

# SculptorCNS : Biomolecular Structure Determination and Refinement in Solution from NMR Residual Dipolar Couplings, Paramagnetic Phenomena and Spin Relaxation Data

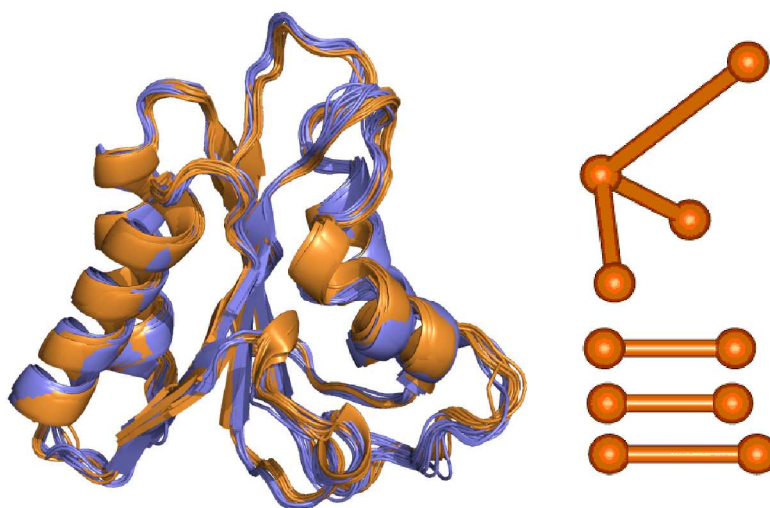
*Céline Charavay<sup>1</sup>, Jean-Paul Eynard<sup>1</sup>, Jean-Christophe Hus<sup>2</sup>, Guillaume Bouvignies<sup>2</sup> and  
Martin Blackledge<sup>2</sup>*

<sup>1</sup> GIPSE, scientific software development group;

CEA ; CNRS IFR 27, 17 rue des Martyrs, Grenoble 38054, France

<sup>2</sup> Protein Dynamics and Flexibility by NMR, Institut de Biologie Structurale Jean-Pierre Ebel,

CEA; CNRS; UJF UMR 5075, 41 Rue Jules Horowitz, Grenoble 38027, France



We present a suite of programs, including functional representations of the following long-range structural restraints in the form of explicit hybrid potentials that are included in the total energy function of an energy minimization, restrained molecular dynamics or simulated annealing approach:

*Residual dipolar couplings*

*Residual Chemical Shift Anisotropy*

*Paramagnetic Residual Dipolar Couplings*

*Paramagnetic Pseudocontact Shifts*

*Curie-Dipole Cross Correlated Relaxation*

*R<sub>2</sub>/R<sub>1</sub> Auto-Relaxation Rates*

Sculptor was specifically written to allow maximal flexibility in the development of conformational search algorithms using long-range structural restraints. As many of these interactions depend on second rank tensorial interactions (alignment tensors, magnetic susceptibility tensors or rotational diffusion tensors), particular attention is paid to the treatment of this kind of tensor in the program.

The program was originally interfaced in its initial format to the program Discover, and has now been entirely rewritten, updated to include the most recent developments of tensor-based constraints for NMR structure refinement, and interfaced to CNS version 1.2, a popular program for crystallographic and NMR structure determination. Sculptor-CNS is distributed free of charge to academic users.

## OUTLINE

We present Sculptor (Structure Calculation Using Long-range, Paramagnetic, Tensorial and Orientational Restraints) a program that allows the use of liquid crystal-induced and paramagnetic residual dipolar couplings, residual chemical shifts, paramagnetic pseudo contact shifts, paramagnetic relaxation effects and autorelaxation R2/R1 ratios, to determine or refine protein and nucleic acid structures.

Sculptor was specifically written to allow maximal flexibility in the development of conformational search algorithms using long-range structural restraints. The program was originally interfaced, in its initial format to the program Discover, and has now been entirely rewritten and updated to include the most recent developments of tensor-based constraints for NMR structure refinement. The program is now embedded in CNS version 1.2, a popular program for crystallographic and NMR structure determination and is distributed free of charge to academic users.

### *Implementation of Sculptor Restraints into CNS*

Molecular dynamic programs allow the simulation of the evolution of atoms in a molecule as a function of time using a simplified potential energy function that describes both covalence and through space interactions in a simple but efficient form. They can also be used to drive a molecular structure into a conformation in agreement with experimental data using a hybrid energy function with components deriving from the physical energy, and from the experimental constraint:

$$E_{tot} = E_{physical} + E_{experimental}$$

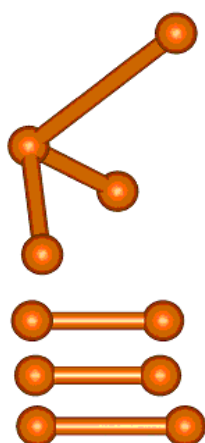
Each of the six constraint types introduced in Sculptor is described in terms of a harmonic potential of the form :

$$E = k \sum_{i=1}^N \left[ \frac{X_i^{exp} - X_i^{calc}}{\sigma_i} \right]^2$$

where  $k$  represents the force constant,  $N$  is the number of constraints,  $X_i^{\text{exp}}$  the experimental value of the constraint  $i$ ,  $X_i^{\text{calc}}$  the calculated value of constraint  $i$  and  $\sigma_i$  is the experimental error associated with constraint  $i$ .

### *Sculptor-CNS and Second Rank Tensors*

As many of these interactions depend on second rank tensorial interactions (alignment tensors, magnetic susceptibility tensors or rotational diffusion tensors), particular attention is paid to the treatment of this kind of tensor in the program. Tensor parameters are treated as independent pseudomolecules, and are read independently of the coordinates of the molecule of interest. Eigenvalues and eigenvectors are treated separately, such that one two-point molecule represents each of the eigenvalues and a third, 3 point orthogonal axis system represents the orientation  $(\alpha, \beta, \gamma)$  of each tensor, so that each term could be manipulated independently. An alternative description was tested, using the lengths of the axes to define the tensor eigenvalues, but this description was found to be less stable. Any number of independent alignment tensor functions are available that can be used separately for restraints from different alignment media or paramagnetic centres or to refine different regions of a macromolecule independently if differential order is suggested to exist in different molecular subdomains.



Note that for tensors with zero trace, only two eigenvalues are defined, and that for the orientation of the tensor only two axes are required. The axes orientations therefore only require a three atom representation.

### *Gradient Calculation*

Calculations of the change of energy as a function of Cartesian coordinate displacement are essential for energy minimization and molecular dynamics calculations. In Sculptor-CNS this can be achieved either analytically or numerically as follows :

$$\frac{\partial}{\partial x_i} f(x_1, \dots, x_i, \dots, x_n) = \lim_{h \rightarrow 0} \frac{f(x_1, \dots, x_i + h, \dots, x_n) - f(x_1, \dots, x_i, \dots, x_n)}{h}$$

### *Addition of new constraints*

CNS already treats a number of NMR constraint types, for example:

- NOE distances
- NOE intensities
- Torsion angle restraints
- Scalar coupling constants
- Chemical shifts
- $^{13}\text{C}\alpha - ^{13}\text{C}\beta$  secondary shifts
- RDCs
- $R_2 / R_1$

As we can see, SANI (RDC) and DANI ( $R_2/R_1$ ) are already handled in CNS. The aim of Sculptor is not to usurp these protocols, but to provide an alternative that has a slightly different way of handling the restraints. The main differences are in the way the tensors are treated and the way the constraint types are managed (see below).

