CRYSTALS Workshop

Advanced uses of CRYSTALS

The CRYSTALS Database

&

Structure Manipulation

Richard Cooper, David Watkin, Simon Parsons, Anna Collins & Stefan Pantos.

Data kindly provided by Ann Chippindale, Andrew Cowley, Tony Linden, Simon Parsons & Sofia Pascu.

Introduction, derived parameters & molecular axes

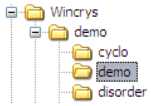
This example sets out to demonstrate the relationship between working from the GUI and working from the COMMAND LINE. It starts by introducing the database, crfilev2.dsc, and then looks at some derived parameter calculations.

All operations performed by CRYSTALS are in response to COMMANDS. These may be typed into CRYSTALS by the user, may be typed into file which is then executed, or may be created by the GUI and executed immediately. Most of the COMMANDS executed by CRYSTALS are also copied to the .LOG file.

Text that you must type into the command line verbatim is in **bold**.
Text that you must replace with your own values and type in is in ***bold italic***.

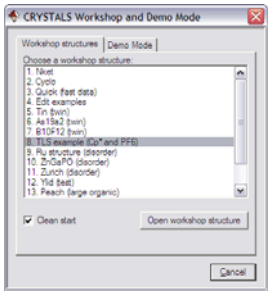
Starting the workshop initially

1. Double-click the CRYSTALS icon on the desktop
2. Choose the folder c:\wincrys\demo\demo
3. Click "OK".
4. From the list of workshops choose the one that you want. (E.g. number 8. TLS example)
5. Click "Open Workshop Structure".



Starting the workshop when CRYSTALS is running

1. From the **Help** menu, choose **Workshop/Demo**
2. In the dialog, click "Proceed".
3. From the list of workshops choose the one that you want. (E.g. number 8. TLS example)
4. Click "Open Workshop Structure".



The CRYSTALS database - crfilev2.dsc

Open the workshop structure entitled "8. TLS Example (Cp* and PF6)".

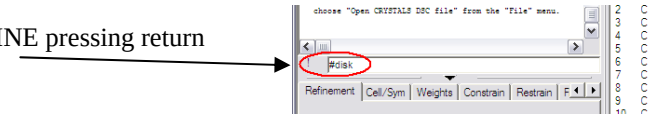
What information does the program know?

CRYSTALS organises all of its information in lists. These are stored in the working directory in a file called *crfilev2.dsc* (by default). Unfortunately these lists don't have very memorable names; here are some important ones to remember:

Cell parameters (List 1)

Type into the COMMAND LINE pressing return between each line:

#sum list 1
end

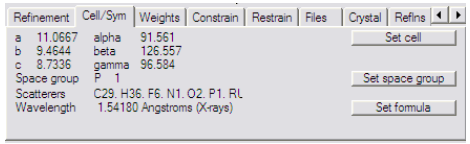


CRYSTALS prints a summary of the cell parameters (list 1). These could also be seen by clicking on the **Cell/Sym** tab, or edited by choosing **Edit Cell** from the **X-ray Data** menu.

Symmetry/Space group (List 2)

#sum l 2
end

prints a summary of the symmetry operators and space group. The space group can also be seen in the GUI by looking in the **Cell/Sym** tab, and can be changed by choosing **Edit space group** from the **X-ray Data** menu.



Scattering factors (List 3)

#sum l 3
end

lists the elements for which scattering factors have been input. These are also listed in the **Cell/Sym** tab, and can be changed by choosing **Edit formula / radiation** from the **X-ray Data** menu. This also sets 'List 29' which contains atomic properties.

Weights (List 4)

#sum l 4
end

shows the formula used to generate weights. This is also shown in the **Weights** tab and can be changed by choosing **Choose Weights** from the **Refinement** menu, (though see the manual for many more weighting schemes not available through this menu).

Atoms & other parameters (List 5)

#sum l 5
end

shows a quick summary of all the atoms. These are also displayed as a 3-D model on the right hand side of the screen, and in a list below it.
The parameter list may be edited manually (use **Edit Co-ordinates** from the **Structure** menu), or by using the **#EDIT** command (see the manual), or by right-clicking on atoms or groups of selected atoms in the user interface (either in the 3D model, or the list).

Reflections & reflection filter (Lists 6 & 28)s

#sum l 6
end

prints a summary of the stored reflections, including the current R-factors, the total number of reflections and the number accepted by 'list 28'. List 28 allows reflections to be omitted by many criteria (use **#sum l 28 / end** to view, and **Filter reflections** from the **X-ray Data** menu to edit this).

Constraints (List 12)

#sum l 12
end

prints the constraints (*aka* refinement directives) that tell the program which parameters to refine. They are also displayed in the **Constrain** tab. For simple cases list 12 can be setup automatically using **Setup and Refine** from the **Refinement** menu. For more complicated problems, they can be manually edited using **Edit directives and constraints** from the **Refinement** menu.

Restraints (List 16)

#sum l 16
end

lists any restraints which are to be imposed on the parameters. They are also displayed in the **Restrain** tab. Use **Edit Restraints** from the **Refinement** menu to change them. An empty list (no restraints) contains just the word 'NO'.

These are the most important lists to know about. To see which lists CRYSTALS has stored away for the current structure, type:

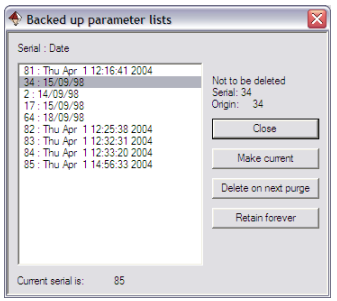
#disk
print disk
end

Note that LIST 5 (the parameters) serial numbers 2, 17, 34, & 58 are tagged 'not to be deleted'. When you are working, any time that you have a 'valuable' set of model parameters, you can tag them by typing:

#disk
retain 5
end

Note that most lists (except 5) overwrite old ones when you change the data. If you want to prevent this for a particular list, say the definition of a complicated refinement (List 12), retaining that list will protect it.

