by the Free Software Foundation, either version 3 of the
c     License or (at your option) any later version.
c
c     This program is distributed in the hope that it will be useful,
c     but WITHOUT ANY WARRANTY; without even the implied warranty of
c     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See GNU
c     General Public License for more details at this address:
c     <http://www.gnu.org/licenses/>.
c
c     Benjamin P. Hay
c     Supramolecular Design Institute
c     127 Chestnut Hill Road
c     Oak Ridge, TN 37830
c
c     hayben@comcast.net
c
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
c     This program is distributed as part of the Supporting Information
c     for the Cryst. Growth Des. 2019 article entitled: Crystal
c     structure evidence for the directionality of lone pair-pi
c     interactions - fact or fiction?  It reads a data spreadsheet
c     generated by the CSD, computes Cartesian coordinates for each
c     contact, and writes files containing various data subsets
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
```

```
      program subset

      implicit none

      integer MAXPTS
      parameter (MAXPTS=550000)

      integer i,j,mark,numhits,ncontact
      integer atno(MAXPTS),atnox(MAXPTS),cnx(MAXPTS)

      double precision qm(MAXPTS),qpl(MAXPTS),tilt(MAXPTS)
      double precision qc1(MAXPTS),qc2(MAXPTS),qc3(MAXPTS)
      double precision qc4(MAXPTS),qc5(MAXPTS),qc6(MAXPTS)
      double precision qx(MAXPTS),qf2(MAXPTS),qf3(MAXPTS)
      double precision qf4(MAXPTS),qf5(MAXPTS),qf6(MAXPTS)
      double precision c1m(MAXPTS),density(MAXPTS),sphere,dist
      double precision offset(MAXPTS),leg1(MAXPTS)
      double precision theta(MAXPTS),ptheta
      double precision angle,vdwq(MAXPTS),vdwx(MAXPTS)
      double precision x(MAXPTS),y(MAXPTS),z(MAXPTS)
      double precision vdwqc,vdwqf,vdwqx

      character*1 iarene,inkey
      character*2 labq(MAXPTS)
      character*8 ref(MAXPTS)
      character*80 input
      character*300 aline

      logical flag,perchlorate

c--------------------------------------------------------------------------
c     get input file name
c--------------------------------------------------------------------------

    5 continue

      write(6,'(a)') 'Enter the name of the input file:'
      write(6,'(a)') ' '
      read(5,'(a80)') input

      flag = .false.
      inquire(file=input,exist=flag)
      if(.not.flag) then
         write(6,'(a,a)') 'Do not find file named ',input
         write(6,'(a)') 'Check file name and try again '
         write(6,'(a)') ' '
         goto 5
      end if

c--------------------------------------------------------------------------
c     initialize
c--------------------------------------------------------------------------

      do i = 1, MAXPTS
         ref(i) = ' '            ! CCSD reference code
         atno(i) = 0             ! atomic number of contacting atom, q
         atnox(i) = 0            ! atomic number of X
         cnx(i) = 0              ! atomic number of X
         vdwq(i) = 0.0d0         ! vdw radius of q
         vdwx(i) = 0.0d0         ! vdw radius of substituent on c1, x
         qm(i) = 0.0d0           ! distance from q to centroid
         qpl(i) = 0.0d0          ! distance from q to arene plane
         tilt(i) = 0.0d0         ! q-centroid-plane angle
         qc1(i) = 0.0d0          ! distance from q to c1
         qc2(i) = 0.0d0          ! distance from q to c2
         qc3(i) = 0.0d0          ! distance from q to c3
         qc4(i) = 0.0d0          ! distance from q to c4
         qc5(i) = 0.0d0          ! distance from q to c5
         qc6(i) = 0.0d0          ! distance from q to c6
         qx(i) = 0.0d0           ! distance from q to x
         qf2(i) = 0.0d0          ! distance from q to f2
```

```
         qf3(i) = 0.0d0          ! distance from q to f3
         qf4(i) = 0.0d0          ! distance from q to f4
         qf5(i) = 0.0d0          ! distance from q to f5
         qf6(i) = 0.0d0          ! distance from q to f6
         c1m(i) = 0.0d0          ! distance from c1 to centroid
         offset(i) = 0.0d0       ! offset
         density(i) = 0.0d0      ! contact density
         leg1(i) = 0.0d0         ! projection of q-c1 onto plane
         x(i) = 0.0d0            ! x coordinate for q
         y(i) = 0.0d0            ! y coordinate for q
         z(i) = 0.0d0            ! z coordinate for q
         theta(i) = 0.0d0        ! angle theta
      end do

c-----------------------------------------------------------------------
c     inquire arene identity
c-----------------------------------------------------------------------

   10 continue
      write(6,'(a)') ' '
      write(6,'(a)') ' '
      write(6,'(a)') 'Arene Type:'
      write(6,'(a)') '(1) C6F5X'
      write(6,'(a)') '(2) C6H5X'
      write(6,'(a)') ' '
      write(6,'(a)') 'enter 1 or 2:'
      write(6,'(a)') ' '
      read(5,'(a1)')  iarene
      if((iarene.ne.'1').and.(iarene.ne.'2')) then
         goto 10
      endif

c-----------------------------------------------------------------------
c     lone pair atom or tetrahedral anion contact?
c-----------------------------------------------------------------------

   15 continue
      write(6,'(a)') ' '
      write(6,'(a)') ' '
      write(6,'(a)') 'Type of contacting atom:'
      write(6,'(a)') '(1) perchlorate'
      write(6,'(a)') '(2) anything else'
      write(6,'(a)') ' '
      write(6,'(a)') 'enter 1 or 2:'
      write(6,'(a)') ' '
      read(5,'(a1)') inkey
      if(inkey.eq.'1') then
         perchlorate = .true.
      elseif(inkey.eq.'2') then
         perchlorate = .false.
      else
         goto 15
      endif

c-----------------------------------------------------------------------
c     read data from input file, each line has 21 variables:
c     ref,atnoq,atnox,cnx,vdwx,qm,qpl,tilt,qc1,qc2,qc3,qc4,qc5,qc6,
c     qx,qf2,qf3,qf4,qf5,qf6,c1m
c-----------------------------------------------------------------------

      numhits = 0
      open(unit=10,file=input,status='old')

      do i = 1, MAXPTS

         read(10,'(A400)',err=100,end=200) aline

         read(aline,*) ref(i),atno(i),atnox(i),cnx(i),vdwx(i),
     &    qm(i),qpl(i),tilt(i),qc1(i),qc2(i),qc3(i),qc4(i),qc5(i),
     &    qc6(i),qx(i),qf2(i),qf3(i),qf4(i),qf5(i),qf6(i),c1m(i)
```

```
         numhits = numhits+1

         print*,' '
         print *, numhits

c--------------------------------------------------------------------------
c     if difference between qm(i) and qpl(i) is small then
c        offset = qpl(i) cos(tilt)
c     else
c        offset = dsqrt(qm(i)*qm(i) - qpl(i)*qpl(i))
c--------------------------------------------------------------------------

         tilt(i) = 90.0d0 - tilt(i)

         if(qm(i)*qm(i).lt.qpl(i)*qpl(i)) qpl(i) = qm(i)

         if(abs(qm(i)-qpl(i)).le.0.005d0) then
            offset(i) = qpl(i) * dcos(tilt(i)/57.296d0)
         else
            offset(i) = dsqrt(qm(i)*qm(i) - qpl(i)*qpl(i))
         end if

         if(qc1(i)*qc1(i).lt.qpl(i)*qpl(i)) qpl(i) = qc1(i)
         leg1(i) = dsqrt(qc1(i)*qc1(i) - qpl(i)*qpl(i))

         ptheta = ( ( (offset(i)*offset(i)) + (c1m(i)*c1m(i))
     &    - (leg1(i)*leg1(i)) ) / (2*c1m(i)*offset(i)) )

c--------------------------------------------------------------------------
c     If ptheta < -1 or ptheta > 1, then arc cosine is undefined
c     To handle this situation, if ptheta < -1, then ptheta = -1
c     and if ptheta > 1, then ptheta = 1.   Set mark = 1
c--------------------------------------------------------------------------

         mark = 0
         if(ptheta.gt.1.00d0) then
            ptheta = 1.00000d0
            mark = 1
         end if
         if(ptheta.lt.-1.00d0) then
            ptheta = -1.00000d0
            mark = 1
         end if

         angle = dacos(ptheta)
         x(i) = offset(i)*dcos(angle)

c--------------------------------------------------------------------------
c     If mark = 1, then apply a small random offset about x-axis
c--------------------------------------------------------------------------

         if(mark.eq.1) then
            if(dabs(x(i)).gt.1.0d0) then
               y(i) = 0.00d0 + 0.20d0*rand()
            else
               y(i) = dabs(x(i))*(0.00d0+0.15d0*rand())
            end if
         else
            y(i) = offset(i)*dsin(angle)
         end if

         z(i) = qpl(i)

         theta(i) = angle*180.0000d0/3.141592654d0

c--------------------------------------------------------------------------
c     Use atomic number to assign label and vdw radius to lone pair
c     atom.  Effective vdw radii for BF4 and ClO4 anion tetrahedra are
c     estimated as the average of the distances to the vertex + center
c     of a face plus the vdw radius of the substituent.
c--------------------------------------------------------------------------
```