Sculptor-CNS is of course completely compatible with all existing restraint types available in CNS version 1.2.

### **Sculptor Constraint types**

*1)  $R_2/R_1$  restraints are given by :*

$$\frac{R_2}{R_1} = \frac{\frac{d}{2}[4J(0) + J(\omega_H - \omega_N) + 3J(\omega_N) + 6J(\omega_H) + 6J(\omega_H + \omega_N)] + \frac{c}{6}[4J(0) + 3J(\omega_N)]}{d[J(\omega_H - \omega_N) + 3J(\omega_N) + 6J(\omega_H + \omega_N)] + cJ(\omega_N)}$$

$$d = \frac{1}{10} \left( \frac{\mu_0}{4\pi} \right)^2 \frac{\gamma_N^2 \gamma_H^2 \hbar^2}{r_{NH}^6} \quad c = \frac{2}{15} (\sigma_{//} - \sigma_{\perp})^2 \gamma_S^2 B^2 \quad J(\omega) = \sum_{k=1}^5 A_k \frac{\tau_k}{1 + \omega^2 \tau_k^2}$$

$J$  : spectral density function

$\mu_0$  : permiitiivity of free space

$\omega = \gamma B$  : frequency

$\gamma$  : gyromagnetic ratio

$B$  : magnetic field (Tesla)

$\hbar$  : Planck's constant

$r_{NH}$  : internuclear NH distance (Å)

$(\sigma_{//} - \sigma_{\perp}) = 160 \text{ ppm}$

2) Residual dipolar couplings induced by dilute liquid crystal are given by :

$$D_{ij} = -S \frac{\gamma_i \gamma_j \mu_0 \hbar}{8\pi^3 r_{ij}^3} \left[ A_{zz} \cos^2 \theta + A_{yy} \sin^2 \theta \sin^2 \varphi + A_{xx} \sin^2 \theta \cos^2 \varphi \right]$$

$$D_{ij} = -S \frac{\gamma_i \gamma_j \mu_0 \hbar}{16\pi^3 r_{ij}^3} \left[ D_a (3\cos^2 \theta - 1) + \frac{3}{2} D_r \sin^2 \theta \cos(2\varphi) \right]$$

By convention  $|A_{zz}| > |A_{yy}| > |A_{xx}|$  and

$$D_a = \frac{1}{3} \left( A_{zz} - \frac{A_{xx} + A_{yy}}{2} \right) = \frac{A_{zz}}{2}$$

$$D_r = \frac{1}{3} (A_{xx} - A_{yy})$$

$$D_{xx} + D_{yy} + D_{zz} = 0$$

$\gamma_i, \gamma_j$  : gyromagnetic ratio of nuclei i and j

$D_a$  : axial component of the alignment tensor

$D_r$  : rhombic component of the alignment tensor

$\theta, \varphi$  : polar coordinates of  $\vec{ij}$  in the alignment tensor frame

$r_{ij}$  : distance between spins  $\vec{ij}$

$S$  : local motional order parameter

$h$  : Planck's constant

$\mu_0$  : permiitiivity of free space

RDCs can be introduced between any atoms in the molecule.

3) *Residual chemical shifts induced by dilute liquid crystal are given by :*

$$\delta_i = \sum_{k=x,y,z} \sum_{l=1,2,3} \{A_{kk} \sigma_{ll} \cos^2 \theta_{kl}\}$$

(x,y,z) : axes of the alignment tensor

$\sigma_{ll}$  : eigenvalues of the CSA tensor.  $\sigma_{11}$  and  $\sigma_{33}$  are taken from Markwick et al (J. Am. Chem. Soc., 126, 11424 -11425, 2004).  $\sigma_{22}$  is calculated on the basis of the isotropic shift as outlined in this publication.

$A_{kk}$  : eigenvalues of the alignment tensor

$\theta_{kl}$  : angle between the axes of the csa and alignment tensors

4) *Residual dipolar couplings induced from intrinsic or paramagnetic susceptibility anisotropy are given by :*

$$D_{ij} = -\frac{\gamma_i \gamma_j \hbar}{8\pi^2 15 N_A k_B T r_{ij}^3} (B_2^2 - B_1^2) S \left[ \Delta\chi_a (3\cos^2 \theta - 1) + \frac{3}{2} \Delta\chi_r \sin^2 \theta \cos 2\phi \right]$$

$\gamma_i, \gamma_j$  : gyromagnetic ratio of nuclei i and j

$\Delta\chi_a$  : axial composant of the magnetic susceptibility anisotropy tensor (m<sup>3</sup>/mol)

$\Delta\chi_r$  : rhombic composant of the magnetic susceptibility anisotropy tensor (m<sup>3</sup>/mol)

$\theta, \phi$  : polar coordinates of  $\vec{ij}$  in the alignment tensor frame

$r_{ij}$  : distance between spins  $\vec{ij}$

$S$  : local motional oder parameter

$h$  : Planck's constant

$\mu_0$  : permiitiivity of free space

$T$  : temperature

$B_2, B_1$  : static field (if two, otherwise one can be set to zero)

$k_B$  : Boltzmann's constant

$N_A$  : Avogadro's number

$$\Delta\chi_a = \left( \Delta\chi_{zz} - \frac{\Delta\chi_{xx} + \Delta\chi_{yy}}{2} \right) \quad \Delta\chi_r = (\Delta\chi_{xx} - \Delta\chi_{yy})$$

( $\Delta\chi_{xx} + \Delta\chi_{yy} + \Delta\chi_{zz} = 0$  ; tensor eigenvalues)

5) Pseudo contact shifts induced from paramagnetic susceptibility anisotropy are given by :

$$\delta_{dep} = \frac{1}{12N_A\pi r^3} \left[ \Delta\chi_a(3\cos^2\theta - 1) + \frac{3}{2}\Delta\chi_r \sin^2\theta \cos 2\phi \right]$$

Constants as in (4)

6) Curie-dipole cross correlated relaxation is given by :

$$\Gamma_{H,H-N}^{Curie,DD} = \frac{2}{5} \left( \frac{\mu_0}{4\pi} \right)^2 \frac{h}{2\pi} \frac{\gamma_H^2 \gamma_N}{r_{H-N}^3} \chi B_0 \tau_c \frac{3\cos^2\theta - 1}{r_{H-S}^3}$$

$\chi$  : magnetic susceptibility

$B_0$  : magnetic field

$\tau_c$  : correlation time

$r_{H-S}$  : distance between paramagnetic spin S and proton H

$\theta$  : angle between vectors H-N et H-S

The value of the magnetic susceptibility  $\chi$  is treated as a scalar value in SCULPTOR, that can be optimized during the calculation.