#disk
retain 12
end



Choosing **Undo/Backup Model** from the **Structure** menu provides an interface for safeguarding and for resetting to old parameter lists.

After you have been working for a while, the crfilev2.dsc file will fill up with old models (List 5). These can be eliminated by typing

#purge list=5
end

If you make a mess of a model, you can restore an earlier retained version by typing

#disk
reset 5 *n*
end

where ***n*** is the serial number of the LIST 5 you want to recover (or use the **Undo/Backup Model** option described above).

Derived Parameters via the GUI

The GUI provides instant access to some local geometry calculations:

- Hover the cursor over a bond to see its length and type.
- Right click on an atom and select **Environment** to list the local geometry in the text pane.
- Right click an atom and choose **Select Fragment Containing** to highlight a whole fragment and right click again to choose **TLS** analysis. A summary of this calculation appears in the text pane, and the full details go to the listing file. To see this file, choose the **Files** tab under the text output pane, and then click **View Listing**. Note the effect of libration on bond lengths.

To disconnect part of a structure from the rest (so that it becomes an isolated residue) you can right click on each bond, select **Change Type of Bond**, then **Break Bond**. This can be tedious for highly coordinated atoms, but see below.

Derived Parameters via the Command Line

In this section, we will classify the structure into four separate residues, and then carry out some calculations on them.

First, break the structure into four residues.

On the **Structure** drop-down menu choose
Make and break bonds

This will open a text editor showing the current state of the bonding modifiers.

- If you broke any bonds earlier, restore them by deleting the appropriate lines beginning **BREAK.**
- To break all the bonds around a hetero-atom set its bonding radius to zero, by adding the line in bold below:

#BONDING
DEFAULTS TOLTYPE= 1 TOLERANCE= 1.210 MAXBONDS = 12
ELEMENT *Ru* RADIUS = 0.0 MAXBONDS = 0
END

- Close and save the file. It will automatically be executed.

To see a table of all the covalent bonds, type:

#distance
end

There are many qualifiers to this command (see the manual), but as an example, type:

#dist
exclude h
end

Including the line

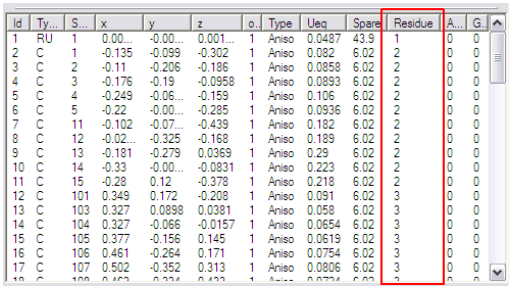
e.s.d yes

causes the standard uncertainties to be computed, but only if the atom list has not been modified since the last cycle of refinement (to be sure that the covariance matrix really applies to the atom list).

Each group of bonded atoms is called a *residue*. Disconnecting the Ru from the other atoms in this example produces a structure consisting of 4 residues (the Cp*, the steroid, the metal and the PF_6). CRYSTALS contains a powerful atom editor (see manual for details) one function of which is to associate a residue number with each residue

#edit
insert residue
end

(see also the menu **Structure**)



Id	Ty...	S...	x	y	z	o.	Type	Ueq	Spare	Residue	A...	G.
1	RU	1	0.00...	-0.00...	0.001...	1	Aniso	0.0487	43.9	1	0	0
2	C	1	-0.135	-0.099	-0.302	1	Aniso	0.082	6.02	2	0	0
3	C	2	-0.11	-0.206	-0.186	1	Aniso	0.0858	6.02	2	0	0
4	C	3	-0.176	-0.19	-0.0958	1	Aniso	0.0893	6.02	2	0	0
5	C	4	-0.249	-0.06...	-0.159	1	Aniso	0.106	6.02	2	0	0
6	C	5	-0.22	-0.00...	-0.285	1	Aniso	0.0936	6.02	2	0	0
7	C	11	-0.102	-0.07...	-0.439	1	Aniso	0.182	6.02	2	0	0
8	C	12	-0.02...	-0.325	-0.168	1	Aniso	0.189	6.02	2	0	0
9	C	13	-0.181	-0.279	0.0369	1	Aniso	0.29	6.02	2	0	0
10	C	14	-0.33	-0.00...	-0.0831	1	Aniso	0.223	6.02	2	0	0
11	C	15	-0.28	0.12	-0.378	1	Aniso	0.218	6.02	2	0	0
12	C	101	0.349	0.172	-0.208	1	Aniso	0.091	6.02	3	0	0
13	C	103	0.327	0.0898	0.0381	1	Aniso	0.058	6.02	3	0	0
14	C	104	0.327	-0.066	-0.0157	1	Aniso	0.0654	6.02	3	0	0
15	C	105	0.377	-0.156	0.145	1	Aniso	0.0619	6.02	3	0	0
16	C	106	0.461	-0.264	0.171	1	Aniso	0.0754	6.02	3	0	0
17	C	107	0.502	-0.352	0.313	1	Aniso	0.0806	6.02	3	0	0

Even if the Ru is reconnected to the other atoms (using EDBONDS as above) the residues will remain defined. The table below the graphics pane shows which residue each atom belongs to. (Note that right clicking on an atom in the table enables you to edit its parameters, including the residue number.)