```
         if(atno(i).eq.5) then
            labq(i) = 'B '
            vdwq(i) = 2.38d0
         elseif(atno(i).eq.7) then
            labq(i) = 'N '
            vdwq(i) = 1.55d0
         elseif(atno(i).eq.8) then
            labq(i) = 'O '
            vdwq(i) = 1.52d0
         elseif(atno(i).eq.9) then
            labq(i) = 'F '
            vdwq(i) = 1.47d0
         elseif(atno(i).eq.16) then
            labq(i) = 'S '
            vdwq(i) = 1.80d0
         elseif(atno(i).eq.17) then
            labq(i) = 'Cl'
            vdwq(i) = 1.75d0
            if(perchlorate) vdwq(i) = 2.46d0
         elseif(atno(i).eq.35) then
            labq(i) = 'Br'
            vdwq(i) = 1.85d0
         elseif(atno(i).eq.53) then
            labq(i) = 'I '
            vdwq(i) = 1.98d0
         else
            write(6,'(a,i3)') 'Unknown atomic number ',atno(i)
            stop
         end if

      end do

  200 print *,'end of file'
      close(10)

c--------------------------------------------------------------------------
c     do density calculations
c--------------------------------------------------------------------------

      sphere = 0.20d0
      sphere = sphere*sphere

      do i = 1, numhits
         density(i) = 0.0d0
         print*,'  i',i
         do j = 1, numhits
            dist = (x(i)-x(j))*(x(i)-x(j)) +
     &              (y(i)-y(j))*(y(i)-y(j)) +
     &              (z(i)-z(j))*(z(i)-z(j))
            if(dist.lt.sphere) then
               density(i) = density(i) + 1.0d0
            end if
         end do
      end do

c--------------------------------------------------------------------------
c     Make last half of offset(i) and y(i) negative
c--------------------------------------------------------------------------

      do i = numhits/2, numhits
         offset(i) = -1.0d0*offset(i)
         y(i) = -1.0d0*y(i)
      end do

c--------------------------------------------------------------------------
c     Write vdw+0.0 file - contains all points within the sum of vdw
c     radii of at least one atom of the arene
c--------------------------------------------------------------------------

      ncontact = 0
```

```
      do i = 1, numhits
         vdwqc = vdwq(i) + 1.70d0
         if(iarene.eq.'1') then          ! C6F5X
            vdwqf = vdwq(i) + 1.47d0
         elseif(iarene.eq.'2') then      ! C6H5X
            vdwqf = vdwq(i) + 1.20d0
         end if
         vdwqx = vdwq(i) + vdwx(i)
         if ( (qc1(i).le.vdwqc).or.(qc2(i).le.vdwqc).or.
     &        (qc3(i).le.vdwqc).or.(qc4(i).le.vdwqc).or.
     &        (qc5(i).le.vdwqc).or.(qc6(i).le.vdwqc).or.
     &        ( qx(i).le.vdwqx).or.(qf2(i).le.vdwqf).or.
     &        (qf3(i).le.vdwqf).or.(qf4(i).le.vdwqf).or.
     &        (qf5(i).le.vdwqf).or.(qf6(i).le.vdwqf) ) then
            ncontact = ncontact + 1
         end if
      end do

      open(unit=11,file='vdw+0.0')
         write(11,*) ncontact
         do i = 1, numhits
            vdwqc = vdwq(i) + 1.70d0
            if(iarene.eq.'1') then          ! C6F5X
               vdwqf = vdwq(i) + 1.47d0
            elseif(iarene.eq.'2') then      ! C6H5X
               vdwqf = vdwq(i) + 1.20d0
            end if
            vdwqx = vdwq(i) + vdwx(i)
            if ( (qc1(i).le.vdwqc).or.(qc2(i).le.vdwqc).or.
     &           (qc3(i).le.vdwqc).or.(qc4(i).le.vdwqc).or.
     &           (qc5(i).le.vdwqc).or.(qc6(i).le.vdwqc).or.
     &           ( qx(i).le.vdwqx).or.(qf2(i).le.vdwqf).or.
     &           (qf3(i).le.vdwqf).or.(qf4(i).le.vdwqf).or.
     &           (qf5(i).le.vdwqf).or.(qf6(i).le.vdwqf) ) then
               write(11,11) labq(i),x(i),y(i),z(i),density(i),
     &         ref(i),qm(i),qpl(i),offset(i),atnox(i),cnx(i)
            end if
         end do
      close(11)

   11 FORMAT(A2,X,4F8.3,X,A8,X,3F8.3,X,I5,X,I5)

c-------------------------------------------------------------------------
c     Write vdw+0.1 file - contains all points within the sum of
c     vdw radii + 0.1 of at least one atom of the C6A5X moeity
c-------------------------------------------------------------------------

      ncontact = 0
      do i = 1, numhits
         vdwqc = vdwq(i) + 1.70d0 + 0.1d0
         if(iarene.eq.'1') then          ! C6F5X
            vdwqf = vdwq(i) + 1.47d0 + 0.1d0
         elseif(iarene.eq.'2') then      ! C6H5X
            vdwqf = vdwq(i) + 1.20d0 + 0.1d0
         end if
         vdwqx = vdwq(i) + vdwx(i) + 0.1d0
         if ( (qc1(i).le.vdwqc).or.(qc2(i).le.vdwqc).or.
     &        (qc3(i).le.vdwqc).or.(qc4(i).le.vdwqc).or.
     &        (qc5(i).le.vdwqc).or.(qc6(i).le.vdwqc).or.
     &        ( qx(i).le.vdwqx).or.(qf2(i).le.vdwqf).or.
     &        (qf3(i).le.vdwqf).or.(qf4(i).le.vdwqf).or.
     &        (qf5(i).le.vdwqf).or.(qf6(i).le.vdwqf) ) then
            ncontact = ncontact + 1
         end if
      end do

      open(unit=11,file='vdw+0.1')
         write(11,*) ncontact
         do i = 1, numhits
            vdwqc = vdwq(i) + 1.70d0 + 0.1d0
            if(iarene.eq.'1') then          ! C6F5X
```

```
               vdwqf = vdwq(i) + 1.47d0 + 0.1d0
            elseif(iarene.eq.'2') then      ! C6H5X
               vdwqf = vdwq(i) + 1.20d0 + 0.1d0
            end if
            vdwqx = vdwq(i) + vdwx(i) + 0.1d0
            if ( (qc1(i).le.vdwqc).or.(qc2(i).le.vdwqc).or.
     &           (qc3(i).le.vdwqc).or.(qc4(i).le.vdwqc).or.
     &           (qc5(i).le.vdwqc).or.(qc6(i).le.vdwqc).or.
     &           ( qx(i).le.vdwqx).or.(qf2(i).le.vdwqf).or.
     &           (qf3(i).le.vdwqf).or.(qf4(i).le.vdwqf).or.
     &           (qf5(i).le.vdwqf).or.(qf6(i).le.vdwqf) ) then
               write(11,11) labq(i),x(i),y(i),z(i),density(i),
     &         ref(i),qm(i),qpl(i),offset(i),atnox(i),cnx(i)
            end if
         end do
       close(11)

c--------------------------------------------------------------------------
c     Write vdw+0.2 file - contains all points within the sum of
c     vdw radii + 0.2 of at least one atom of the C6A5X moeity
c--------------------------------------------------------------------------

       ncontact = 0
       do i = 1, numhits
          vdwqc = vdwq(i) + 1.70d0 + 0.2d0
          if(iarene.eq.'1') then          ! C6F5X
             vdwqf = vdwq(i) + 1.47d0 + 0.2d0
          elseif(iarene.eq.'2') then      ! C6H5X
             vdwqf = vdwq(i) + 1.20d0 + 0.2d0
          end if
          vdwqx = vdwq(i) + vdwx(i) + 0.2d0
          if ( (qc1(i).le.vdwqc).or.(qc2(i).le.vdwqc).or.
     &         (qc3(i).le.vdwqc).or.(qc4(i).le.vdwqc).or.
     &         (qc5(i).le.vdwqc).or.(qc6(i).le.vdwqc).or.
     &         ( qx(i).le.vdwqx).or.(qf2(i).le.vdwqf).or.
     &         (qf3(i).le.vdwqf).or.(qf4(i).le.vdwqf).or.
     &         (qf5(i).le.vdwqf).or.(qf6(i).le.vdwqf) ) then
             ncontact = ncontact + 1
          end if
       end do

       open(unit=11,file='vdw+0.2')
          write(11,*) ncontact
          do i = 1, numhits
             vdwqc = vdwq(i) + 1.70d0 + 0.2d0
             if(iarene.eq.'1') then          ! C6F5X
                vdwqf = vdwq(i) + 1.47d0 + 0.2d0
             elseif(iarene.eq.'2') then      ! C6H5X
                vdwqf = vdwq(i) + 1.20d0 + 0.2d0
             end if
             vdwqx = vdwq(i) + vdwx(i) + 0.2d0
             if ( (qc1(i).le.vdwqc).or.(qc2(i).le.vdwqc).or.
     &            (qc3(i).le.vdwqc).or.(qc4(i).le.vdwqc).or.
     &            (qc5(i).le.vdwqc).or.(qc6(i).le.vdwqc).or.
     &            ( qx(i).le.vdwqx).or.(qf2(i).le.vdwqf).or.
     &            (qf3(i).le.vdwqf).or.(qf4(i).le.vdwqf).or.
     &            (qf5(i).le.vdwqf).or.(qf6(i).le.vdwqf) ) then
               write(11,11) labq(i),x(i),y(i),z(i),density(i),
     &         ref(i),qm(i),qpl(i),offset(i),atnox(i),cnx(i)
             end if
          end do
       close(11)

c--------------------------------------------------------------------------
c     Write vdw+0.3 file - contains all points within the sum of
c     vdw radii + 0.3 of at least one atom of the C6A5X moeity
c--------------------------------------------------------------------------

       ncontact = 0
       do i = 1, numhits
          vdwqc = vdwq(i) + 1.70d0 + 0.3d0
```

```
         if(iarene.eq.'1') then          ! C6F5X
            vdwqf = vdwq(i) + 1.47d0 + 0.3d0
         elseif(iarene.eq.'2') then      ! C6H5X
            vdwqf = vdwq(i) + 1.20d0 + 0.3d0
         end if
         vdwqx = vdwq(i) + vdwx(i) + 0.3d0
         if ( (qc1(i).le.vdwqc).or.(qc2(i).le.vdwqc).or.
     &        (qc3(i).le.vdwqc).or.(qc4(i).le.vdwqc).or.
     &        (qc5(i).le.vdwqc).or.(qc6(i).le.vdwqc).or.
     &        ( qx(i).le.vdwqx).or.(qf2(i).le.vdwqf).or.
     &        (qf3(i).le.vdwqf).or.(qf4(i).le.vdwqf).or.
     &        (qf5(i).le.vdwqf).or.(qf6(i).le.vdwqf) ) then
            ncontact = ncontact + 1
         end if
      end do

      open(unit=11,file='vdw+0.3')
         write(11,*) ncontact
         do i = 1, numhits
            vdwqc = vdwq(i) + 1.70d0 + 0.3d0
            if(iarene.eq.'1') then          ! C6F5X
               vdwqf = vdwq(i) + 1.47d0 + 0.3d0
            elseif(iarene.eq.'2') then      ! C6H5X
               vdwqf = vdwq(i) + 1.20d0 + 0.3d0
            end if
            vdwqx = vdwq(i) + vdwx(i) + 0.3d0
            if ( (qc1(i).le.vdwqc).or.(qc2(i).le.vdwqc).or.
     &           (qc3(i).le.vdwqc).or.(qc4(i).le.vdwqc).or.
     &           (qc5(i).le.vdwqc).or.(qc6(i).le.vdwqc).or.
     &           ( qx(i).le.vdwqx).or.(qf2(i).le.vdwqf).or.
     &           (qf3(i).le.vdwqf).or.(qf4(i).le.vdwqf).or.
     &           (qf5(i).le.vdwqf).or.(qf6(i).le.vdwqf) ) then
               write(11,11) labq(i),x(i),y(i),z(i),density(i),
     &         ref(i),qm(i),qpl(i),offset(i),atnox(i),cnx(i)
            end if
         end do
      close(11)