## ***Articles describing use of SCULPTOR :***

*De novo* Determination of Protein Structure by NMR using Orientational and Long-Range Order Restraints. J.-C. Hus, D. Marion and M. Blackledge. *J.Mol.Biol.* 298, 927-936, (2000).

Rotational diffusion anisotropy and local backbone dynamics of carbon monoxide-bound *Rhodobacter capsulatus* Cytochrome *c'*. P. Tsan, J.-C. Hus, M. Caffrey, D. Marion and M. Blackledge. *J.Am.Chem.Soc.* 122, 5603-5612, (2000).

Refinement of Local and Long-Range Structural Order in Theophylline-Binding RNA using <sup>13</sup>C-1H Residual Dipolar Couplings and Restrained Molecular Dynamics. N.Sibille, A.Pardi, J.-P. Simorre and M.Blackledge. *J.Am.Chem.Soc.* 123, 12135-12146, (2001).

Reactivity, Secondary Structure and Molecular Topology of the /*Escherichia coli*/ Sulfite Reductase Flavodoxin-like Domain. L. Champier, N. Sibille, B. Bersch, B. Brutscher, M. Blackledge and J. Coves. *Biochemistry*, 41, 3770-3780 (2002).

Reaction site conformation of methionine sulfoxide reductase in solution: Direct structure determination from dipolar couplings. Sabine Béraud, Beate Bersch, Bernhard Brutscher, Pierre Gans, Frédéric Barras and Martin Blackledge *J.Am.Chem.Soc.* 124, 13709-13715 (2002).

Relaxation-based Structure Refinement and Backbone Molecular Dynamics of the Dynein Motor Domain Associated Light Chain. Hongwei Wu, Martin Blackledge, Mark Maciejewski, Gregory Mullen and Steve King *Biochemistry* 42, 57-71 (2003).

Biophysical characterisation of the amino-terminal extension of the mercuric reductase from *Ralstonia metallidurans* CH34 is a MerP-like module. E. Rossy, L.Champier, B. Bersch, B. Brutscher, M. Blackledge and J. Covès *Journal of Biological Inorganic Chemistry* 9, 49-58 (2004).

Structure and dynamics of the nucleocapsid-binding domain of the Sendai virus phosphoprotein in solution. L. Blanchard, N. Tabouriech, M. Blackledge, P.Timmins, W.P. Burmeister, R.W.H.Ruigrok and D.Marion *Virology* 319, 201-211 (2004)

Effect of the solvent on the conformation of a depsipeptide: \* NMR derived solution structure of hormaomycin in DMSO by using residual dipolar couplings in a novel DMSO compatible alignment medium U. Reinscheid, J. Farjon, M. Radzom, P. Haberz, A. Zeeck, M. Blackledge, C. Griesinger. *ChemBioChem.* 7, 287-296 (2006).

Solution structure of the C-terminal nucleoprotein-RNA binding domain of the vesicular stomatitis virus phosphoprotein. E. Ribeiro, A. Favier, F.C.A. Gerard, C. Leyrat, B. Brutscher, D. Blondel, R.W.Ruigrok, M. Blackledge\* and M. Jamin\*. *J. Mol. Biol.* 382, 525-538 (2008).

## Command structures of SculptorCNS

Seven command structures have been added to the CNS language in order to account for the 6 new constraint types. These commands, whose syntax is described in more detail in the annexe are listed below :

- The command STENSOR :

The command STENSOR (SCULPTOR-Tensor) allows the declaration of tensors. Using this command we can define all of the relevant characteristics of the tensor :

- name
- definition of the molecule representing the tensor principal axis system
- definition of the molecules representing the eigenvalues of the tensor
- definition of the constants allowing the calculation of the tensor eigenvalues.
- definition of a harmonic potential restraining the eigenvalues

Once the tensor has been declared it is possible to specify the experimental orientational constraints via one of the commands SRDC, SRELAXATION, SPARA, SCURIE, SDPL, SCSA, referring to the name of the declared tensor, facilitating the declaration of the constraints. STENSOR thus allows the declaration of as many tensors as are required.

The STENSOR command is organised around five sub-commands : ALIGNMENT, CURIE, DELTGRAD, RELAXATION et SUSCEPTIBILITY. These will be described briefly here, while the syntax is presented in detail in the Annexe.

- Sub-commands ALIGNMENT, RELAXATION and SUSCEPTIBILITY :

These three commands allow declaration and examination of the tensor properties

For each sub-command there are five respective additional sub-commands :

- sub-command **declare** declares a new tensor: tensor name (**name**), molecule representing the tensor principal axis system (**orientation**), definition of the molecules representing the eigenvalues of the tensor (**eigenvalues**), definition of the constants allowing the calculation of the tensor eigenvalues (**evcalcul**) definition of a harmonic potential restraining the eigenvalues (**evrestraints**).
- **reset** removes all declared tensors
- **print** presents all tensor properties

- **order** presents the tensor using standard conventions (see above)

- The command CURIE :

The command CURIE allows declaration and examination of the properties of molecules that approximate the value of the coefficient  $\chi$  for associated restraints (Curie-dipole cross correlated relaxation. Although the program represents this as a scalar it has been included in the STENSOR section.

CURIE proposes four sub-commands:

- **declare** declares a new molecule (**name**), molecule atoms (**molecule**) and parameters defining restraints on  $\chi$  (**khirestraints**).
- **reset** removes all such declarations
- **print** presents tensor properties.

- The command DELTGRAD :

DELTGRAD sets the numeric value of the step used to calculate the gradient

- The command SRDC :

The command SRDC (SCULPTOR Residual Dipolar Constraint) allows the declaration, treatment and visualization of RDCs measured in dilute liquid crystalline media.

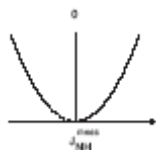
We thus find 12 sub-commands dedicated to this command :

- **nrestraints** declares the maximum number of this type of restraint
- **deltgrad** fixes the step for gradient calculation.
- **class** declares a new class containing a type of restraint : class name (**name**), force constant (**forceconstant**), RDC type (**rdctype**), alignment tensor associated with this class (**tensorname**), declaration of the experimental values (**assign**)
- **reset** removes all such declarations of this type of restraint
- **print** prints all experimental and calculated values of this class of restraint
- **rccconstant**, **rcc0**, **rchconstant**, **rch0**, **rcnconstant**, **rcn0**, **rnhconstant** and **rnh0** define whether CC, CH, CN NH RDCs should be calculated from the structure or held constant at fixed values.