Instead of typing individual atom names on the command line, all of the atoms in each residue can be referred to in subsequent calculations with the single identifier:

RESIDUE(n)

Where n is the number of the residue.

Molecular Axes

On the command line and with the help of the mouse to select atoms from the model, type:

#geometry
atom ***select a few bonded atoms by clicking the model***
plane
execute

(Note that the last command is **execute** not **end).**

This causes the best plane to be computed through the selected atoms and the resulting plane is given a sequence number starting from 1.
Continue the computation by typing the following (substituting your own values for the residue number and the calculation sequence numbers. Choose the residue corresponding to the CP*):

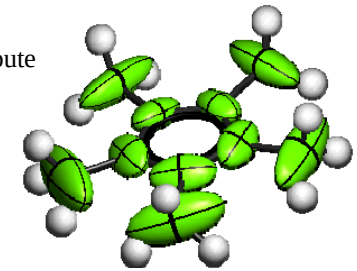
atom residue*(2)*
plane
execute

The results of this plane calculation are assigned the next available sequence number (in this case, 2). Compute the dihedral angle between the normals of the two planes that you have just defined, using their sequence numbers to identify them:

dihedral *1* and *2*
execute

Now do a TLS calculation on the same residue and compute the dihedral angle between the best plane normal and the axis of maximal libration

atom residue(2)
tls
execute
dihedral 2 and 3
execute



To get information about individual atoms or groups of atoms, type

atom residue(2) ru(1)
axes
execute
end

This lists the principal axes of the adps of the atoms. Note that C(11) to C(15) are labelled as potentially being split. The TLS and PLANE calculations showed that they can quite reasonably be regarded as a rigid planar body librating about its principal axis on inertia.

Finally, do a TLS calculation on the PF_6 residue.

#geom
atom residue(4)
tls
execute

Note that the smallest libration (L11) is negative. Since the units are degrees^2, this is clearly an artefact. This negative value can be replaced by a more reasonable estimate, and the new TLS tensors be used to 'regularise' the individual atomic apds. If residue 4 corresponds to the PF_6, type (where 50 is an estimate of L11, and 570 & 750 are roughly L22 and L33 and the zeros are the off-diagonal terms):

modl 50 570 750 0 0 0
execute
replace residue(4)
execute

Verify that the new libration is as expected:

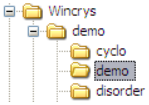
atoms residue(4)
tls
execute
end

Notice that the ellipsoids in the PF_6 now conform to a systematic pattern. This can be a useful before adding restraint to a poor anisotropic model. You can even add atoms to the **replace** line which were not in the original residue used to compute the TLS tensor, and their adps will be adjusted to conform to the rigid body motion.

Z'>1: Structure matching and numbering

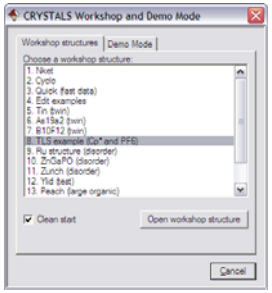
The next demonstration is a structure with 4 molecules in the asymmetric unit.

Starting the workshop initially



6. Double-click the CRYSTALS icon on the desktop
7. Choose the folder c:\wincrys\demo\demo
8. Click "OK".
9. From the list of workshops choose "4. Edit examples"
10. Click "Open Workshop Structure".

Starting the workshop when CRYSTALS is running



5. From the **Help** menu, choose **Workshop/Demo**
6. In the dialog, click "Proceed".
7. From the list of workshops choose "4. Edit examples"
8. Click "Open Workshop Structure".

A Z'=4 structure

Type:

#use work3.dat

The original coordinate list contains all the atoms or peaks in a unique half of the cell. CRYSTALS can assemble the atoms into molecules:

On the **Structure** menu, select **Collect atoms by Symmetry** to collect all the atoms together.

When there is only one molecule in the cell, this is often all that is necessary. For Z'>1 structures, or extended lattice structures, a more hands-on approach is required. This can be done through Cameron.

Moving molecules around

The idea here is to move the four individual molecules so that they can be seen and worked with more easily. We will:

- colour the four molecules differently
- use the space group symmetry to pack the structure
- choose one molecule of each colour to form our new model, such that the molecules are well spaced out and not all on top of each other.

Choose **Cameron Graphics** from the **Graphics** menu.
Type at the Cameron command line:

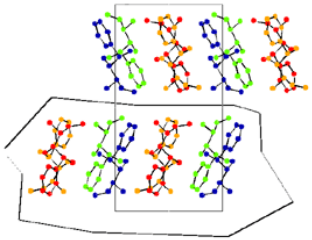
COLOUR FRAG <*click an atom*> RED VIEW

(<*click an atom*> from one molecule with the mouse)

Repeat the command above selecting an atom from each of the other molecules and a different colour. Permitted colour names are:

BLACK BLUE CYAN GREEN GREY LBLUE LGREEN LGREY
LRED MAGENTA ORANGE PINK PURPLE RED WHITE YELLOW

You should now have 4 differently coloured molecules.



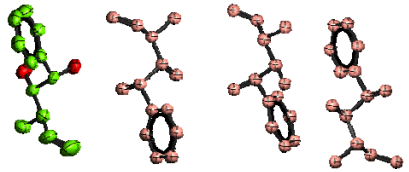
Change the pull-down listbox on the right, which currently says "**Unpack**", to "**Complete**".
This will create a mass of related molecules.
Use the **AXIS a/b/c** buttons to get different views of the structure. The view along the c-axis separates the molecules into layers.

Click the **Incl Area** button, then draw a polygon to enclose one layer of molecules
Click **+90** about the *x* axis to look onto the layer.
Identify four molecules (one of each colour) that you wish to keep. Then type:

Exclude all Include frag *aa bb cc dd* View

(where ***aa*** etc. are single atoms selected from each fragment).