c-------------------------------------------------------------------------
c     Write vdw+0.4 file - contains all points within the sum of
c     vdw radii + 0.4 of at least one atom of the C6A5X moeity
c-------------------------------------------------------------------------

      ncontact = 0
      do i = 1, numhits
         vdwqc = vdwq(i) + 1.70d0 + 0.4d0
         if(iarene.eq.'1') then          ! C6F5X
            vdwqf = vdwq(i) + 1.47d0 + 0.4d0
         elseif(iarene.eq.'2') then      ! C6H5X
            vdwqf = vdwq(i) + 1.20d0 + 0.4d0
         end if
         vdwqx = vdwq(i) + vdwx(i) + 0.4d0
         if ( (qc1(i).le.vdwqc).or.(qc2(i).le.vdwqc).or.
     &        (qc3(i).le.vdwqc).or.(qc4(i).le.vdwqc).or.
     &        (qc5(i).le.vdwqc).or.(qc6(i).le.vdwqc).or.
     &        ( qx(i).le.vdwqx).or.(qf2(i).le.vdwqf).or.
     &        (qf3(i).le.vdwqf).or.(qf4(i).le.vdwqf).or.
     &        (qf5(i).le.vdwqf).or.(qf6(i).le.vdwqf) ) then
            ncontact = ncontact + 1
         end if
      end do

      open(unit=11,file='vdw+0.4')
         write(11,*) ncontact
         do i = 1, numhits
            vdwqc = vdwq(i) + 1.70d0 + 0.4d0
            if(iarene.eq.'1') then          ! C6F5X
               vdwqf = vdwq(i) + 1.47d0 + 0.4d0
            elseif(iarene.eq.'2') then      ! C6H5X
               vdwqf = vdwq(i) + 1.20d0 + 0.4d0
            end if
```

```
            vdwqx = vdwq(i) + vdwx(i) + 0.4d0
            if ( (qc1(i).le.vdwqc).or.(qc2(i).le.vdwqc).or.
     &           (qc3(i).le.vdwqc).or.(qc4(i).le.vdwqc).or.
     &           (qc5(i).le.vdwqc).or.(qc6(i).le.vdwqc).or.
     &           ( qx(i).le.vdwqx).or.(qf2(i).le.vdwqf).or.
     &           (qf3(i).le.vdwqf).or.(qf4(i).le.vdwqf).or.
     &           (qf5(i).le.vdwqf).or.(qf6(i).le.vdwqf) ) then
               write(11,11) labq(i),x(i),y(i),z(i),density(i),
     &         ref(i),qm(i),qpl(i),offset(i),atnox(i),cnx(i)
            end if
         end do
       close(11)

c--------------------------------------------------------------------------
c     Write vdw+0.5 file - contains all points within the sum of
c     vdw radii + 0.5 of at least one atom of the C6A5X moeity
c--------------------------------------------------------------------------

      ncontact = 0
      do i = 1, numhits
         vdwqc = vdwq(i) + 1.70d0 + 0.5d0
         if(iarene.eq.'1') then          ! C6F5X
            vdwqf = vdwq(i) + 1.47d0 + 0.5d0
         elseif(iarene.eq.'2') then      ! C6H5X
            vdwqf = vdwq(i) + 1.20d0 + 0.5d0
         end if
         vdwqx = vdwq(i) + vdwx(i) + 0.5d0
         if ( (qc1(i).le.vdwqc).or.(qc2(i).le.vdwqc).or.
     &        (qc3(i).le.vdwqc).or.(qc4(i).le.vdwqc).or.
     &        (qc5(i).le.vdwqc).or.(qc6(i).le.vdwqc).or.
     &        ( qx(i).le.vdwqx).or.(qf2(i).le.vdwqf).or.
     &        (qf3(i).le.vdwqf).or.(qf4(i).le.vdwqf).or.
     &        (qf5(i).le.vdwqf).or.(qf6(i).le.vdwqf) ) then
            ncontact = ncontact + 1
         end if
      end do

      open(unit=11,file='vdw+0.5')
         write(11,*) ncontact
         do i = 1, numhits
            vdwqc = vdwq(i) + 1.70d0 + 0.5d0
            if(iarene.eq.'1') then          ! C6F5X
               vdwqf = vdwq(i) + 1.47d0 + 0.5d0
            elseif(iarene.eq.'2') then      ! C6H5X
               vdwqf = vdwq(i) + 1.20d0 + 0.5d0
            end if
            vdwqx = vdwq(i) + vdwx(i) + 0.5d0
            if ( (qc1(i).le.vdwqc).or.(qc2(i).le.vdwqc).or.
     &           (qc3(i).le.vdwqc).or.(qc4(i).le.vdwqc).or.
     &           (qc5(i).le.vdwqc).or.(qc6(i).le.vdwqc).or.
     &           ( qx(i).le.vdwqx).or.(qf2(i).le.vdwqf).or.
     &           (qf3(i).le.vdwqf).or.(qf4(i).le.vdwqf).or.
     &           (qf5(i).le.vdwqf).or.(qf6(i).le.vdwqf) ) then
               write(11,11) labq(i),x(i),y(i),z(i),density(i),
     &         ref(i),qm(i),qpl(i),offset(i),atnox(i),cnx(i)
            end if
         end do
       close(11)

c--------------------------------------------------------------------------
c     Write vdw+1.0 file - contains all points within the sum of
c     vdw radii + 1.0 of at least one atom of the C6A5X moeity
c--------------------------------------------------------------------------

      ncontact = 0
      do i = 1, numhits
         vdwqc = vdwq(i) + 1.70d0 + 1.0d0
         if(iarene.eq.'1') then          ! C6F5X
            vdwqf = vdwq(i) + 1.47d0 + 1.0d0
         elseif(iarene.eq.'2') then      ! C6H5X
            vdwqf = vdwq(i) + 1.20d0 + 1.0d0
```

```
         end if
         vdwqx = vdwq(i) + vdwx(i) + 1.0d0
         if ( (qc1(i).le.vdwqc).or.(qc2(i).le.vdwqc).or.
     &        (qc3(i).le.vdwqc).or.(qc4(i).le.vdwqc).or.
     &        (qc5(i).le.vdwqc).or.(qc6(i).le.vdwqc).or.
     &        ( qx(i).le.vdwqx).or.(qf2(i).le.vdwqf).or.
     &        (qf3(i).le.vdwqf).or.(qf4(i).le.vdwqf).or.
     &        (qf5(i).le.vdwqf).or.(qf6(i).le.vdwqf) ) then
            ncontact = ncontact + 1
         end if
      end do

      open(unit=11,file='vdw+1.0')
         write(11,*) ncontact
         do i = 1, numhits
            vdwqc = vdwq(i) + 1.70d0 + 1.0d0
            if(iarene.eq.'1') then          ! C6F5X
               vdwqf = vdwq(i) + 1.47d0 + 1.0d0
            elseif(iarene.eq.'2') then      ! C6H5X
               vdwqf = vdwq(i) + 1.20d0 + 1.0d0
            end if
            vdwqx = vdwq(i) + vdwx(i) + 1.0d0
            if ( (qc1(i).le.vdwqc).or.(qc2(i).le.vdwqc).or.
     &           (qc3(i).le.vdwqc).or.(qc4(i).le.vdwqc).or.
     &           (qc5(i).le.vdwqc).or.(qc6(i).le.vdwqc).or.
     &           ( qx(i).le.vdwqx).or.(qf2(i).le.vdwqf).or.
     &           (qf3(i).le.vdwqf).or.(qf4(i).le.vdwqf).or.
     &           (qf5(i).le.vdwqf).or.(qf6(i).le.vdwqf) ) then
               write(11,11) labq(i),x(i),y(i),z(i),density(i),
     &         ref(i),qm(i),qpl(i),offset(i),atnox(i),cnx(i)
            end if
         end do
      close(11)

c---------------------------------------------------------------------
c     Write actual 2002_data file - contains points within the sum of
c     vdw radii + 1.0 from all six C atoms in the C6F5X group.
c     This is search criterion actually used in Angew. Chem. Int. Ed.
c     2002, 41, 3389
c---------------------------------------------------------------------

      ncontact = 0
      do i = 1, numhits
         vdwqc = vdwq(i) + 1.70d0 + 1.00d0
         if ( (qc1(i).le.vdwqc).and.(qc2(i).le.vdwqc).and.
     &        (qc3(i).le.vdwqc).and.(qc4(i).le.vdwqc).and.
     &        (qc5(i).le.vdwqc).and.(qc6(i).le.vdwqc) ) then
            ncontact = ncontact + 1
         end if
      end do

      open(unit=11,file='actual_2002_data')
         write(11,*) ncontact
         do i = 1, numhits
            vdwqc = vdwq(i) + 1.70d0 + 1.00d0
            if ( (qc1(i).le.vdwqc).and.(qc2(i).le.vdwqc).and.
     &           (qc3(i).le.vdwqc).and.(qc4(i).le.vdwqc).and.
     &           (qc5(i).le.vdwqc).and.(qc6(i).le.vdwqc) ) then
               write(11,11) labq(i),x(i),y(i),z(i),density(i),
     &         ref(i),qm(i),qpl(i),offset(i),atnox(i),cnx(i)
            end if
         end do
      close(11)

c---------------------------------------------------------------------
c     Write 2002_data file - contains all points within the sum of
c     vdw radii of every carbon atom in the C6F5X moeity.  This is
c     the incorrect search criterion reported in Angew. Chem.
c     Int. Ed. 2002, 41, 3389
c---------------------------------------------------------------------
```