For the energy calculation three options are possible :

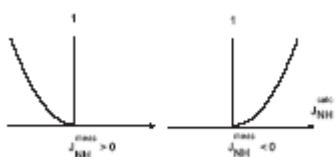
- option 1  $\Leftrightarrow$  classical harmonic potential

$$E = k \left[ \frac{X^{\text{exp}} - X^{\text{calc}}}{\sigma} \right]^2$$



- option 2 ⇔ allows values to be smaller than the constraint (due to mobility)

If  $|X^{\text{calc}}| > |X^{\text{obs}}|$  then  $E = 0$  ; else  $E = k \left[ \frac{X^{\text{exp}} - X^{\text{calc}}}{\sigma} \right]^2$



- option 3 ⇔ unsigned potential

$$E = k \left[ \frac{|X^{\text{exp}}| - |X^{\text{calc}}|}{\sigma} \right]^2$$

These different options are selected when the constraint is declared (**assign** under the command class)

- **The command SRELAXATION :**

SRELAXATION (SCULPTOR Relaxation) treats  $R_2 / R_1$  constraints.

The structure, declarations and potential types of the SRELAXATION command are very similar to the command SRDC. See above.

- **The command SPARA :**

SPARA (SCULPTOR Paramagnetic) treats RDCs due to paramagnetic alignment, its structure, declaration and potential types are again similar to SRDC.

- **The command SCURIE :**

The command SCURIE (SCULPTOR Curie) treats Curie-dipole cross correlated relaxation restraints and is again similar in structure, declaration and potential types to SRDC.

- **The command SDPL :**

The command SDPL treats pseudo contact shifts due to the presence of a paramagnetic centre and is again similar in structure, declaration and potential types to SRDC.

- **The command SCSA :**

SCSA (SCULPTOR Chemical Shift Anisotropy) treats residual chemical shifts due to partial molecular alignment and is again similar in structure, declaration and potential types to SRDC.

## **Output**

- Sub-command print of commands SRDC, SRELAXATION, SPARA, SCURIE, SDPL and SCSA

The syntax of the **print** command :

**print** threshold <real> **class** <classname> **output** <output> **format** module|human **end**

where <classname> ::= <word> | all

<output> ::= <filename> | screen

<filename> ::= <word>

This command prints constraint violations above a specified threshold. Additional options :

- « **class** <classname> » selects the class (« all » can also be specified).
- « **output** <output> » : screen or filename
- « **format** module|human » selects between MODULE format or other

- Sub-command print of commands ALIGNMENT, CURIE , RELAXATION and SUSCEPTIBILITY of module TENSOR

**print data** <data> **tensor** <tensorname> **output** <output> **end**

where <data> ::= angles | coordinates | eigenvalues | all

<tensorname> ::= <word> | all

<output> ::= <filename>|screen

<filename> ::= <word>

Shows tensor parameters. Additional options :

- « **data** <data> » selects properties of tensor molecules (eigenvalues/angles).
- « **tensor** <tensorname> » selects tensor
- « **output** <output> » selects output (screen/file)

### Sub-command print of the command CURIE

Syntax :

**print data** <datacurie> **curie** <curiname> **output** <output> **end**

where <datacurie> ::= coordinates | khivalue | all

<curiname> ::= <word> | all

<output> ::= <filename>|screen

<filename> ::= <word>

Options

- « **data** <datacurie> » selects properties of tensor molecules
- « **curie** <curiname> » selects scalars to be output
- « **output** <output> » type, screen or file.

# ANNEXE

## SculptorCNS commands syntax

### 1. Introduction

CNS language consists of an instruction set described in the « syntax manual » of the on-line documentation (<http://cns.csb.yale.edu/v1.2/> ). This description is in the form of Backus Naur Form notation (see [http://en.wikipedia.org/wiki/Backus\\_Naur\\_form](http://en.wikipedia.org/wiki/Backus_Naur_form) for more details).

CNS instructions are organized around the main instruction :

**cns\_solve {<cns-statement>} stop**

### 2. New commands

SculptorCNS patch adds seven new command structures to the CNS program :

**<cns-statement> ::=**

**stensor {<tensor-statement>} end**  
**srdc {<srdc-statement>} end**  
**srelaxation {<srelaxation-statement>} end**  
**spara {<spara-statement>} end**  
**scurie {<scurie-statement>} end**  
**sdpl {<sdpl-statement>} end**  
**scsa {<scsa-statement>} end**

The syntax of these seven new commands is described below.

#### a. Stensor command

**<tensor-statement> ::=**

**alignment {<tensor-alignment-statement>} end** # Define alignment tensors  
**curie {<curie-statement>} end** # Define  $\chi$  tensors  
**deltgrad <real>** # Define step value used to calculate some gradients numerically  
# Default=0.00001

```

relaxation {<tensor-relaxation-statement>} end      # Define relaxation tensors
susceptibility {<tensor-susceptibility-statement>} end # Define susceptibility
                                                    # tensors

```

**<tensor-alignment-statement> ::=**

```

declare <alignment-declare-statement> end      # Declare or update an alignment
                                                    # tensor

order <alignment-order-statement> end # Print ordered alignment tensors
# (the order in eigenvalues is changed
# so that :
#  $|D_{XX}| < |D_{YY}| < |D_{ZZ}|$  where
#  $D_{aa} = 1/3 (D_{ZZ} - (D_{XX} + D_{YY}) / 2)$ 
# and  $D_{ra} = 1/3 (D_{XX} - D_{YY})$ 

print <alignment-print-statement> end # Print alignment tensors
reset                                  # Erase the alignment tensor table

```

**<alignment-declare-statement> ::=**

```

name <word>          # Name of the alignment tensor to declare or to update

orientation <atom-selection> <atom-selection>
            <atom-selection> # Define the 3 atoms (o x z ) of the orientation
            # molecule

eigenvalues <atom-selection> <atom-selection>
            <atom-selection> <atom-selection> # Define atoms (ox1 x1,
            # oz1 z1) of the 2 eigen
            # value molecules

evcalcul <real> <real> # Define min / scale values used in the eigenvalue
            # calcul :
            #  $D_{aa} = \text{min} + \text{scale} \times \text{norm}(\text{ox1}-\text{x1})$ 
            #  $D_{ra} = \text{min} + \text{scale} \times \text{norm}(\text{oz1}-\text{z1})$ 
            # Default : min = - 10.0 ; scale = 10.0

evrestraints <real> <real> <real>
            <real> <real> <real> # Define Daamin, Daamax, Daascale,
            # Dramin, Dramax, Drascale values used
            # in the potential on eigenvalues.
            # This potential is defined as follows:
            # if  $D_{aa} > D_{aamax} \Rightarrow E = D_{aascale} (D_{aamax} - D_{aa})^2$ 
            # if  $D_{aa} < D_{aamin} \Rightarrow E = D_{aascale} (D_{aamin} - D_{aa})^2$ 

```



```

# if  $D_{aamin} < D_{aa} < D_{aamax} \Rightarrow E = 0$ 
# if  $D_{ra} > D_{ramax} \Rightarrow E = D_{rascale} (D_{ramax} - D_{ra})^2$ 
# if  $D_{ra} < D_{ramin} \Rightarrow E = D_{rascale} (D_{ramin} - D_{ra})^2$ 
# if  $D_{ramin} < D_{ra} < D_{ramax} \Rightarrow E = 0$ 
# Default :  $D_{aamin} = - 4.00 ; D_{aamax} = 4.00 ;$ 
#  $D_{aascale} = 100.0 ; D_{ramin} = - 4.00 ;$ 
#  $D_{ramax} = 4.00 ; D_{rascale} = 100.0$ 