The EXCLUDE and INCLUDE *must* be on the same line.

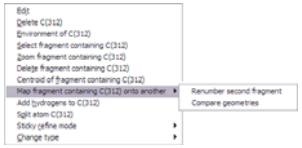


Cameron can now be closed (**File->Exit Cameron**) and the altered structure read back into CRYSTALS (click **Yes**). It mightlook something like the image on the right. In case of mistakes, start again with **#use work3.dat**.

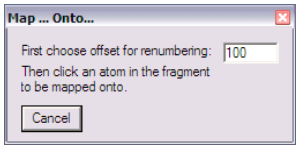
Re-naming the Q peaks to real atoms

The numbering scheme for the molecule with assigned element types has been carefully worked out. We now want to systematically apply it to the other molecules, with a serial number offset so that for example, the methyl carbon is C(4), C(104), C(204) and C(304) in each molecule.

1. Right-click on any atom in the correctly numbered molecule.
2. From the popup menu choose **Map fragment containing *C*(*n*) onto another -> Renumber second fragment**.



3. Enter a serial offset (e.g. *100*).



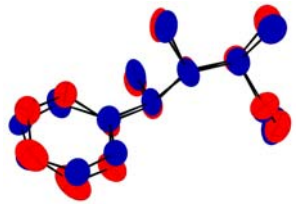
4. Click one atom in the molecule that is to be renumbered.

Provided that your molecule contains no internal non-crystallographic symmetry, the second molecule will be assigned the same elements and an offset serial number from the first.

Repeat for the remaining two unassigned molecules, but use different offsets, e.g. 200 and 300.

Comparing molecular geometry

The **Map fragment containing *C*(*n*) onto another -> Compare geometries** popup menu option displays the two fragments overlapped using a least squares fit. It gives a qualitative idea of the differences between the two molecules. (Quantitative ideas are in the listing file - see **View Listing** in the **Files** tab.)



To colour the fragments as shown: In Cameron, type:

```
OBEY regular.oby
```

Don't pack the structure, and **don't** save the changes after exiting Cameron. The overlapped structure is in an orthogonal unit cell defined by the best plane system of the molecule.

Same restraints

The **Map fragment containing *C*(*n*) onto another -> Same restraint** menu option uses the mapping of one molecule onto another to automatically generate a 'SAME' restraint (in List 16). This restrains equivalent 1,2 and 1,3 distances in each molecule to be the same.

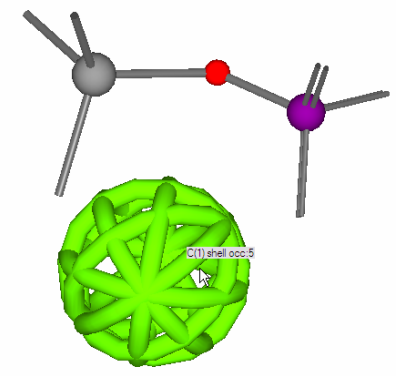
Use the Restraints tab to see the generated restraints. Type #CHECK <CR/LF> ENDto see the details.

Special shapes & squeeze

This section introduces some of the features of CRYSTALS for dealing with non-atomic electron density distributions.

GaPO with a Disorderly Guest

This example structure (courtesy of Andrew Cowley and Ann Chippindale) is a material containing a highly disordered cyclo-pentane molecule *on a special position*. The special position has 12-fold symmetry, being at the intersection of the 2-fold and 3-fold axes in the cubic space group **P -4 3 n**.
It could be modelled very tediously with several partially occupied cyclopentane molecules. Two alternative strategies are



- include electron density distributed around a spherical shell.
- remove the guest and use the difference Fourier transform of the resulting void to correct the structure factors.

From the workshop dialog, choose “10. ZnGaPO (disorder)”.

Strategy 1: Distributed density

Andrew has prepared all the data as a single *command file*. To read in the structure and the data type:

#use zngapo.dat

Note that:

1. the ‘shell’ is represented graphically with holes in – that is so you can see through to the inside, there aren’t really gaps, it is a smooth shell of electron density.
2. the Ga site should really be a disordered Ga and Zn, but this makes little difference to this refinement.

Visualise the whole structure, by running **Cameron** and choosing the **Complete** packing option. Unfortunately, Cameron cannot display the spherical shell, but you do see an atom C1 at its centre.

View down the *a*, *b* and *c* axes.
The disordered cyclopentane, represented here by C1, is at the centre of the framework’s cavity. Close Cameron (**File->Exit Cameron**)

The shell is centred on a special position, so we will not refine its co-ordinates, but we can refine its size and the isotropic displacement (think of this as the 'thickness' of the shell). Type

#sum l 12
end

to see the refinement directives which have already been set up.

To carry out some refinement, type:

#sfls
refine
refine
end

(FYI: SFLS stands for Structure Factor Least Squares)

To show that the shell really is in a fairly stable minimum, you might like to try shaking and re-refining the structure:

#perturb
end
#sfls
refine
refine
end

NB: Sometimes the size parameter of the shell may be perturbed *too far*, and the structure will not recover - don't panic - start again with **#use zngapo.dat.**

Note the R-factors in the **Refinement** tab - they should be R=3.56 Rw=4.14

Comparison: No guest at all

Right-click on the shell and choose **Delete C(1)** from the popup menu..

The refinement directives are now invalid as they refer to C(1), so set up some new ones by choosing **Setup and Refine** from the **Refinement** menu.
In the following dialog ensure that "Refine Scale, positions and Uij's" is selected and then click **OK.**

Note the R-factors in the Refinement tab - they will be R=3.99 and Rw=5.12.