```
      ncontact = 0
      do i = 1, numhits
         vdwqc = vdwq(i) + 1.70d0
         if ( (qc1(i).le.vdwqc).and.(qc2(i).le.vdwqc).and.
     &        (qc3(i).le.vdwqc).and.(qc4(i).le.vdwqc).and.
     &        (qc5(i).le.vdwqc).and.(qc6(i).le.vdwqc) ) then
            ncontact = ncontact + 1
         end if
      end do

      open(unit=11,file='2002_data')
         write(11,*) ncontact
         do i = 1, numhits
            vdwqc = vdwq(i) + 1.70d0
            if ( (qc1(i).le.vdwqc).and.(qc2(i).le.vdwqc).and.
     &           (qc3(i).le.vdwqc).and.(qc4(i).le.vdwqc).and.
     &           (qc5(i).le.vdwqc).and.(qc6(i).le.vdwqc) ) then
               write(11,11) labq(i),x(i),y(i),z(i),density(i),
     &         ref(i),qm(i),qpl(i),offset(i),atnox(i),cnx(i)
            end if
         end do
      close(11)

c--------------------------------------------------------------------------
c     Write 2008_data file - within 4 angstroms of each ring C atom in
c     C6F5X, the search criterion used in CrystEngComm 2008, 10, 1501
c--------------------------------------------------------------------------

      ncontact = 0
      do i = 1, numhits
         if(qm(i).le.4.0d0) then
            if ( (qc1(i).le.4.0d0).and.(qc2(i).le.4.0d0).and.
     &           (qc3(i).le.4.0d0).and.(qc4(i).le.4.0d0).and.
     &           (qc5(i).le.4.0d0).and.(qc6(i).le.4.0d0) ) then
               ncontact = ncontact + 1
            end if
         end if
      end do

      open(unit=11,file='2008_data')
         write(11,*) ncontact
         do i = 1, numhits
            if(qm(i).le.4.0d0) then
               if ( (qc1(i).le.4.0d0).and.(qc2(i).le.4.0d0).and.
     &              (qc3(i).le.4.0d0).and.(qc4(i).le.4.0d0).and.
     &              (qc5(i).le.4.0d0).and.(qc6(i).le.4.0d0) ) then
                  write(11,11) labq(i),x(i),y(i),z(i),density(i),
     &            ref(i),qm(i),qpl(i),offset(i),atnox(i),cnx(i)
               end if
            end if
         end do
      close(11)

c--------------------------------------------------------------------------
c     Write 2011_data file - contains all pts within the sum of vdw
c     radii from any ring carbon atom in the C6F5X moeity.  This is the
c     search criterion used in the Angew. Chem. Int. Ed. 2011, 50, 9564
c--------------------------------------------------------------------------

      ncontact = 0
      do i = 1, numhits
         vdwqc = vdwq(i) + 1.70d0
         if ( (qc1(i).le.vdwqc).or.(qc2(i).le.vdwqc).or.
     &        (qc3(i).le.vdwqc).or.(qc4(i).le.vdwqc).or.
     &        (qc5(i).le.vdwqc).or.(qc6(i).le.vdwqc) ) then
            ncontact = ncontact + 1
         end if
      end do

      open(unit=11,file='2011_data')
         write(11,*) ncontact
```

```
         do i = 1, numhits
            vdwqc = vdwq(i) + 1.70d0
            if ( (qc1(i).le.vdwqc).or.(qc2(i).le.vdwqc).or.
     &           (qc3(i).le.vdwqc).or.(qc4(i).le.vdwqc).or.
     &           (qc5(i).le.vdwqc).or.(qc6(i).le.vdwqc) ) then
               write(11,11) labq(i),x(i),y(i),z(i),density(i),
     &         ref(i),qm(i),qpl(i),offset(i),atnox(i),cnx(i)
            end if
         end do
       close(11)

c--------------------------------------------------------------------------
c     Write 2012_data file - contains all pts within 5 angstroms of
c     the C6F5X centroid.  This is the stated criterion used in
c     CrystEngComm 2012, 14, 1027
c--------------------------------------------------------------------------

       ncontact = 0
       do i = 1, numhits
          if(qm(i).le.5.0d0) then
             ncontact = ncontact + 1
          end if
       end do

       open(unit=11,file='2012_data')
          write(11,*) ncontact
          do i = 1, numhits
             if(qm(i).le.5.0d0) then
                write(11,11) labq(i),x(i),y(i),z(i),density(i),
     &         ref(i),qm(i),qpl(i),offset(i),atnox(i),cnx(i)
             end if
          end do
       close(11)

c--------------------------------------------------------------------------
c     Write 2012_vdw file - contains all pts within 5 angstroms of the
c     C6F5X centroid and in vdw contact with at least one C6F5X atom
c--------------------------------------------------------------------------

       ncontact = 0
       do i = 1, numhits
          vdwqc = vdwq(i) + 1.70d0
          if(iarene.eq.'1') then          ! C6F5X
             vdwqf = vdwq(i) + 1.47d0
          elseif(iarene.eq.'2') then      ! C6H5X
             vdwqf = vdwq(i) + 1.20d0
          end if
          vdwqx = vdwq(i) + vdwx(i)
          if(qm(i).le.5.0d0) then
             if ( (qc1(i).le.vdwqc).or.(qc2(i).le.vdwqc).or.
     &            (qc3(i).le.vdwqc).or.(qc4(i).le.vdwqc).or.
     &            (qc5(i).le.vdwqc).or.(qc6(i).le.vdwqc).or.
     &            ( qx(i).le.vdwqx).or.(qf2(i).le.vdwqf).or.
     &            (qf3(i).le.vdwqf).or.(qf4(i).le.vdwqf).or.
     &            (qf5(i).le.vdwqf).or.(qf6(i).le.vdwqf) ) then
                ncontact = ncontact + 1
             end if
          end if
       end do

       open(unit=11,file='2012_vdw')
          write(11,*) ncontact
          do i = 1, numhits
             vdwqc = vdwq(i) + 1.70d0
             if(iarene.eq.'1') then          ! C6F5X
                vdwqf = vdwq(i) + 1.47d0
             elseif(iarene.eq.'2') then      ! C6H5X
                vdwqf = vdwq(i) + 1.20d0
             end if
             vdwqx = vdwq(i) + vdwx(i)
             if(qm(i).le.5.0d0) then
```

```
               if ( (qc1(i).le.vdwqc).or.(qc2(i).le.vdwqc).or.
     &              (qc3(i).le.vdwqc).or.(qc4(i).le.vdwqc).or.
     &              (qc5(i).le.vdwqc).or.(qc6(i).le.vdwqc).or.
     &              ( qx(i).le.vdwqx).or.(qf2(i).le.vdwqf).or.
     &              (qf3(i).le.vdwqf).or.(qf4(i).le.vdwqf).or.
     &              (qf5(i).le.vdwqf).or.(qf6(i).le.vdwqf) ) then
                  write(11,11) labq(i),x(i),y(i),z(i),density(i),
     &            ref(i),qm(i),qpl(i),offset(i),atnox(i),cnx(i)
               end if
             end if
          end do
       close(11)

c--------------------------------------------------------------------------
c     normal end of code
c--------------------------------------------------------------------------

      goto 999

c--------------------------------------------------------------------------
c     here if problem reading master.prn
c--------------------------------------------------------------------------

  100 print *,'problem reading master.prn'

  999 continue
      end
```

MAKEVIEW source code

```
c--------------------------------------------------------------------------
c--------------------------------------------------------------------------
c                              MAKEVIEW
c                Copyright (C) 2019 by Benjamin P Hay
c
c     This program is free software: you can redistribute it and/or
c     modify it under the terms of the GNU General Public License as
c     published by the Free Software Foundation, either version 3 of the
c     License or (at your option) any later version.
c
c     This program is distributed in the hope that it will be useful,
c     but WITHOUT ANY WARRANTY; without even the implied warranty of
c     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See GNU
c     General Public License for more details at this address:
c     <http://www.gnu.org/licenses/>.
c
c     Benjamin P. Hay
c     Supramolecular Design Institute
c     127 Chestnut Hill Road
c     Oak Ridge, TN 37830
c
c     hayben@comcast.net
c
c--------------------------------------------------------------------------
c--------------------------------------------------------------------------
c     This program is distributed as part of the Supporting Information
c     for the Cryst. Growth Des. 2019 article entitled: Crystal
c     structure evidence for the directionality of lone pair-pi
c     interactions - fact or fiction? After reading an input file
c     giving atom labels and Cartesian coordinates, it writes a series
c     of input files to create POVRay images
c
c     The input file, which can be any of the files generated by
c     SUBSET or RANDOM,  must have the following format:
c     line 1: integer natom giving the number of atoms
c     line 2 - natom: first five variables in the line are a left-
c     justified two character atom label, x, y, z, density
c--------------------------------------------------------------------------
c--------------------------------------------------------------------------
```

```
      program makeview

      implicit none

      integer size
      parameter (size = 300000)

      integer i,j,k,natom,moreatom
      integer nbond,ibond(size),jbond(size)

      real x(size),y(size),z(size)
      real redval,greenval,blueval
      real rdens,gdens,bdens,density(size)
      real xmid,ymid,zmid,dmax,radius
      real dxji,dyji,dzji,dxjk,dyjk,dzjk
      real dist,dx,dy,dz
      real dotji,dotjk,dotik,cosa,angle(6,size)

      character*1  label,labeli,labelj,inkey
      character*2  atlab(size)
      character*20 input
      character*200 record

      logical flag

c--------------------------------------------------------------------------
c     get input file name
c--------------------------------------------------------------------------

    5 continue

      write(6,'(a)') 'Enter the name of the input file:'
      write(6,'(a)') ' '
      read(5,'(a20)') input

      flag = .false.
      inquire(file=input,exist=flag)
      if(.not.flag) then
         write(6,'(a,a)') 'Do not find file named ',input
         write(6,'(a)') 'Check file name and try again '
         write(6,'(a)') ' '
         goto 5
      end if

c--------------------------------------------------------------------------
c     define C6F5X group - atomic coordinates derived from Chem3D
c     optimized C6F6 molecule using MM2+ force field
c--------------------------------------------------------------------------

      natom = 12

      x(1) =  1.400e0
      y(1) =  0.000e0
      z(1) =  0.000e0
      atlab(1) = 'C '

      x(2) =  0.697e0
      y(2) = -1.200e0
      z(2) =  0.000e0
      atlab(2) = 'C '

      x(3) = -0.697e0
      y(3) = -1.200e0
      z(3) =  0.0000
      atlab(3) = 'C '

      x(4) = -1.400e0
      y(4) =  0.000e0
      z(4) =  0.000e0
      atlab(4) = 'C '
```