```

**<alignment-order-statement> ::=**

```

tensor <tensorname>      # Name of the tensor to print
output <output>          # Output (file or screen)

```

**<alignment-print-statement> ::=**

```

data <data>              # Data to print (angles, eigenvalues, coordinates or all)
tensor <tensorname>      # Name of the tensor to print
output <output>          # Output (file or screen)

```

**<curie-statement> ::=**

```

declare <curie-declare-statement> end  # Declare or update a curie tensor  $\chi$ 
                                           # approximation
print <curie-print-statement> end      # Print " $\chi$  tensor"
reset          # Erase the  $\chi$  tensor table

```

**<curie-declare-statement> ::=**

```

name <word>              # Name of the  $\chi$  tensor to declare or to update
molecule   <atom-selection> <atom-selection>  # Define 2 atoms (o x )
                                                    #of tensor  $\chi$ 
                                                    # (norm of molecule
                                                    # will be equal to the  $\chi$ 
                                                    # for magnetic
                                                    # susceptibility tensor in
                                                    # its isotropic
                                                    # approximation)
khirestraints <real> <real> <real>  # Define  $\chi_{min}$ ,  $\chi_{max}$ ,  $\chi_{scale}$  values used
                                                    # in the potential on  $\chi$  coefficient.
                                                    # This potential is defined as followed:
                                                    # if  $\chi > \chi_{max} \Rightarrow E = \chi_{scale} (\chi_{max} - \chi)^2$ 

```

```

# if  $\chi < \chi_{\min} \Rightarrow E = \chi_{\text{scale}} (\chi_{\min} - \chi)^2$ 
# if  $\chi_{\min} < \chi < \chi_{\max} \Rightarrow E = 0$ 
# Default :  $\chi_{\min} = 1.00$  ;  $\chi_{\max} = 5.00$  ;
#  $\chi_{\text{scale}} = 1.0$ 

```

**<curie-print-statement> ::=**

```

data <datacurie> # Data to print (khi value, coordinates or both)
curie <curiname> #  $\chi$  tensor to print
output <output> # Output (file or screen)

```

**< tensor-relaxation-statement> ::=**

```

declare <relaxation-declare-statement> end # Declare or update a relaxation
# tensor
order <relaxation-order-statement> end # Print ordered relaxation tensors
# (order recalculated
# so that  $D_{XX} < D_{YY} < D_{ZZ}$ )
#  $\alpha, \beta, \gamma$  amended accordingly
print <relaxation-print-statement> end # Print relaxation tensors
reset # Erase the relaxation tensor table

```

**<relaxation-declare-statement> ::=**

```

name <word> # Name of the relaxation tensor to declare or to update
orientation <atom-selection> <atom-selection>
# Define the 3 atoms (o x z) of the orientation
# molecule
eigenvalues <atom-selection> <atom-selection>
# Define atoms (ox1 x1,
# oy1 y1, oz1 z1) of 3
# eigenvalue molecules
evcalcul <real> <real> # Define min / scale values used in the eigenvalue
# calculation :
#  $D_{XX} = \text{min} + \text{scale} \times \text{norm}(\text{ox1-x1})$ 
#  $D_{YY} = \text{min} + \text{scale} \times \text{norm}(\text{oy1-y1})$ 

```

#  $D_{ZZ} = \text{min} + \text{scale} \times \text{norm}(oz1-z1)$

# Default : min = 0.05 ; scale = 0.10

**evrestraints** <real> <real> <real>

<real> <real> <real>

<real> <real> <real>

# Define D<sub>xxmin</sub>, D<sub>xxmax</sub>, D<sub>xxscale</sub>,  
# D<sub>yymin</sub>, D<sub>yymax</sub>, D<sub>yyscale</sub>, D<sub>zzmin</sub>,  
# D<sub>zzmax</sub>, D<sub>zzscale</sub> values used in the  
# potential on eigenvalues.

# This potential is defined as followed:

# if  $D_{XX} > D_{xxmax} \Rightarrow$

#  $E = D_{xxscale} (D_{xxmax} - D_{XX})^2$

# if  $D_{XX} < D_{xxmin} \Rightarrow$

#  $E = D_{xxscale} (D_{xxmin} - D_{XX})^2$

# if  $D_{xxmin} < D_{XX} < D_{xxmax} \Rightarrow E = 0$

# if  $D_{YY} > D_{yymax} \Rightarrow$

#  $E = D_{yyscale} (D_{yymax} - D_{YY})^2$

# if  $D_{YY} < D_{yymin} \Rightarrow$

#  $E = D_{yyscale} (D_{yymin} - D_{YY})^2$

# if  $D_{yymin} < D_{YY} < D_{yymax} \Rightarrow E = 0$

# if  $D_{ZZ} > D_{zzmax} \Rightarrow$

#  $E = D_{zzscale} (D_{zzmax} - D_{ZZ})^2$

# if  $D_{ZZ} < D_{zzmin} \Rightarrow$

#  $E = D_{zzscale} (D_{zzmin} - D_{ZZ})^2$

# if  $D_{zzmin} < D_{ZZ} < D_{zzmax} \Rightarrow E = 0$

# Default :  $D_{xxmin} = 0.05$  ;  $D_{xxmax} = 0.25$  ;

#  $D_{xxscale} = 100.0$  ;  $D_{yymin} = 0.05$  ;

#  $D_{yymin} = 0.05$  ;  $D_{yyscale} = 100.0$  ;

#  $D_{zzmin} = 0.05$  ;  $D_{zzmax} = 0.25$  ;  $D_{zzscale} = 100.0$

**<relaxation-order-statement> ::=**

**tensor** <tensorname> # Name of the tensor to print

**output** <output> # Output (file or screen)

**<relaxation-print-statement> ::=**

**data** <data> # Data to print (angles, eigenvalues, coordinates or all)

**tensor** <tensorname> # Name of the tensor to print

**output** <output> # Output (file or screen)

**<tensor-susceptibility-statement> ::=**

```
declare <susceptibility-declare-statement> end # Declare or update susceptibility
# tensor
order <susceptibility-order-statement> end #Print ordered susceptibility tensors
print <susceptibility-print-statement> end # Print susceptibility tensors
reset # Erase the susceptibility tensor table
```