A difference Fourier will show up the cyclopentane electron density:

```
#fourier
map type=diff
end
#peaks
end
```

The single peak shows up at coordinates displaced about 1.2Å from the special position at (0,0,0) as we would expect.

Delete the Fourier peak by right-clicking it and choosing **Delete Q(1)** from the popup menu.

Strategy 2: Squeeze

The structure factor is a complex number (has both magnitude and phase). The magnitude can be represented by $F^2=A^2+B^2$, where A is the real and B the imaginary part.
Each part can be thought of either as the integral over all the electron density in the crystal:

$$A_{hkl} = \int_v \rho_{xyz} \cos 2\pi(hx+ky+lz)\partial v$$

or as the sum of the scattering from a model made of atoms:

$$A_{hkl} = \sum_j f_j \cos 2\pi(hx+ky+lz)$$

with *B* given by analogous sin terms.

The traditional method of running Ton Spek's Squeeze computes the contribution to the structure factor from the difference electron density in 'void' areas of the crystal structure. That is, $F^2_{squeezed} = F^2_{observed} - F^2_{void}$. This method loses the *phase* information from the scattering void and also requires a correction to F_{obs} which crystallographers will instinctively find distasteful. However, PLATON also outputs the individual A and B parts of the scattering from the difference density in the void. This means that the structure factor can be thought of as a hybrid expression which includes the phase of the scattering from the void density. $F^2=(A_{atoms}+A_{void})^2 + (B_{atoms}+B_{void})^2$

$$A_{hkl} = \sum_j f_j \cos 2\pi(hx+ky+lz) + \int_v \rho_{xyz} \cos 2\pi(hx+ky+lz)\partial v$$

thus the electron density of the void effectively becomes part of the model and is added into the structure factor during all calculations. Another advantage of using this method is that the scattering from the void can be recomputed at any point (as the original F_{obs} have not been tampered with), and this leads to the possibility of improving the electron density in the void as the model improves.

From the **Refinement** menu, choose **PLATON Squeeze.**

When PLATON has finished running (it may take a while), close down the output window.

CRYSTALS will ask "Do you want to apply the results?" - Click **Yes**.

A command window will open summarising the number of reflections passed into PLATON. It always reports that PLATON returned one more reflection than it got passed - this is just a minor bug. Hit return to continue.

Now try some refinement:

```
#sfls
refine
refine
refine
end
```

The final 'squeezed' structure should have R=3.17 and Rw=3.87, so for cosmetic purposes, squeeze is the most rewarding strategy. However, in cases such as this it is probably better (*i.e. more correct*) to model the known solvent either from many partially occupied fragments or from distributed non-atomic electron density. A more thorough treatment would include an additional shell of electron density for the H atoms in the cyclopentane.

Disorder and Part numbers

Background to part numbers

Part numbers may be assigned to atoms to enable quick and easy handling of disorder in complicated structures.

The CRYSTALS part number combines two numbers which are used in CIF files to group and identify disorder.

Firstly, all atoms in a disordered region of a molecule are assigned the same ASSEMBLY number. For example if a CF_3 group is spinning around its axis, and can be better modelled by six partially occupied fluorine atoms, then *all* the fluorine atoms will be assigned assembly number 1.

Secondly, the set of atoms which are simultaneously occupied are given the same GROUP number. For example in one CF_3 orientation all the fluorine atoms will be in group 1, and in the other orientation all the atoms will be in group 2.

CRYSTALS combines these numbers into a single PART number:

PART = 1000*ASSEMBLY + GROUP

Thus, in the example above we have three fluorines in part 1001 and three in part 1002.

These may be referred to in CRYSTALS commands using the same syntax as atoms. *E.g.*

```
#edit
delete part(1002)
end

#list 12
full part(1001,occ) part(1002,occ)
end

#edit
reset occ 0.5 part(1001)
end
```

$C_{16}H_{22}OS$: Disorder

Import data and solve structure

From the workshop dialog, choose “11. Zurich (disorder)”.

From the **X-Ray Data** menu choose **Pre-process > Run Kccdin – Kccd data**.
A command prompt dialog will open.

Enter **2** to choose a monoclinic crystal system.
Type in the space group with spaces between the operators, *e.g.* **P 21/c**.
The value of Z is OK just press return.
Enter the crystal dimensions, (.1 x .2 x .2) each length separated by a space.
Enter the temperature (210).
Enter the colour (colourless).
The command prompt will close automatically.

In the “Input KCCD data” dialog, click **OK** to import the data.
Choose refinement against **F-squared**.
In the “Merge equivalent reflections” dialog, click **Yes**.
Click **OK** in the “Merge” dialog.
When the “Filter reflections” dialog opens, click **OK**.

From the **Solve** menu, choose **Sir92**. In the “Sir92” dialog, click **OK**. Sir will successfully solve the structure. Click the **Quit** button at the top left of the window twice to close.

When CRYSTALS asks “Do you want to use the structure from Sir92?” choose **Yes**.

The structure is correct, so click **Done** and choose **Automatic** numbering.

Refinement

From the **Refinement** menu, choose **Setup and Refine**. Make sure “Refine scale, positions and Uiso” is selected and click **OK**. After a few cycles of refinement, it will become apparent that something is up with one of the carbon atoms. **Setup and Refine** again, but select “Refine scale, positions and Uijs” to refine the atoms anisotropically. The adp of that carbon atom will become elongated, indicating that it is disordered.