```
x(5) = -0.697e0
y(5) =  1.200e0
z(5) =  0.000e0
atlab(5) = 'C '

x(6) =  0.697e0
y(6) =  1.200e0
z(6) =  0.000e9
atlab(6) = 'C '

x(7) =  2.900e0
y(7) =  0.000e0
z(7) =  0.000e0
atlab(7) = 'C '

x(8) =  1.362e0
y(8) = -2.365e0
z(8) =  0.000e0
atlab(8) = 'F '

x(9) = -1.362e0
y(9) = -2.365e0
z(9) =  0.000e0
atlab(9) = 'F '

x(10) = -2.750e0
y(10) =  0.000e0
z(10) =  0.000e0
atlab(10) = 'F '

x(11) = -1.362e0
y(11) =  2.365e0
z(11) =  0.000e0
atlab(11) = 'F '

x(12) =  1.362e0
y(12) =  2.365e0
z(12) =  0.000e0
atlab(12) = 'F '

nbond = 12

ibond(1) = 1
jbond(1) = 2

ibond(2) = 1
jbond(2) = 6

ibond(3) = 1
jbond(3) = 7

ibond(4) = 2
jbond(4) = 3

ibond(5) = 2
jbond(5) = 8

ibond(6) = 3
jbond(6) = 4

ibond(7) = 3
jbond(7) = 9

ibond(8) = 4
jbond(8) = 5

ibond(9) = 4
jbond(9) = 10

ibond(10) = 5
```

```
      jbond(10) = 6

      ibond(11) = 5
      jbond(11) = 11

      ibond(12) = 6
      jbond(12) = 12

c-----------------------------------------------------------------------
c     read number of atoms then loop to read atom label, x, y, z
c-----------------------------------------------------------------------

      open (unit=7,file=input,status='old')
      read (7,'(a200)') record
      read (record,*,err=90,end=100) moreatom
      do i = natom+1, natom+moreatom
         read (7,'(a200)') record
         read (record,*,err=90,end=100) atlab(i),x(i),y(i),z(i),
     &   density(i)
         z(i) = -z(i)
      end do
      natom = natom + moreatom
      goto 100

   90 continue
      write(6,'(a)') 'Error reading input file '

  100 close (7)

c-----------------------------------------------------------------------
c     distribution with ball and stick model
c-----------------------------------------------------------------------

      open(unit=8,file='cloud_ball.pov',status='unknown')
         call header
         do i = 1, 12
           label = atlab(i)(1:1)
           write(8,401) x(i),y(i),z(i),label,label
         end do
         do i = 1, nbond
           write(8,402) x(ibond(i)),y(ibond(i)),z(ibond(i)),
     &     x(jbond(i)),y(jbond(i)),z(jbond(i))
         end do
         radius = 0.015e0
         do i = 13, natom
           if(atlab(i)(2:2).eq.' ') then
             label = atlab(i)(1:1)
             write(8,403) x(i),y(i),z(i),radius,label
           else
             write(8,404) x(i),y(i),z(i),radius,atlab(i)
           end if
         end do
         call footer
      close(8)

c-----------------------------------------------------------------------
c     distribution with tube model
c-----------------------------------------------------------------------

      open(unit=8,file='cloud_tube.pov',status='unknown')
         call header
         do i = 1, 12
           label = atlab(i)(1:1)
           write(8,406) x(i),y(i),z(i),label
         end do
         do i = 1, nbond
            xmid = (x(ibond(i)) + x(jbond(i)))/2.0e0
            ymid = (y(ibond(i)) + y(jbond(i)))/2.0e0
            zmid = (z(ibond(i)) + z(jbond(i)))/2.0e0
            labeli = atlab(ibond(i))(1:1)
            labelj = atlab(jbond(i))(1:1)
```

```
            write(8,407) x(ibond(i)),y(ibond(i)),z(ibond(i)),
     &      xmid, ymid, zmid, labeli
            write(8,407) x(jbond(i)),y(jbond(i)),z(jbond(i)),
     &      xmid, ymid, zmid, labelj
         end do
         radius = 0.015e0
         do i = 13, natom
            if(atlab(i)(2:2).eq.' ') then
               label = atlab(i)(1:1)
               write(8,403) x(i),y(i),z(i),radius,label
            else
               write(8,404) x(i),y(i),z(i),radius,atlab(i)
            end if
         end do
         call footer
      close(8)

c-------------------------------------------------------------------------
c     distribution with space-filling model
c-------------------------------------------------------------------------

      open(unit = 8, file = 'cloud_space.pov', status='unknown')
         call header
         do i = 1, 12
           label = atlab(i)(1:1)
           write(8,408) x(i),y(i),z(i),label,label
         end do
         radius = 0.015e0
         do i = 13, natom
           if(atlab(i)(2:2).eq.' ') then
             label = atlab(i)(1:1)
             write(8,403) x(i),y(i),z(i),radius,label
           else
             write(8,404) x(i),y(i),z(i),radius,atlab(i)
           end if
         end do
         call footer
      close(8)

c-------------------------------------------------------------------------
c     write angles - list of all lp - centroid - C angles
c       i is the contacting atom
c       j is the centroid (0,0,0)
c       k is the carbon atom
c-------------------------------------------------------------------------

      do i = 13, natom
         do j = 1, 6
            dxji = - x(i)
            dyji = - y(i)
            dzji = - z(i)
            dxjk = - x(j)
            dyjk = - y(j)
            dzjk = - z(j)
            dotji = dxji*dxji + dyji*dyji + dzji*dzji
            dotjk = dxjk*dxjk + dyjk*dyjk + dzjk*dzjk
            dotik = 1.0e0/sqrt(dotji*dotjk)
            cosa = (dxji*dxjk + dyji*dyjk + dzji*dzjk) * dotik
            cosa = max(-1.0e0,cosa)
            cosa = min(1.0e0,cosa)
            angle(j,i) = acos(cosa)*57.29578e0
         end do
      end do
      open(unit=8,file='angles',status='unknown')
         do i = 13, natom
            do j = 1, 6
               write(8,'(F10.4)') angle(j,i)
            end do
         end do
      close(8)
```

```
c-------------------------------------------------------------------------
c     set density variables and find maximum density
c-------------------------------------------------------------------------

      rdens = 0.60e0
      gdens = 0.50e0
      bdens = 0.40e0

      dmax = 0.00e0
      do i = 13, natom
         if(density(i).gt.dmax) then
            dmax = density(i)
         endif
      enddo

c-------------------------------------------------------------------------
c     density with ball and stick model
c-------------------------------------------------------------------------

      open(unit=8,file='dense_ball.pov',status='unknown')
         call header
         do i = 1, 12
           label = atlab(i)(1:1)
           write(8,401) x(i),y(i),z(i),label,label
         end do
         do i = 1, nbond
           write(8,402) x(ibond(i)),y(ibond(i)),z(ibond(i)),
     &     x(jbond(i)),y(jbond(i)),z(jbond(i))
         end do
         do i = 13, natom
            if((density(i)/dmax).gt.rdens) then
               redval = 1.0e0
               greenval = 0.0e0
               blueval = 0.0e0
               radius = 0.025e0
            elseif((density(i)/dmax).gt.gdens) then
               redval = 0.0e0
               greenval = 1.0e0
               blueval = 0.0e0
               radius = 0.025e0
            elseif((density(i)/dmax).gt.bdens) then
               redval = 0.0e0
               greenval = 0.0e0
               blueval = 1.0e0
               radius = 0.025e0
            else
               redval = 0.00+(density(i)/dmax)
               greenval = 0.00+(density(i)/dmax)
               blueval = 0.00+(density(i)/dmax)
               radius = 0.015e0
            end if
            write(8,'(a)') '#declare L_tex=texture{pigment{'
            write(8,'(a,x,F5.3)') 'colour'
            write(8,'(a,x,F5.3)') 'red',redval
            write(8,'(a,x,F5.3)') 'green',greenval
            write(8,'(a,x,F5.3)') 'blue',blueval
            write(8,'(a)') '} finish{ATM_FINISH}}'
            write(8,405) x(i),y(i),z(i),radius
         end do
         call footer
      close(8)

c-------------------------------------------------------------------------
c     density with tube model
c-------------------------------------------------------------------------

      open(unit=8,file='dense_tube.pov',status='unknown')
         call header
         do i = 1, 12
           label = atlab(i)(1:1)
           write(8,406) x(i),y(i),z(i),label
```

```
         end do
         do i = 1, nbond
            xmid = (x(ibond(i)) + x(jbond(i)))/2.0e0
            ymid = (y(ibond(i)) + y(jbond(i)))/2.0e0
            zmid = (z(ibond(i)) + z(jbond(i)))/2.0e0
            labeli = atlab(ibond(i))(1:1)
            labelj = atlab(jbond(i))(1:1)
            write(8,407) x(ibond(i)),y(ibond(i)),z(ibond(i)),
     &      xmid, ymid, zmid, labeli
            write(8,407) x(jbond(i)),y(jbond(i)),z(jbond(i)),
     &      xmid, ymid, zmid, labelj
         end do
         do i = 13, natom
            if((density(i)/dmax).gt.rdens) then
               redval = 1.0e0
               greenval = 0.0e0
               blueval = 0.0e0
               radius = 0.025e0
            elseif((density(i)/dmax).gt.gdens) then
               redval = 0.0e0
               greenval = 1.0e0
               blueval = 0.0e0
               radius = 0.025e0
            elseif((density(i)/dmax).gt.bdens) then
               redval = 0.0e0
               greenval = 0.0e0
               blueval = 1.0e0
               radius = 0.025e0
            else
               redval = 0.00+(density(i)/dmax)
               greenval = 0.00+(density(i)/dmax)
               blueval = 0.00+(density(i)/dmax)
               radius = 0.015e0
            end if
            write(8,'(a)') '#declare L_tex=texture{pigment{'
            write(8,'(a,x,F5.3)') 'colour'
            write(8,'(a,x,F5.3)') 'red',redval
            write(8,'(a,x,F5.3)') 'green',greenval
            write(8,'(a,x,F5.3)') 'blue',blueval
            write(8,'(a)') '} finish{ATM_FINISH}}'
            write(8,405) x(i), y(i), z(i), radius
         end do
         call footer
      close(8)

c--------------------------------------------------------------------------
c     density with space-filling model
c--------------------------------------------------------------------------

      open(unit=8,file='dense_space.pov',status='unknown')
         call header
         do i = 1, 12
           label = atlab(i)(1:1)
           write(8,408) x(i),y(i),z(i),label,label
         end do

         do i = 13, natom
            if((density(i)/dmax).gt.rdens) then
               redval = 1.0e0
               greenval = 0.0e0
               blueval = 0.0e0
               radius = 0.025e0
            elseif((density(i)/dmax).gt.gdens) then
               redval = 0.0e0
               greenval = 1.0e0
               blueval = 0.0e0
               radius = 0.025e0
            elseif((density(i)/dmax).gt.bdens) then
               redval = 0.0e0
               greenval = 0.0e0
               blueval = 1.0e0
```