**<susceptibility-declare-statement> ::=**

```
name <word> # Name of the susceptibility tensor to declare or to update
orientation <atom-selection> <atom-selection>
<atom-selection> # Define the 3 atoms (o x z ) of the orientation
# molecule
eigenvalues <atom-selection> <atom-selection>
<atom-selection> <atom-selection># Define the atoms (ox1 x1,
# oz1 z1) of the 2 eigen
# value molecules
evcalcul <real> <real> # Define min and scale values
# calculation :
#  $D_{aa} = \text{min} + \text{scale} \times \text{norm}(\text{ox1-x1})$ 
#  $D_{ra} = \text{min} + \text{scale} \times \text{norm}(\text{oz1-z1})$ 
# Default : min = - 10.0 ; scale = 10.0
evrestraints <real> <real> <real>
<real> <real> <real> # Define Daamin, Daamax, Daascale,
# Dramin, Dramax, Drascale values used
# in the potential on eigenvalues.
# This potential is defined as followed:
# if  $D_{aa} > D_{aamax} \Rightarrow E = D_{aascale} (D_{aamax} - D_{aa})^2$ 
# if  $D_{aa} < D_{aamin} \Rightarrow E = D_{aascale} (D_{aamin} - D_{aa})^2$ 
# if  $D_{aamin} < D_{aa} < D_{aamax} \Rightarrow E = 0$ 
# if  $D_{ra} > D_{ramax} \Rightarrow E = D_{rascale} (D_{ramax} - D_{ra})^2$ 
# if  $D_{ra} < D_{ramin} \Rightarrow E = D_{rascale} (D_{ramin} - D_{ra})^2$ 
# if  $D_{ramin} < D_{ra} < D_{ramax} \Rightarrow E = 0$ 
# Default :  $D_{aamin} = - 4.00$  ;  $D_{aamax} = 4.00$  ;
#  $D_{aascale} = 100.0$  ;  $D_{ramin} = - 4.00$  ;
#  $D_{ramax} = 4.00$  ;  $D_{rascale} = 100.0$ 
```

**< susceptibility -order-statement > ::=**

**tensor <tensorname>** # Name of the tensor to print

**output <output>** # Output (file or screen)

**< susceptibility -print-statement > ::=**

**data <data>** # Data to print (angles, eigenvalues, coordinates or all)

**tensor <tensorname>** # Name of the tensor to print

**output <output>** # Output (file or screen)

**<tensorname > ::=**

**<word>|all**

**<curienname > ::=**

**<word>|all**

**<data > ::=**

**angles | coordinates | eigenvalues | all**

**<datacurie > ::=**

**coordinates | khvalue | all**

**<output > ::=**

**<filename > | screen**

**<filename > ::= <word >**

## **b. Srdc command**

**<sdrc-statement > ::=**

**class <class-statement > end** # Declare or update a class of RDC

# restraints

**deltgrad <real >** # Define step value used to calculate some

# gradients numerically

# Default=0.00001

**nrestraints <integer >** # Number of slots for residual dipolar coupling restraints to

```

# allocate in memory
# Default=2000

print <print-statement> end      # Print residual dipolar coupling restraints

rccconstant <boolean> # Define if C-C distances are constant (and equal to rcc0) or
# calculated
# Default = true

rcc0 <real>           # Default = 1.53

rchconstant <boolean> # Define if C-H distances are constant (and equal to rch0) or
# calculated
# Default = true

rch0 <real>          # Default = 1.08

rcnconstant <boolean> # Define if C-N distances are constant (and equal to rcn0) or
# calculated
# Default = true

rcn0 <real>         # Default = 1.33

rnhconstant <boolean> # Define if N-H distances are constant (and equal to rnh0) or
# calculated
# Default = true

rnh0 <real>         # Default = 1.04

rnnconstant <boolean> # Define if N-N distances are constant (and equal to rnn0) or
# calculated
# Default = false

rnn0 <real>         # Default = 1.00

rhhconstant <boolean> # Define if H-H distances are constant (and equal to rhh0) or
# calculated
# Default = false

rhh0 <real>         # Default = 1.00

reset # Erase the RDC assignment table

```

**<class-statement> ::=**

**assign**      **<atom-selection> <atom-selection> <real> <real>**  
**<real> <integer>**    # Declares a new RDC restraint :  
# Parameters 1 and 2 : atoms concerned ;  
# Parameter 3 : value of the RDC  
# restraint  
# Parameter 4 : error  
# Parameter 5 : S value  
# Parameter 6 : type of potential to use (0 is the  
# classical harmonic potential, 1 is such that  
# mobility is taken into account and 2 is the  
# unsigned  
# potential useful for example for H-H RDCs)

**forceconstant <real>**    # Default = 1.00

**name <word>**            # Name of the RDC restraint class to declare or to update

**rdctype nh|ch|cc|cn|nn|hh** # Declare the type of RDC restraints (allows to know  
# which distance constant to use for the class)

**tensorname <word>**      # Name of the tensor

**<print-statement> ::=**

**threshold <real>**    # Print RDC violations greater than the specified threshold

**class <classname>**    # Only violations of the specified RDC class are printed

**output <output>**      # Output (file or screen)

**format module|human**    # Output format (module format or human readable format)

**<classname> ::=**  
**<word>|all**

**<output> ::=**  
**<filename>|screen**

### c. Srelaxation command

**<srelaxation-statement> ::=**

**class <class-statement> end**    # Declare or update a class of relaxation restraints

**csaval <real>**            # Value of CSA used in relaxation

```

# (the actual value is -csaval x 10-6 in ppm)
# Default = 160.0
deltgrad <real> # Define step value used to calculate some gradients numerically
# Default = 0.00001
nrestraints <integer> # Number of slots for relaxation restraints to allocate in memory
# Default=2000
rnhconstant <boolean> # Define if N-H distances are constant (and are rnh0) or
# calculated
# Default = true
rnh0 <real> # Default = 1.04
print <print-statement> end # Print relaxation restraints
reset # Erase the relaxation assignment table

```

**<class-statement> ::=**

```

assign <atom-selection> <atom-selection>
<real> <real> # Declare a new relaxation restraint :
# Parameters 1 and 2 : atoms concerned ;
# Parameter 3 : value of the relaxation restraint (R2/R1
# ratio)
# Parameter 4 : error
field <real> # Value relative to the magnetic field expressed in 1H frequency
# (the actual value is field/600.0 x 600.14297 106)
# Default = 400.0
forceconstant <real> # Default = 1.00
name <word> # Name of the class to declare or to update
tensorname <word> # Name of the tensor

```

**<print-statement> ::=**

```

threshold <real> # Print relaxation violations greater than the specified threshold
class <classname> # Only violations of the specified relaxation class are printed
output <output> # Output (file or screen)
format human # Output format (human readable format)