Splitting the atom

The first step in modelling the disorder is to split the disordered atom, C(15). Right-click on it and choose **Split atom C(15)** from the popup menu. The original atom now needs to be deleted. Right-click on it again, and choose **Delete C(15)**. You should now be able to see two isotropic carbon atoms, which will have serial numbers C(150) and C(151). From the atomic parameters list below the model, you can see that each has as occupancy of 0.5.

From here, it would be relatively straight-forward to edit the constraints (List 12) and refine the disordered structure. However, problems will arise when hydrogen atoms are added. By using *parts*, it is possible to simplify the constraints to overcome these problems. Most of this will be done using the command line.

Adding hydrogens

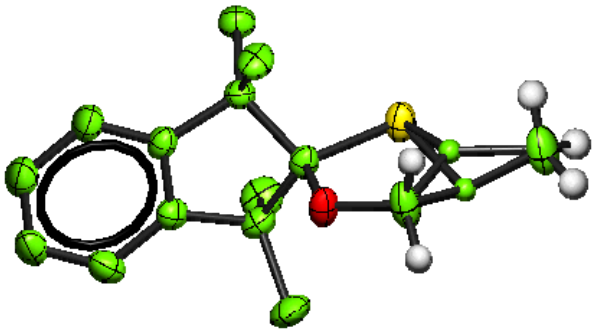
Both of the carbon atoms next to the disordered atom will need to have their hydrogens attached "by hand". This can be done by defining the type of carbon atom (in both cases sp^3) and the number of hydrogens attached to it.

In the command line type:

#hydrogens
serial 101 [101 is the serial number of the first H atom that will be added]
h33 c(18) c(150) c(16) [3 sp^3 H]
h23 c(16) c(150) o(17) [2 sp^3 H]
end

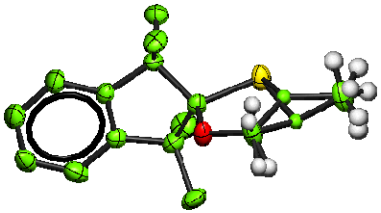
Then make these hydrogen atoms and C(150) into group 1 of assembly 1, *i.e.* part 1001. Type:

#edit
reset part 1001 c(150) h(101) until last [1001 is the part number]
end



Repeat this process for C(151):

#hydrogens
serial 201
h33 c(18) c(151) c(16)
h23 c(16) c(151) o(17)
end



Put these atoms into group 2 of assembly 1, *i.e.* part 1002:

#edit
reset part 1002 c(151) h(201) until last
end

However, all the hydrogen atoms have an occupancy of 1. Now that we have defined parts, this is not much of a problem. Start by changing all occupancies of both part 1001 and 1002 to 0.5:

#edit
reset occ 0.5 part(1001) part(1002)
end

To keep the total occupancy at each atom site as 1 during refinement we need to set up some restraints (list 12). Type:

#script edlist12
end

A text editor will open. Much of the text is commented out (lines starting with "#"). Insert a new line above "END" containing:

SUMFIX part(1001,occ) AND part(1002,occ)

Save and close the window.

Finally, add the rest of the Hydrogen atoms by typing:

#perhydro
end

Now the structure is ready to be refined. In the **Refinement** menu, choose **Refine**. (If you choose "Setup and refine" a dialog will open warning you that continuing will overwrite any refinement instructions that you may have written". You don't want to do this.) The structure should refine successfully, with each atom in a given part having the same occupancy, and both sets of occupancies adding up to unity. The final R-factor will be just under 10%.

Twinning & ROTAX

This section introduces some of the features of CRYSTALS for analyzing and dealing with twinned crystal structures.

The material includes one merohedral twin (twin law is part of the higher symmetry Laue class of the tetragonal system), one pseudo-merohedral twin (twinning occurs through a higher symmetry supercell), and one non-merohedral twins.

Data handling background

Using the user-interface tools makes handling twinning quite simple - however if you like to understand what is going on, here are the details:

```
H K L  /FO/  /FOT/ ELEMENTS
1 2 6  12.4  18.7    1234
```

CRYSTALS does not use HKLF5 format for twinned data. Instead it uses a more compact format, storing each reflection with just two additional pieces of data: Firstly the usual value of /Fo/ is called /FoT/ instead – that is, the observed intensity when the scattering from all twin components is added. /Fo/ now holds the scattering from the principal twin component only, and is used, for example, for Fourier maps.
Secondly, an element 'tag' is added to the data, this specifies which twin components contribute to the reflection, and is of the form "12" to represent contributions from elements 1 and 2. (The maximum is 9 twin components)

For *normal* data every tag will contain at least a "1", since this will be the lattice which the data was indexed on.

The given *hkl* indices of the reflection always correspond to the indices of component 1.

In addition to the extra reflection data, CRYSTALS also needs a list of twin laws (held internally as 'LIST 25'). The first twin law corresponds to component "1" and is almost always a unit matrix.

```
#LIST 25
READ NELEMENT = 2
MATRIX 1  0 0   0 1 0   0 0 1
MATRIX 0 -1 0  -1 0 0   0 0 1
END
```

The twin laws can be seen in the **Twin** tab of the CRYSTALS window.
The other twin laws correspond to components "2"-"9". These matrices are used with the hkl indices of component "1" to compute the indices of the overlapping reflections from other twin components.