```
               radius = 0.025e0
            else
               redval = 0.00+(density(i)/dmax)
               greenval = 0.00+(density(i)/dmax)
               blueval = 0.00+(density(i)/dmax)
               radius = 0.015e0
            end if
            write(8,'(a)') '#declare L_tex=texture{pigment{'
            write(8,'(a,x,F5.3)') 'colour'
            write(8,'(a,x,F5.3)') 'red',redval
            write(8,'(a,x,F5.3)') 'green',greenval
            write(8,'(a,x,F5.3)') 'blue',blueval
            write(8,'(a)') '} finish{ATM_FINISH}}'
            write(8,405) x(i), y(i), z(i), radius
         end do
         call footer
      close(8)

c---------------------------------------------------------------------
c     before generating cut-away side views of bullet-shaped search
c     volumes, move points into the +y domain
c---------------------------------------------------------------------

      do i = 13, natom
         y(i) = abs(y(i))
      enddo

c---------------------------------------------------------------------
c     illustation of actual data selection used in Angew. Chem. Int.
c     Ed. 2002, 41, 3389, that is, contacting all six ring C atoms
c     at distances less than the sum of vdw radii + 1.0 angstroms
c---------------------------------------------------------------------

      open(unit=8,file='actual_2002.pov',status='unknown')
         call header
         do i = 1, 12
           label = atlab(i)(1:1)
           write(8,401) x(i),y(i),z(i),label,label
         end do
         do i = 1, nbond
           write(8,402) x(ibond(i)),y(ibond(i)),z(ibond(i)),
     $     x(jbond(i)),y(jbond(i)),z(jbond(i))
         end do
         radius = 0.015e0
         do i = 13, natom
            if(atlab(i)(2:2).eq.' ') then
               label = atlab(i)(1:1)
               write(8,403) x(i),y(i),z(i),radius,label
            else
               write(8,404) x(i),y(i),z(i),radius,atlab(i)
            end if
         end do
         write(8,'(a)') '} //end of cloud '
         write(8,'(a)') '#declare bigone = '
         write(8,'(a)') 'intersection{ '
         write(8,'(a,a)') 'sphere{<1.400,0.000,0.000>, 4.17',
     &    ' texture{S_tex}}'
         write(8,'(a,a)') 'sphere{<0.697,-1.200,0.000>, 4.17',
     &    ' texture{S_tex}}'
         write(8,'(a,a)') 'sphere{<-0.697,-1.200,0.000>, 4.17',
     &    ' texture{S_tex}}'
         write(8,'(a,a)') 'sphere{<-1.400,0.000,0.000>, 4.17',
     &    ' texture{S_tex}}'
         write(8,'(a,a)') 'sphere{<-0.697,1.200,0.000>, 4.17',
     &    ' texture{S_tex}}'
         write(8,'(a,a)') 'sphere{< 0.697,1.200,0.000>, 4.17',
     &    ' texture{S_tex}}'
         write(8,'(a)') '} '
         write(8,'(a)') '#declare littleone = '
         write(8,'(a)') 'intersection{ '
         write(8,'(a,a)') 'sphere{<1.400,0.000,0.000>, 4.16',
```

```
     &    ' texture{S_tex}}'
          write(8,'(a,a)') 'sphere{<0.697,-1.200,0.000>, 4.16',
     &    ' texture{S_tex}}'
          write(8,'(a,a)') 'sphere{<-0.697,-1.200,0.000>, 4.16',
     &    ' texture{S_tex}}'
          write(8,'(a,a)') 'sphere{<-1.400,0.000,0.000>, 4.16',
     &    ' texture{S_tex}}'
          write(8,'(a,a)') 'sphere{<-0.697,1.200,0.000>, 4.16',
     &    ' texture{S_tex}}'
          write(8,'(a,a)') 'sphere{< 0.697,1.200,0.000>, 4.16',
     &    ' texture{S_tex}}'
          write(8,'(a)') '} '
          write(8,'(a)') '#declare diff= '
          write(8,'(a)') 'difference{ '
          write(8,'(a)') 'object{bigone} '
          write(8,'(a)') 'object{littleone} '
          write(8,'(a)') '} '
          write(8,'(a)') '#declare clip1 = '
          write(8,'(a)') 'difference{ '
          write(8,'(a)') 'object{diff} '
          write(8,'(a)') 'box{<5,5,5>,<-5,-5,0> texture{S_tex}}'
          write(8,'(a)') '} '
          write(8,'(a)') '#declare clip2 = '
          write(8,'(a)') 'difference{ '
          write(8,'(a)') 'object{clip1} '
          write(8,'(a)') 'box{<-5,-5,0>,<5,0,-5> texture{S_tex}}'
          write(8,'(a)') '} '
          write(8,'(a)') ' '
          write(8,'(a)') 'object{clip2 '
          write(8,'(a)') 'translate<0,0,0> '
          write(8,'(a)') 'rotate<90,0,0> '
          write(8,'(a)') '} '
          write(8,'(a)') ' '
          write(8,'(a)') 'object{cloud'
          write(8,'(a)') 'translate<0,0,0>'
          write(8,'(a)') 'rotate<90,0,0>'
          write(8,'(a)') '} '
          write(8,'(a)') ' '
          write(8,'(a)') '//END'
       close(8)

c--------------------------------------------------------------------------
c     illustation of selection criterion reported in Angew. Chem. Int.
c     Ed. 2002, 41, 3389, that is, contacting all six ring C atoms
c     at distances less than the sum of vdw radii
c--------------------------------------------------------------------------

       open(unit=8,file='2002.pov',status='unknown')
          call header
          do i = 1, 12
            label = atlab(i)(1:1)
            write(8,401) x(i),y(i),z(i),label,label
          end do
          do i = 1, nbond
            write(8,402) x(ibond(i)),y(ibond(i)),z(ibond(i)),
     $      x(jbond(i)),y(jbond(i)),z(jbond(i))
          end do
          radius = 0.015e0
          do i = 13, natom
             if(atlab(i)(2:2).eq.' ') then
                label = atlab(i)(1:1)
                write(8,403) x(i),y(i),z(i),radius,label
             else
                write(8,404) x(i),y(i),z(i),radius,atlab(i)
             end if
          end do
          write(8,'(a)') '} //end of cloud '
          write(8,'(a)') '#declare bigone = '
          write(8,'(a)') 'intersection{ '
          write(8,'(a,a)') 'sphere{<1.400,0.000,0.000>, 3.17',
     &    ' texture{S_tex}}'
```

```
          write(8,'(a,a)') 'sphere{<0.697,-1.200,0.000>, 3.17',
     &     ' texture{S_tex}}'
          write(8,'(a,a)') 'sphere{<-0.697,-1.200,0.000>, 3.17',
     &     ' texture{S_tex}}'
          write(8,'(a,a)') 'sphere{<-1.400,0.000,0.000>, 3.17',
     &     ' texture{S_tex}}'
          write(8,'(a,a)') 'sphere{<-0.697,1.200,0.000>, 3.17',
     &     ' texture{S_tex}}'
          write(8,'(a,a)') 'sphere{< 0.697,1.200,0.000>, 3.17',
     &     ' texture{S_tex}}'
          write(8,'(a)') '} '
          write(8,'(a)') '#declare littleone = '
          write(8,'(a)') 'intersection{ '
          write(8,'(a,a)') 'sphere{<1.400,0.000,0.000>, 3.16',
     &     ' texture{S_tex}}'
          write(8,'(a,a)') 'sphere{<0.697,-1.200,0.000>, 3.16',
     &     ' texture{S_tex}}'
          write(8,'(a,a)') 'sphere{<-0.697,-1.200,0.000>, 3.16',
     &     ' texture{S_tex}}'
          write(8,'(a,a)') 'sphere{<-1.400,0.000,0.000>, 3.16',
     &     ' texture{S_tex}}'
          write(8,'(a,a)') 'sphere{<-0.697,1.200,0.000>, 3.16',
     &     ' texture{S_tex}}'
          write(8,'(a,a)') 'sphere{< 0.697,1.200,0.000>, 3.16',
     &     ' texture{S_tex}}'
          write(8,'(a)') '} '
          write(8,'(a)') '#declare diff= '
          write(8,'(a)') 'difference{ '
          write(8,'(a)') 'object{bigone} '
          write(8,'(a)') 'object{littleone} '
          write(8,'(a)') '} '
          write(8,'(a)') '#declare clip1 = '
          write(8,'(a)') 'difference{ '
          write(8,'(a)') 'object{diff} '
          write(8,'(a)') 'box{<5,5,5>,<-5,-5,0> texture{S_tex}}'
          write(8,'(a)') '} '
          write(8,'(a)') '#declare clip2 = '
          write(8,'(a)') 'difference{ '
          write(8,'(a)') 'object{clip1} '
          write(8,'(a)') 'box{<-5,-5,0>,<5,0,-5> texture{S_tex}}'
          write(8,'(a)') '} '
          write(8,'(a)') ' '
          write(8,'(a)') 'object{clip2 '
          write(8,'(a)') 'translate<0,0,0> '
          write(8,'(a)') 'rotate<90,0,0> '
          write(8,'(a)') '} '
          write(8,'(a)') ' '
          write(8,'(a)') 'object{cloud'
          write(8,'(a)') 'translate<0,0,0>'
          write(8,'(a)') 'rotate<90,0,0>'
          write(8,'(a)') '} '
          write(8,'(a)') ' '
          write(8,'(a)') '//END'
       close(8)

c--------------------------------------------------------------------------
c     illustration of data selection used in CrystEngComm 2008, 10, 1501
c     in other words, within 4 Å of all six ring C atoms
c--------------------------------------------------------------------------

       open(unit = 8, file = '2008.pov', status='unknown')
          call header
          do i = 1, 12
            label = atlab(i)(1:1)
            write(8,401) x(i),y(i),z(i),label,label
          end do
          do i = 1, nbond
            write(8,402) x(ibond(i)),y(ibond(i)),z(ibond(i)),
     $      x(jbond(i)),y(jbond(i)),z(jbond(i))
          end do
          radius = 0.015e0
```