```

**<classname> ::=**

**<word>|all**

**<output> ::=**

**<filename>|screen**



#### d. Spara command

**<spara-statement> ::=**

```
class <class-statement> end      # Declare or update a class of RDC
                                     # restraints

deltgrad <real>      # Define step value used to calculate some gradients numerically
                                     # Default=0.00001

nrestraints <integer>      # Number of slots for residual dipolar coupling restraints to
                                     # allocate in memory
                                     # Default=2000

print <print-statement> end      # Print residual dipolar coupling restraints

rccconstant <boolean>      # Define if C-C distances are constant (and are rcc0) or
                                     # calculated
                                     # Default = true

rcc0 <real>            # Default = 1.53

rchconstant <boolean>      # Define if C-H distances are constant (and are rch0) or
                                     #calculated
                                     # Default = true

rch0 <real>            # Default = 1.08

rcnconstant <boolean>      # Define if C-N distances are constant (and are rcn0) or
                                     #calculated
                                     # Default = true

rcn0 <real>            # Default = 1.33

rnhconstant <boolean>      # Define if N-H distances are constant (and are rnh0) or
                                     # calculated
                                     # Default = true

rnh0 <real>            # Default = 1.04

rnnconstant <boolean>      # Define if N-N distances are constant (and are rnn0) or
                                     # calculated
                                     # Default = false

rnn0 <real>            # Default = 1.00

rhconstant <boolean>      # Define if H-H distances are constant (and are rhh0) or
                                     #calculated
                                     # Default = false

rhh0 <real>            # Default = 1.00

reset      # Erase the RDC assignment table
```

```

<class-statement> ::=
    assign      <atom-selection> <atom-selection> <real> <real>
                <real> <integer> # Declare a new RDC restraint :
                                # Parameters 1 and 2 : atoms concerned ;
                                # Parameter 3 : value of the residual dipolar coupling
                                # restraint
                                # Parameter 4 : error
                                # Parameter 5 : S value
                                # Parameter 6 : type of potential to use (0 is the
                                # classical harmonic potential and 1 is such that
                                # mobility is taken into account)

    fie1 <real> # Magnetic field (in Tesla)
                # Default = 9.4

    fie2 <real> # Magnetic field (in Tesla)
                # Default = 14.1

    forceconstant <real> # Default = 1.00

    name <word> # Name of the RDC restraint class to declare or to update

    rdctype nh|ch|cc|cn|nn|hh # Declare the type of RDC restraints (allows to know
                                # which distance constant to use for the class)

    temperature <real> # Temperature
                       # Default = 300.0

    tensorname <word> # Name of the tensor

```

```

<print-statement> ::=
    threshold <real> # Print RDC violations greater than the specified threshold

    class <classname> # Only violations of the specified RDC class are printed

    output <output> # Output (file or screen)

    format module|human # Output format (module format or human readable format)

```

```

<classname> ::=

```

```

    <word>|all

```

```

<output> ::=

```

```

    <filename>|screen

```

e. Scurie command

```

<scurie-statement> ::=

```

```

    class <class-statement> end # Declare or update a class of Curie restraints

```

**deltgrad** <real> # Define step value used to calculate some gradients numerically  
# Default=0.00001

**nrestraints** <integer> # Number of slots for Curie restraints to allocate in memory  
# Default=2000

**print** <print-statement> **end** # Print Curie restraints

**reset** # Erase the curie assignment table

**rnhconstant** <boolean> # Define if N-H distances are constant (and are rnh0) or  
# calculated  
# Default = true

**rnh0** <real> # Default = 1.04

**toc** <real> # Correlation time of the protein  
# Default = 10.0

<class-statement> ::=

**assign** <atom-selection> <atom-selection>  
<real> <real> # Declare a new Curie restraint :  
# Parameter 1 : atom N concerned  
# Parameter 2 : atom H concerned  
# Parameter 3 : value of the Curie restraint  
# Parameter 4 : error

**field** <real> # Magnetic field (in Tesla)  
# Default = 9.4

**forceconstant** <real> # Default = 1.00

**name** <word> # Name of the Curie restraint class to declare or to update

**paramagneticcenter** <atom-selection> # Paramagnetic center

**curienname** <word> # Name of the  $\chi$  tensor

<print-statement> ::=

**threshold** <real> # Print Curie violations greater than the specified threshold

**class** <classname> # Only violations of the specified Curie class are printed

**output** <output> # Output (file or screen)

**format human** # Output format (human readable format)

<classname> ::=

<word>|all

<output> ::=

<filename>|screen

## f. Sdpl command

**<sdpl-statement> ::=**

```
class <class-statement> end      # Declare or update a class of pseudo contact shifts
deltgrad <real>      # Define step value used to calculate some gradients numerically
                        # Default=0.00001
nrestraints <integer>      # Number of slots for pseudo contact shift restraints
                        # Default=2000
print <print-statement> end      # Print pseudo contact shift restraints
reset # Erase pseudo contact shifts assignment table
```

**<class-statement> ::=**

```
assign <atom-selection> <real>
    <real>      # Declare a new pseudo contact shifts:
                # Parameter 1 : atom concerned ;
                # Parameter 2 : value of the pseudo contact shifts
                # Parameter 3 : error

forceconstant <real>      # Default = 1.0
name <word>      # Name of the pseudo contact shifts class to declare or to
                  # update
paramagneticcenter <atom-selection> # Paramagnetic center
tensorname <word>      # Name of the tensor
```

**<print-statement> ::=**

```
threshold <real>      # Print pseudo contact shifts violations greater than threshold
class <classname>      # Only violations of the specified pseudo contact shifts class
output <output>      # Output (file or screen)
format human      # Output format (human readable format)
```

**<classname> ::=**

**<word>|all**

**<output> ::=**

**<filename>|screen**

### g. Scsa command

**<scsa-statement> ::=**

```
class <class-statement> end      # Declare or update a class of chemical shift
                                     # anisotropy restraints

deltgrad <real>      # Define step value used to calculate some gradients numerically
                       # Default=0.00001

nrestraints <integer>      # Number of slots for CS anisotropic restraints to
                               # allocate in memory
                               # Default=2000

print <print-statement> end      # Print chemical shift anisotropy restraints

reset      # Erase the CSA assignment table
```

**<class-statement> ::=**

```
assign      <atom-selection> <atom-selection> <atom-selection>
              <atom-selection> <real> <real> <real> <integer>
              # Declares a new CSA restraint :
              # Parameters 1 and 2 : define the first axis of the plane
              # Parameters 3 and 4 : define a vector in the plane
              # Parameter 5 : value of the chemical shift restraint
              # Parameter 6 : error
              # Parameter 7 isotropic chemical shift value (in ppm)
              # Parameter 8 : type of potential to use (0 is the
              # classical harmonic potential, 1 is such that
              # mobility is allowed for

forceconstant <real>      # Default = 1.00

name <word>      # Name of the CSA restraint class to declare or to update

tensorname <word>      # Name of the tensor
```

**<print-statement> ::=**

```
threshold <real>      # Print CSA violations greater than the specified threshold

class <classname>      # Only violations of the specified CSA class are printed

output <output>      # Output (file or screen)

format module|human      # Output format (module format or readable format)
```

**<classname> ::=**  
**<word>|all**

**<output> ::=**  
**<filename>|screen**

### 3. Existing commands modifications

#### a. Flags command

The CNS flags command is used to include or exclude some terms of energy in the energy calculations. Because SculptorCNS patch add some new energetic terms it is necessary to modify the flags command like this:

**<flags-statement> ::=**  
include <energy-term> | \*  
exclude <energy-term> | \*

**<energy-term> ::=** bond|angl|dihe|impr|vdw|elec|pvdw|pele|harm|cdih|ncs|plan|xref|noe|dg|coup|carb|prot|onej|rama|andb|sani|dani|**srdc|srel|spar|scur|sdpl|scsa|stal|stre|stcu|stsu**

**srdc** : energy term associated with residual dipolar coupling in liquid crystal environment

**srel** : energy term associated with R2 / R1 type restraints

**spar** : energy term associated with residual dipolar coupling in a paramagnetic molecule

**scur** : energy term associated with curie restraints

**sdpl** : energy term associated with paramagnetic chemical shift

**scsa** : energy term associated with chemical shift anisotropy

**stal** : energy term associated with alignment tensor

**stre** : energy term associated with relaxation tensor

**stcu** : energy term associated with curie tensor

**stsu** : energy term associated with magnetic susceptibility tensor

#### b. Constants

The gyromagnetic constants used in the orientational restraints are coded in the same way as CNS constants.