The 'twin element scales' (*i.e.* the relative volume of each twin component in the crystal) need to be added to the list of parameters using the ELEMENTS directive. *E.g.* for a two component twin, starting from a 50:50 mixture of twin components:

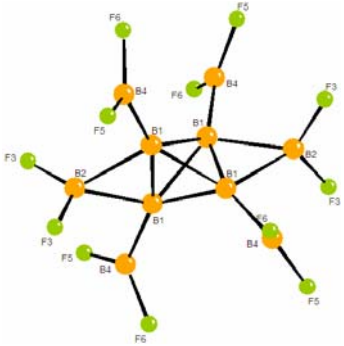
```
#LIST 5
READ     NATOM=42,   NELEMENTS =  2
OVERALL    1.0 0.5  0.5  1.0  0.0 1.0
ELEMENTS 0.5 0.5
ATOM O      1.000000   1.000000   1.000000   0.779320   0.474030   0.000000
CON U[11]=   0.050000   0.000000   0.000000   0.000000   0.000000   0.000000
CON SPARE=      8.05        0        0        1      O1
etc...
END
```

Finally, the sum of these elements must be constrained to be fixed, otherwise the normal matrix will become singular when they are included with the overall scale factor:

```
#LIST 12
FULL X'S U'S
CONTINUE   etc...
SUMFIX ELEMENT SCALES
END
```

ALL OF THESE CHANGES ARE MADE AUTOMATICALLY BY THE SCRIPTS WHICH YOU WILL BE USING IN THE FOLLOWING SECTION.

$B_{10}F_{12}$: Merohedral Twin: Summary information



This data for this crystal structure was collected in Edinburgh.

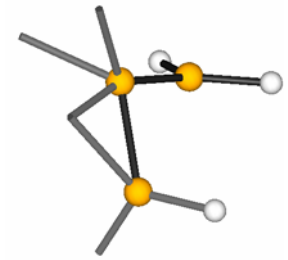
The cell is tetragonal with a=b=6.412, c=27.551.

The space group is **I 4_1/a.**

Import data, and solve structure

From the workshop dialog, choose "7. B10F12 (Twin)".
Choose **Import Shelx file (INS or RES)** from the **X-ray Data** menu.
Click **Browse**.
Choose the file *hibflq.ins* or *hibflq* (depending how your PC is set up). Click **Open**.
Verify that the space group is correct, then click **OK**.
When the "Input reflection data" dialog appears, click **OK.**
When the "Merge" dialog appears, click **OK.**
Choose refinement against **F-squared.**
When the "Filter reflections" dialog appears, click **OK.**

Now, from the **Solve** menu, choose **Sir92**. In the "Sir 92" dialog, click **OK.**
Sir will finish. Click the **Quit** button at the top left of the window twice.



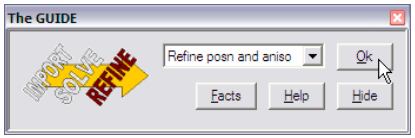
In CRYSTALS - "Do you want to use the structure from Sir92?" - Click **Yes.**

The structure is on a special position in the space group, so the model is only about one-quarter of the molecule. It will look something like the picture on the left.
In the "Rename" dialog, click **Collect Atoms** to assemble the atoms into a fragment. Then click **Done**.
In the "Renumber" dialog, just click **Automatic**.

Attempt normal refinement

From the **Refinement** menu, choose the top entry, **The Guide**.

The Guide is a dialog box which recommends the next step in the analysis. It should recommend *Refine posn and iso*, which is short for "Refine atomic positions and isotropic displacement parameters".



Click **OK** to accept its recommendation.

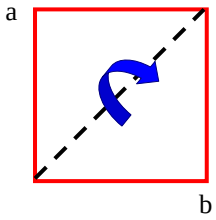
A dialog box appears to confirm the setup of the least squares directives (List 12). Click **OK** to accept and carry out some cycles of refinement.

Keep an eye on the R-factor in the **Refinement** tab at the bottom left of the window. You will find that it fails to refine to a decent R-factor. (22% is about the best you can do with isotropic refinement – this is common with some twins – they will solve, but not refine.)

The twin law

This type of twinning is easily treated in CRYSTALS – once the twin law is identified. The most common type of twinning in tetragonal systems is a 180° rotation about the [110] direction (i.e. the diagonal across the a-b unit cell face) which transforms the indices as shown.

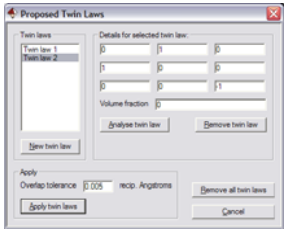
$$h' = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} .h$$



Twinned Refinement: Easy

Guided treatment of twins is implemented in the CRYSTALS GUI:
Choose **Tools** -> **Input Twinned Data** from the menus.

The number of twin laws is **2**, the first is the identity the second, is the rotation around the face diagonal. [***0 1 0; 1 0 0; 0 0 -1]***



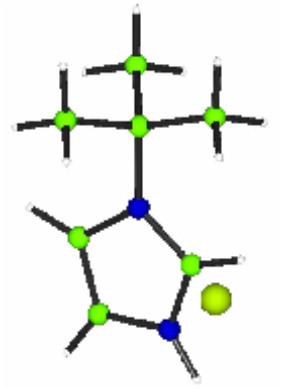
Click "**New twin law**", and then type the new twin law into the 9 edit boxes.
The option "Analyse twin law" isn't any use for merohedral twins as it calculates theoretical overlaps. Similarly there is no need to change the Overlap Tolerance value as all reflections will overlap exactly under the proposed twin law.
Click "**Apply twin law**".to close the dialog.
Click "**Yes**".to verify the element assignments.

In the new list of hkl data, the last column contains a list of twin elements which contribute to the current reflections. Every reflection should have a contribution from both domains in a merohedral twin (i.e. each element tag should be "12").

Close the ***GUIDE*** (by clicking **Hide**) and then restart it (**Refinement->The Guide**) so that it can assess all the new information. Carry on with the refinement.

The R-factor should drop immediately to below 7%, and at the end of the guided refinement should end up below 3%

Keen ($C_7H_{13}ClN_2$): Pseudo-merohdral twin



This data for this material were collected by Sofia Pascu on a Nonius KappaCCD diffractometer with Mo Kα radiation.

