```
module { phase_distribution }
     (
   &fp;  {input: native data}
   &sp;  {input: native sigma}
   &sel; {input: selection of structure factors}
   &fh;  {input: name of heavy atom structure factors}
   &fph; {input: name of derivative data array}
   &sph; {input: name of derivative's sigma array}
   &var; {input: lack-of-isomorphism plus measurement errors}
   &pa;  {output: Hendrickson and Lattman A array}
   &pb;  {output: Hendrickson and Lattman B array}
   &pc;  {output: Hendrickson and Lattman C array}
   &pd;  {output: Hendrickson and Lattman D array}
     )

do (&pa= (cos(centric_phase)*(
        -(abs(&fh+combine(abs(&fp),centric_phase))-abs(&fph))^2/(2*&var)
        +(abs(&fh-combine(abs(&fp),centric_phase))-abs(&fph))^2/(2*&var)))
  ( centric and &sel )

do (&pb=(sin(centric_phase)*(
        -(abs(&fh+combine(abs(&fp),centric_phase))-abs(&fph))^2/(2*&var)
        +(abs(&fh-combine(abs(&fp),centric_phase))-abs(&fph))^2/(2*&var)))
  ( centric and &sel )

do (&pc=0) (centric and &sel)

do (&pd=0) (centric and &sel)

do (&pa=-2 * (amplitude(&fp)^2 + amplitude(&fh)^2 - amplitude(&fph)^2 )
           * amplitude(&fp) * real(&fh)/
           (3 * &var^2 + 4 * (amplitude(&fph)^2+&sph^2) * &var))
  ( acentric and &sel )

do (&pb=-2 * (amplitude(&fp)^2 + amplitude(&fh)^2 - amplitude(&fph)^2 )
           * amplitude(&fp) * imag(&fh)/
           (3 * &var^2 + 4 * (amplitude(&fph)^2+&sph^2) * &var) )
  ( acentric and &sel )

do (&pc=-amplitude(&fp)^2 * (real(&fh)^2 - imag(&fh)^2) /
         (3 * &var^2 + 4 * (amplitude(&fph)^2+&sph^2) * &var) )
  ( acentric and &sel )

do (&pd=-2 * amplitude(&fp)^2 * real(&fh) * imag(&fh) /
        (3 * &var^2 + 4 * (amplitude(&fph)^2+&sph^2) * &var) )
  ( acentric and &sel )
```

(*a*)

```
@phase_distribution
  (
   &fp=fobs;
   &sp=sigma;
   &sel=( d > 3. );
   &fh=f_heavy;
   &fph=f_deriv;
   &sph=s_deriv;
   &var=variance;
   &pa=pa;
   &pb=pb;
   &pc=pc;
   &pd=pd;
  )
```

(*b*)

Fig. 25.2.3.6. Example of (*a*) a CNS module and (*b*) the corresponding module invocation. Input and output parameters are in bold. The module invocation is performed by specifying the '@' character, followed by the name of the module file and the module parameter substitutions. The ampersand (&) indicates that the particular symbol (*e.g.* '&fp') is substituted with the specified value in the invocation statement [*e.g.* 'fobs' in the case of '&fp' in (*b*)]. The module parameter substitution is performed literally, and any string of characters between the equal sign and the semicolon will be substituted.

Experimental phasing
- Heavy-atom (Patterson) searches
- Patterson refinement
- Multiple isomorphous replacement phasing and site refinement
- Multi-wavelength anomalous dispersion phasing and site refinement

Molecular replacement
- Patterson real-space and direct rotation searches
- Patterson-correlation refinement
- Fast FFT translation search

Density modification
- Creation of envelopes
- Solvent flattening
- Density averaging
- Histogram matching

Refinement
- Maximum-likelihood targets
- Torsion-angle molecular dynamics
- Cartesian molecular dynamics
- Conjugate-gradient minimization
- Composite annealed omit map

Other
- Protein Data Bank deposition file generation
- mmCIF file creation

Fig. 25.2.3.7. Procedures and features available in CNS for structure determination by X-ray crystallography.

25.2.3.6. *Modules and procedures*

Modules exist as separate files and contain collections of CNS commands related to a particular task. In contrast, *procedures* can be defined and invoked from within any file. Modules and procedures share a similar parameter-passing mechanism for both input and output. Modules and procedures make it possible to write programs in the CNS language in a manner similar to that of a computing language, such as Fortran or C. CNS modules and procedures have defined sets of input (and output) parameters that are passed into them (or returned) when they are invoked. This enables long collections of CNS language statements to be broken down into modules for greater clarity of the underlying algorithm.

Parameters passed into a module or procedure inherit the scope of the calling task file or module, and thus they exhibit a behaviour analogous to most computing languages. Symbols defined within a module or procedure are purely local variables.

The following example shows how the unit-cell parameters defined above (Fig. 25.2.3.2*b*) are passed into a module named 'compute_unit_cell_volume' (Fig. 25.2.3.5), which computes the volume of the unit cell from the crystal lattice parameters using well established formulae (Stout & Jensen, 1989):

```
@compute_unit_cell_volume (cell = &crystal_lattice.unit_cell;
                           volume = $cell_volume;)
```
(25.2.3.14)

The parameter 'volume' is equated to the symbol '$cell_volume' upon invocation in order to return the result (the unit-cell volume) from this module. Note that the use of compound parameters to define the crystal lattice parameters (Fig. 25.2.3.2*b*) provides a convenient way to pass all required information into the module by referring to the base name of the compound parameter ('&crystal_lattice.unit_cell') instead of having to specify each individual data element.

Fig. 25.2.3.6(*a*) shows another example of a CNS module: the module named 'phase_distribution' computes phase probability distributions using the Hendrickson & Lattman formalism (Hendrickson & Lattman, 1970; Hendrickson, 1979; Blundell & Johnson, 1976). An example for invoking the module is shown in Fig. 25.2.3.6(*b*). This module could be called from task files that need access to isomorphous phase probability distributions. It would be straightforward to change the module in order to compute different expressions for the phase probability distributions.

A large number of additional modules are available for crystallographic phasing and refinement. CNS library modules include space-group information, Gaussian atomic form factors, anomalous-scattering components, and molecular parameter and topology databases.

Copyright © 2006 International Union of Crystallography