```
          do i = 13, natom
            if(atlab(i)(2:2).eq.' ') then
              label = atlab(i)(1:1)
              write(8,403) x(i),y(i),z(i),radius,label
            else
              write(8,404) x(i),y(i),z(i),radius,atlab(i)
            end if
          end do
          write(8,'(a)') '} //end of cloud '
          write(8,'(a)') ' '
          write(8,'(a)') '#declare bigone = '
          write(8,'(a)') 'intersection{ '
          write(8,'(a,a)') 'sphere{<1.400,0.000,0.000>, 4.01',
     &    ' texture{S_tex}}'
          write(8,'(a,a)') 'sphere{<0.697,-1.200,0.000>, 4.01',
     &    ' texture{S_tex}}'
          write(8,'(a,a)') 'sphere{<-0.697,-1.200,0.000>, 4.01',
     &    ' texture{S_tex}}'
          write(8,'(a,a)') 'sphere{<-1.400,0.000,0.000>, 4.01',
     &    ' texture{S_tex}}'
          write(8,'(a,a)') 'sphere{<-0.697,1.200,0.000>, 4.01',
     &    ' texture{S_tex}}'
          write(8,'(a,a)') 'sphere{< 0.697,1.200,0.000>, 4.01',
     &    ' texture{S_tex}}'
          write(8,'(a)') '} '
          write(8,'(a)') '#declare littleone = '
          write(8,'(a)') 'intersection{ '
          write(8,'(a,a)') 'sphere{<1.400,0.000,0.000>, 4.00',
     &    ' texture{S_tex}}'
          write(8,'(a,a)') 'sphere{<0.697,-1.200,0.000>, 4.00',
     &    ' texture{S_tex}}'
          write(8,'(a,a)') 'sphere{<-0.697,-1.200,0.000>, 4.00',
     &    ' texture{S_tex}}'
          write(8,'(a,a)') 'sphere{<-1.400,0.000,0.000>, 4.00',
     &    ' texture{S_tex}}'
          write(8,'(a,a)') 'sphere{<-0.697,1.200,0.000>, 4.00',
     &    ' texture{S_tex}}'
          write(8,'(a,a)') 'sphere{< 0.697,1.200,0.000>, 4.00',
     &    ' texture{S_tex}}'
          write(8,'(a)') '} '
          write(8,'(a)') '#declare diff= '
          write(8,'(a)') 'difference{ '
          write(8,'(a)') 'object{bigone} '
          write(8,'(a)') 'object{littleone} '
          write(8,'(a)') '} '
          write(8,'(a)') '#declare clip1 = '
          write(8,'(a)') 'difference{ '
          write(8,'(a)') 'object{diff} '
          write(8,'(a)') 'box{<5,5,5>,<-5,-5,0> texture{S_tex}}'
          write(8,'(a)') '} '
          write(8,'(a)') '#declare clip2 = '
          write(8,'(a)') 'difference{ '
          write(8,'(a)') 'object{clip1} '
          write(8,'(a)') 'box{<-5,-5,0>,<5,0,-5> texture{S_tex}}'
          write(8,'(a)') '} '
          write(8,'(a)') ' '
          write(8,'(a)') 'object{clip2 '
          write(8,'(a)') 'translate<0,0,0> '
          write(8,'(a)') 'rotate<90,0,0> '
          write(8,'(a)') '} '
          write(8,'(a)') ' '
          write(8,'(a)') 'object{cloud'
          write(8,'(a)') 'translate<0,0,0>'
          write(8,'(a)') 'rotate<90,0,0>'
          write(8,'(a)') '} '
          write(8,'(a)') ' '
          write(8,'(a)') '//END'
        close(8)

c-----------------------------------------------------------------------
c     FORMATS
```

```
c-------------------------------------------------------------------------

  401 format('sphere {<',F6.3,',',F6.3,',',F6.3,'>, r',a1,
     &       ' * ball_scale texture{',a1,'_tex}}')

  402 format('cylinder {<',F6.3,',',F6.3,',',F6.3,'>,<',F6.3,',',F6.3,
     &        ',',F6.3,'>,r_Bnd open texture{bond_tex}}')

  403 format('sphere {<',F6.3,',',F6.3,',',F6.3,'>, ', F6.3,
     &       '*rStar  texture{',a1,'_tex}}')

  404 format('sphere {<',F6.3,',',F6.3,',',F6.3,'>, ', F6.3,
     &       '*rStar  texture{',a2,'_tex}}')

  405 format('sphere {<',F6.3,',',F6.3,',',F6.3,'>, ',F6.3,
     &       ' texture{L_tex}}')

  406 format('sphere {<',F6.3,',',F6.3,',',F6.3,'>, r_Bnd ',
     &       '  texture{',a1,'_tex}}')

  407 format('cylinder {<',F6.3,',',F6.3,',',F6.3,'>,<',F6.3,',',F6.3,
     &        ',',F6.3,'>,r_Bnd open texture{',a1,'_tex}}')

  408 format('sphere {<',F6.3,',',F6.3,',',F6.3,'>, r',a1,
     &       '  texture{',a1,'_tex}}')

c-------------------------------------------------------------------------
c     end of main code
c-------------------------------------------------------------------------
      end

c-------------------------------------------------------------------------
c     top of POVRay file
c-------------------------------------------------------------------------
      subroutine header
      write(8,'(a)') '#version 3'
      write(8,'(a)') ' '
      write(8,'(a)') 'camera { '
      write(8,'(a)') 'orthographic '
      write(8,'(a)') 'location <0,0,-100> '
      write(8,'(a)') 'up <0,15,0> '
      write(8,'(a)') 'right <15, 0, 0> '
      write(8,'(a)') 'look_at <0, 0, 0> '
      write(8,'(a)') '} '
      write(8,'(a)') ' '
      write(8,'(a)') 'object{light_source{<5,100,-5> color rgb 1.0}}'
      write(8,'(a)') 'object{light_source{<-5,0,-100> color rgb 1.5}}'
      write(8,'(a)') ' '
      write(8,'(a)') 'background {color rgb<0,0,0>}'
      write(8,'(a)') ' '
      write(8,'(a)') '#declare ball_scale = 0.25;'
      write(8,'(a)') '#declare rStar = 1.00;'
      write(8,'(a)') '#declare r_Bnd = 0.150;'
      write(8,'(a)') '#declare ATM_FINISH = finish '//
     & '{specular 1 roughness 0.001}'
      write(8,'(a)') '#declare Metal = finish { metallic '//
     & 'ambient 0.2 diffuse 0.7 brilliance 6 reflection 0.25 '//
     & ' phong 0.75 phong_size 80 }'
      write(8,'(a)') '#declare bond_tex = texture { pigment '//
     & '{ colour red 0.650 green 0.490 blue 0.240 } '//
     & 'finish{ Metal }}'
      write(8,'(a)') ' '
      write(8,'(a)') '#declare rB = 2.00;'
      write(8,'(a)') '#declare B_tex = texture { pigment '//
     & '{ colour red 0.600 green 0.800 blue 0.196 } '//
     & 'finish{ ATM_FINISH }}'
      write(8,'(a)') ' '
      write(8,'(a)') '#declare rC = 1.70;'
      write(8,'(a)') '#declare C_tex = texture { pigment '//
     & '{ colour red 0.533 green 0.533 blue 0.533 } '//
     & 'finish{ ATM_FINISH }}'
```

```
      write(8,'(a)') ' '
      write(8,'(a)') '#declare rN = 1.55;'
      write(8,'(a)') '#declare N_tex = texture { pigment '//
     & '{ colour red 0.000 green 0.498 blue 1.000 } '//
     & 'finish{ ATM_FINISH }}'
      write(8,'(a)') ' '
      write(8,'(a)') '#declare rO = 1.52;'
      write(8,'(a)') '#declare O_tex = texture { pigment '//
     & '{ colour red 1.000 green 0.000 blue 0.000 } '//
     & 'finish{ ATM_FINISH }}'
      write(8,'(a)') ' '
      write(8,'(a)') '#declare rF = 1.47;'
      write(8,'(a)') '#declare F_tex = texture { pigment '//
     & '{ colour red 0.600 green 0.800 blue 0.196 } '//
     & 'finish{ ATM_FINISH }}'
      write(8,'(a)') ' '
      write(8,'(a)') '#declare rP = 1.80;'
      write(8,'(a)') '#declare P_tex = texture { pigment '//
     & '{ colour red 1.000 green 0.660 blue 0.090 } '//
     & 'finish{ ATM_FINISH }}'
      write(8,'(a)') ' '
      write(8,'(a)') '#declare rS = 1.80;'
      write(8,'(a)') '#declare S_tex = texture { pigment '//
     & '{ colour red 1.000 green 0.940 blue 0.000 } '//
     & 'finish{ ATM_FINISH }}'
      write(8,'(a)') ' '
      write(8,'(a)') '#declare rCl = 1.75;'
      write(8,'(a)') '#declare Cl_tex = texture { pigment '//
     & '{ colour red 0.000 green 0.500 blue 0.000 } '//
     & 'finish{ ATM_FINISH }}'
      write(8,'(a)') ' '
      write(8,'(a)') '#declare rBr = 1.85;'
      write(8,'(a)') '#declare Br_tex = texture { pigment '//
     & '{ colour red 0.557 green 0.137 blue 0.137 } '//
     & 'finish{ ATM_FINISH }}'
      write(8,'(a)') ' '
      write(8,'(a)') '#declare rI = 1.98;'
      write(8,'(a)') '#declare I_tex = texture { pigment '//
     & '{ colour red 0.600 green 0.196 blue 0.800 } '//
     & 'finish{ ATM_FINISH }}'
      write(8,'(a)') ' '
      write(8,'(a)') ' '
      write(8,'(a)') '#declare cloud ='
      write(8,'(a)') 'union{'
      return
      end
c-----------------------------------------------------------------------
c     bottom of POVRay file
c-----------------------------------------------------------------------
      subroutine footer
      write(8,'(a)') '} //end of cloud '
      write(8,'(a)') ' '
      write(8,'(a)') 'object{cloud'
      write(8,'(a)') 'translate<0,0,0>'
      write(8,'(a)') 'rotate<0,0,0>'
      write(8,'(a)') '} '
      write(8,'(a)') ' '
      write(8,'(a)') '//END'
      return
      end
```

RANDOM source code

```
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
c                              RANDOM
c               Copyright (C) 2019 by Benjamin P Hay
c
c     This program is free software: you can redistribute it and/or
```