Expected Formula: C_7 H_{13} Cl N_2. Temperature 150K
Monoclinic *b* unique

- The structure was solved with 2 molecules in the asymmetric unit in $P2_1$ but failed to refine to an R-factor below 30%.
- Using the systematic *weaknesses* as a guide, it was postulated that the sample was a poor quality in **P 2_1/c**. The structure again solved in this space group, but failed to refine below 20%.
- Finally, it was recognised that the material was twinned. The R-factor falls to below 3%, with all the hydrogen atoms showing clearly in a difference map.

Open the workshop structure "**14. Keen (twin)**".
As before, use **X-ray Data->Inport Shelx file (INS or RES)** to import the data (*keen.ins & keen.hkl*),
Enter the space group **P 2_1/c.**
The structure should solve in SIR92 using **Solve->Sir92.**
Close Sir92, and click **Yes** to import the structure into CRYSTALS.

In the "Rename" dialog, change the 'Click action' setting to *Delete* and then click on the two spurious C atoms. Be careful not to delete the chlorine anion - hover the mouse over an atom to see it's element type.

If you make a mistake, the original solution can be re-imported using the menu item **Structure->Input->Sir92 file.**

Start the guide (**Refinement->The Guide**). and follow its recommendations.
You will find that it still fails to refine to a decent R-factor.

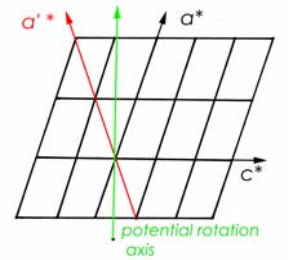
Twin diagnosis:

See **Analyse->Tabbed Initial Analysis** for some of these tools.

- The **Merging R** tab shows that R_{int} is good, especially for the medium and strong reflections.
- The **Absences** tab shows that the mean value for the systematic absences is unusually high (it is almost always greater than zero).

Start worrying about the crystal being twinned. A scale drawing of the reciprocal lattice shows the possibility of a pseudo-orthorhombic super-cell (in red).

The pseudo-orthorhombic super lattice means that twinning is possible by rotation about either *a* (green arrow) or c^*.



A rotation of 180º about *c**, transforms the indices as follows:

$$h' = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 1 & 0 & 1 \end{bmatrix} .h$$

Rotation of the whole lattice about either axis does not result in exact overlap of reciprocal lattice points - this type of twining is termed pseudo-merohedral (TLQS) twinning. However, the almost orthorhombic β angle of 91.13 degrees means that in practice both lattices overlap, so that every observation contains contributions from both twin domains.

Twinned Refinement: Easy

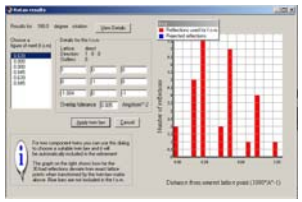
If you can work out the twin law in this way, then you could input it as in the previous example (just read – don't type until you pass *however*!)

You could choose Tools -> Input Twinned Data from the menus.
The number of matrices is **2**, of which the first is the identity.
The second would be [-1 0 0 ; 0 -1 0 ; 1 0 1.]

HOWEVER:

The ROTAX program has been integrated into CRYSTALS, and can often do the work for you.

From the menu select **Analyse**, then **Rotax analysis/twins**, then **ROTAX**.
Click **OK** to start the analysis.



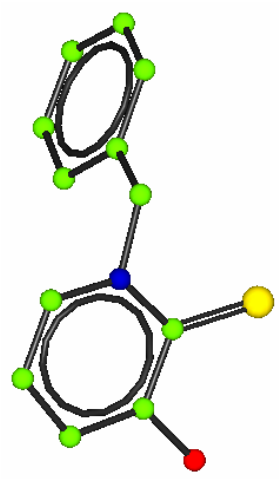
Choose one of the lowest figures of merit by clicking on it. (There should usually be two lowest figures - corresponding to nearly parallel directions in real and reciprocal space. In this case there are four lowest figures as there are two equivalent twin laws about *a* and c^* described above)

The twin law is displayed in the matrix to the right. The graph on the right shows how close a selection of reflections come to exact overlap when the proposed twin law is applied.
Click **Apply Twin Law**.

The twin law is automatically added into the list of proposed twin laws as *Twin Law 2*. You can select it and then click the Analyse Twin Law button to see the distribution of distances between lattice points of the two components. This may be used to pick an Overlap Tolerance value such that overlapping reflections are flagged correctly.
However, in this case, the deviation in reciprocal Angstroms is tiny all the reflections overlap almost exactly – the graph is quite boring.

Click **Apply twin laws.**
You may verify the twin elements assignments. Every reflection should have an element tag "12"
And that's it. Carry on with the refinement as advised by the guide, which will automatically include the twin components in the refinement.
The R-factor should drop very quickly, and at the end of the guided refinement should end up below 4%

AS19A2: Non-merohedral Twin



Summary information

This crystal was synthesised by Andrew Smith and the data was collected by Simon Parsons in Edinburgh.
The cell is monoclinic **P 2_1/n**

Instructions

From the workshop dialog, choose "6. AS19A2 (Twin)"

Import the SHELX *as19a2.ins* & *as19a2.hkl* files.

Solve the structure in SIR92 and read the results back into CRYSTALS.

Sir92 will place the Nitrogen incorrectly.
Use the dialog to correct the atom types to match the diagram above.

The structure won't refine very well (it will stick at R>14%).

Use ROTAX to find the twin law (hint: it's the one with the lowest figure-of-merit). And then complete the refinement.