```
c     modify it under the terms of the GNU General Public License as
c     published by the Free Software Foundation, either version 3 of the
c     License or (at your option) any later version.
c
c     This program is distributed in the hope that it will be useful,
c     but WITHOUT ANY WARRANTY; without even the implied warranty of
c     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See GNU
c     General Public License for more details at this address:
c     <http://www.gnu.org/licenses/>.
c
c     Benjamin P. Hay
c     Supramolecular Design Institute
c     127 Chestnut Hill Road
c     Oak Ridge, TN 37830
c
c     hayben@comcast.net
c
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
c     This program is distributed as part of the Supporting Information
c     for the Cryst. Growth Des. 2019 article entitled: Crystal
c     structure evidence for the directionality of lone pair-pi
c     interactions - fact or fiction? It generates random distributions
c     of F atoms about C6F6.
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------

      program random

      implicit none

      integer size
      parameter (size = 200000)

      integer i, j, k
      integer nmax,nrs,n12,n12v

      real vdwff,vdwfc,thick
      real xr(12),yr(12),zr(12)
      real dist2(12),dmax(12),dmin(12)
      real xtest,ytest,ztest,dist
      real xrs(size),yrs(size),zrs(size)
      real dcenrs(size),dplrs(size),doffrs(size)
      real x12(size),y12(size),z12(size)
      real dcen12(size),dpl12(size),doff12(size)
      real x12v(size),y12v(size),z12v(size)
      real dcen12v(size),dpl12v(size),doff12v(size)

      character*1 inkey

c-----------------------------------------------------------------------
c     define parameters
c-----------------------------------------------------------------------

      nrs = 0
      n12 = 0
      n12v = 0
      vdwff = 2.94e0
      vdwfc = 3.170e0
      thick = 0.00e0
      xtest = rand(8321)

c-----------------------------------------------------------------------
c     how many points?
c-----------------------------------------------------------------------

      write(6,'(a)') ' '
      write(6,'(a)') ' '
      write(6,'(a,i6)') 'Number of random points = ',size
      write(6,'(a)') ' '
      write(6,'(a)') 'use this value? (y/n):'
```

```
      write(6,'(a)') ' '
      read(5,'(a1)') inkey
      if(inkey.eq.'n') then
         write(6,'(a)') ' '
         write(6,'(a,i6)') 'do not exceed max array size of ',size
         write(6,'(a)') 'input new value: '
         write(6,'(a)') ' '
         read(5,*) nmax
      else
         nmax = size
      end if
      if(nmax.gt.size) then
         write(6,'(a)') 'input exceeds maximum array size'
         stop
      end if

c-----------------------------------------------------------------------
c     layer thickness
c-----------------------------------------------------------------------

      write(6,'(a)') ' '
      write(6,'(a)') ' '
      write(6,'(a,F8.4)') 'Layer thickness, sum vdw radii + ',thick
      write(6,'(a)') ' '
      write(6,'(a)') 'use this value? (y/n):'
      write(6,'(a)') ' '
      read(5,'(a1)') inkey
      if(inkey.eq.'n') then
         write(6,'(a)') ' '
         write(6,'(a)') 'input new value: '
         write(6,'(a)') ' '
         read(5,*) thick
      else
         thick = 0.0e0
      end if

c-----------------------------------------------------------------------
c     define C6F5X atom positions
c-----------------------------------------------------------------------

      xr(1) =  1.400e0
      yr(1) =  0.000e0
      zr(1) =  0.000e0

      xr(2) =  0.697e0
      yr(2) = -1.200e0
      zr(2) =  0.000e0

      xr(3) = -0.697e0
      yr(3) = -1.200e0
      zr(3) =  0.000e0

      xr(4) = -1.400e0
      yr(4) =  0.000e0
      zr(4) =  0.000e0

      xr(5) = -0.697e0
      yr(5) =  1.200e0
      zr(5) =  0.000e0

      xr(6) =  0.697e0
      yr(6) =  1.200e0
      zr(6) =  0.000e0

      xr(7) =  2.750e0
      yr(7) =  0.000e0
      zr(7) =  0.000e0

      xr(8) =  1.362e0
      yr(8) = -2.365e0
      zr(8) =  0.000e0
```

```
      xr(9) = -1.362e0
      yr(9) = -2.365e0
      zr(9) =  0.000e0

      xr(10) = -2.750e0
      yr(10) =  0.000e0
      zr(10) =  0.000e0

      xr(11) = -1.362e0
      yr(11) =  2.365e0
      zr(11) =  0.000e0

      xr(12) =  1.362e0
      yr(12) =  2.365e0
      zr(12) =  0.000e0

c-----------------------------------------------------------------------
c     assign max and min contact distances for each atom
c-----------------------------------------------------------------------

      do i = 1, 12
         if(i.le.6) then
            dmin(i) = (vdwfc - 0.20e0)*(vdwfc - 0.20e0)
            dmax(i) = (vdwfc+thick)*(vdwfc+thick)
         else
            dmin(i) = (vdwff - 0.20e0)*(vdwff - 0.20e0)
            dmax(i) = (vdwff+thick)*(vdwff+thick)
         end if
      end do

c-----------------------------------------------------------------------
c     top of loop to generate random points
c-----------------------------------------------------------------------

   10 continue
      xtest = 6.0e0*rand()
      ytest = 6.0e0*rand()
      ztest = 6.0e0*rand()
      dist = sqrt(xtest*xtest + ytest*ytest + ztest*ztest)

c-----------------------------------------------------------------------
c     skip if distance to any atom < dmin(i)
c-----------------------------------------------------------------------

      do i = 1, 12
         dist2(i) = (xtest-xr(i))*(xtest-xr(i)) +
     &           (ytest-yr(i))*(ytest-yr(i)) +
     &           (ztest-zr(i))*(ztest-zr(i))
         if(dist2(i).lt.dmin(i)) then
            goto 10
         end if
      end do

c-----------------------------------------------------------------------
c     store random_shell atom if distance to any atom < dmax(i)
c-----------------------------------------------------------------------

      if(nrs.lt.nmax) then
         if(dist2(1).le.dmax(1).or.dist2(2).le.dmax(2).or.
     &      dist2(3).le.dmax(3).or.dist2(4).le.dmax(4).or.
     &      dist2(5).le.dmax(5).or.dist2(6).le.dmax(6).or.
     &      dist2(7).le.dmax(7).or.dist2(8).le.dmax(8).or.
     &      dist2(9).le.dmax(9).or.dist2(10).le.dmax(10).or.
     &      dist2(11).le.dmax(11).or.dist2(12).le.dmax(12))
     &      then
            nrs = nrs + 1
            xrs(nrs) = xtest
            yrs(nrs) = ytest
            zrs(nrs) = ztest
            dcenrs(nrs) = dist
```

```
            dplrs(nrs) = ztest
            doffrs(nrs) = sqrt(dist*dist-ztest*ztest)
         end if
      end if

c--------------------------------------------------------------------------
c     store random_2012 atom if distance to centroid < 5.0 angstroms
c--------------------------------------------------------------------------

      if(n12.lt.nmax) then
         if(dist.lt.5.0e0) then
            n12 = n12 + 1
            x12(n12) = xtest
            y12(n12) = ytest
            z12(n12) = ztest
            dcen12(n12) = dist
            dpl12(n12) = ztest
            doff12(n12) = sqrt(dist*dist-ztest*ztest)
         end if
      end if

c--------------------------------------------------------------------------
c     store random_2012_vdw atom if distance to centroid < 5.0 angstroms
c     and if distance to any atom < dmax(i)
c--------------------------------------------------------------------------

      if(n12v.lt.nmax) then
         if(dist.lt.5.00e0) then
            if(dist2(1).le.dmax(1).or.dist2(2).le.dmax(2).or.
     &         dist2(3).le.dmax(3).or.dist2(4).le.dmax(4).or.
     &         dist2(5).le.dmax(5).or.dist2(6).le.dmax(6).or.
     &         dist2(7).le.dmax(7).or.dist2(8).le.dmax(8).or.
     &         dist2(9).le.dmax(9).or.dist2(10).le.dmax(10).or.
     &         dist2(11).le.dmax(11).or.dist2(12).le.dmax(12))
     &         then
               n12v = n12v + 1
               x12v(n12v) = xtest
               y12v(n12v) = ytest
               z12v(n12v) = ztest
               dcen12v(n12v) = dist
               dpl12v(n12v) = ztest
               doff12v(n12v) = sqrt(dist*dist-ztest*ztest)
            end if
         end if
      end if

c--------------------------------------------------------------------------
c     done if nrs, n12, and n12v are all eq nmax, otherwise go again
c--------------------------------------------------------------------------

      if((nrs.eq.nmax).and.(n12.eq.nmax).and.(n12v.eq.nmax)) then
        goto 100
      else
        goto 10
      end if

  100 continue

c--------------------------------------------------------------------------
c     reflect half of the points to generate full hemisphere of data
c--------------------------------------------------------------------------

      do i = nrs/2, nrs
         xrs(i) = -1.0e0 * xrs(i)
      end do
      do i = 1, nrs, 2
         yrs(i) = -1.0e0 * yrs(i)
      end do

      do i = n12/2, n12
         x12(i) = -1.0e0 * x12(i)
```

```
      end do
      do i = 1, n12, 2
         y12(i) = -1.0e0 * y12(i)
      end do

      do i = n12v/2, n12v
         x12v(i) = -1.0e0 * x12v(i)
      end do
      do i = 1, n12v, 2
         y12v(i) = -1.0e0 * y12v(i)
      end do

c-----------------------------------------------------------------------
c     write output files
c-----------------------------------------------------------------------

      open(unit=8,file='random_shell')
         write(8,'(I8)') nrs
         do i = 1, nrs
            write(8,200) xrs(i),yrs(i),zrs(i),1,dcenrs(i),
     &      dplrs(i),doffrs(i)
         end do
      close(8)

      open(unit=9,file='random_2012')
         write(9,'(I8)') n12
         do i = 1, n12
            write(9,200) x12(i),y12(i),z12(i),1,dcen12(i),
     &      dpl12(i),doff12(i)
         end do
      close(9)

      open(unit=10,file='random_2012_vdw')
         write(10,'(I8)') n12v
         do i = 1, n12v
            write(10,200) x12v(i),y12v(i),z12v(i),1,dcen12v(i),
     &      dpl12v(i),doff12v(i)
         end do
      close(10)

  200 FORMAT('F ',3F8.4,I8,3F8.4)

      stop
      end
```