# Output formats specifications for the SculptorCNS patch

## 1. Print subcommand output format for ALIGNMENT, CURIE , RELAXATION et SUSCEPTIBILITY commands of the STENSOR module

### a. print subcommand for ALIGNMENT and SUSCEPTIBILITY commands

- Output format with the data = angles option :

*tensorname*

scalxz    *scalar\_product*    xOz    *angle*

- Output format with the data = eigenvalues option :

*tensorname*

DA    *eigenvalues*

DR    *eigenvalues*

- Output format with the data = coordinates option :

*tensorname*

O    *x y z*

X    *x y z*

Z    *x y z*

OX1 *x y z*

X1    *x y z*

OZ1 *x y z*

Z1    *x y z*

- Output format with the data = all option:

*tensorname*

Coordinates

O    *x y z*

X    *x y z*

Z     *x y z*  
OX1   *x y z*  
X1     *x y z*  
OZ1   *x y z*  
Z1     *x y z*

#### Angles

scalxz     *scalar\_product*     xOz     *angle*

#### Eigenvalues

DA     *eigenvalues*  
DR     *eigenvalues*

### **b. Print subcommand for the CURIE command**

- Output format with the datacurie = khivalue option:

*moleculename*

khi     *khi\_value*

- Output format with the data = coordinates option :

*moleculename*

O     *x y z*  
X     *x y z*

- Output format with the data = all option:

*moleculename*

#### Coordinates

O     *x y z*  
X     *x y z*

#### Khi

khi     *khi\_value*



c. **print subcommand for the RELAXATION command**

- Output format with the data = angles option:

*tensorname*

scalxz    *scalar\_product*    xOz    *angle*

- Output format with the data = eigenvalues option:

*tensorname*

DXX    *eigenvalue*

DYY    *eigenvalue*

DZZ    *eigenvalue*

- Output format with the data = coordinates option:

*tensorname*

O    *x y z*

X    *x y z*

Z    *x y z*

OX1 *x y z*

X1    *x y z*

OY1 *x y z*

Y1    *x y z*

OZ1 *x y z*

Z1    *x y z*

- Output format with the data = all option:

*tensorname*

Coordinates

O    *x y z*

X    *x y z*

Z     *x y z*  
OX1   *x y z*  
X1     *x y z*  
OY1   *x y z*  
Y1     *x y z*  
OZ1   *x y z*  
Z1     *x y z*

#### Angles

scalxz     *scalar\_product*     xOz     *angle*

#### Eigenvalues

DXX     *eigenvalue*  
DYY     *eigenvalue*  
DZZ     *eigenvalue*

## **2. Print subcommand output format for SRDC, SRELAXATION, SPARA, SCURIE and SDPL commands**

### **a. Human output format**

#### Format description

Each lines of this file comprise a set of fields described below:

- Field 1 : identifier of the segment containing the first atom of the orientational restraint (this identifier is specified in the pdb file describing the molecule structure)
- Field 2 : number of the residue containing the first atom of the orientational restraint (this number is specified in the pdb file describing the molecule structure)
- Field 3 : name of the residue containing the first atom of the orientational restraint (this name is specified in the pdb file describing the molecule structure)
- Field 4 : name of the first atom of the orientational restraint (this name is specified in the pdb file describing the molecule structure)
- Field 5 : identifier of the segment containing the second atom of the orientational restraint (this identifier is specified in the pdb file describing the molecule structure)
- Field 6 : number of the residue containing the second atom of the orientational restraint (this number is specified in the pdb file describing the molecule structure)
- Field 7 : name of the residue containing the second atom of the orientational restraint (this name is specified in the pdb file describing the molecule structure)
- Field 8 : name of the second atom of the orientational restraint (this name is specified in the pdb file describing the molecule structure)
- Field 9 : calculated value of the orientational restraint
- Field 10 : experimental value of the relational restraint
- Field 11 : absolute value of the difference between the calculated value and the experimental value for the orientational restraint

Important comment for the print subcommand of the sdpl command:

For the sdpl restraints (pseudo contact shifts) the restraint value is measured between an atom and a paramagnetic center. For these restraints the first four Fields correspond to the atom of the restraint, and the next four Fields correspond to the atom corresponding to the class paramagnetic center.

When several orientational restraints classes are displayed, each class will be separated from the others by a line with the syntax described below:

*type\_class* class *class\_restraint\_name*

where *type\_class* will be equal to *srdc*, *srelaxation*, *sparamagnetic*, *scurie* or *sdpl*

o Example

```
srdc class JNH
      1   PHE  N          1   PHE  HN       7.149   7.149   0.000
      3   THR  N          3   THR  HN       9.375   9.375   0.000
      4   LYS  N          4   LYS  HN       7.928   7.928   0.000
      5   TRP  N          5   TRP  HN       1.169   1.169   0.000
      6   PHE  N          6   PHE  HN       5.975   5.975   0.000
srdc class JCAC
      1   PHE  CA         1   PHE  C         9.212   9.212   0.000
      2   PRO  CA         2   PRO  C        12.144  12.144   0.000
      3   THR  CA         3   THR  C       -19.273 -19.273   0.000
      4   LYS  CA         4   LYS  C       -10.991 -10.991   0.000
      5   TRP  CA         5   TRP  C        14.036  14.036   0.000
      6   PHE  CA         6   PHE  C       -25.215 -25.215   0.000
```

**b. Module output format**

This format corresponds to the input format of the Module program developed by the Blackledge laboratory. This format is only available for the print commands of *sdrc* and *spara* commands.

o Format description

A description of the Fields of each line of this file is given below:

- Field 1 : number of the residue containing the first atom of the orientational restraint (this number is specified in the pdb file describing the molecule structure)
- Field 2 : name of the first atom of the orientational restraint (this name is specified in the pdb file describing the molecule structure)
- Field 3 : number of the residue containing the second atom of the orientational restraint (this number is specified in the pdb file describing the molecule structure)

- Field 4 : name of the second atom of the orientational restraint (this name is specified in the pdb file describing the molecule structure)
- Field 5 : calculated value of the orientational restraint
- Field 6 : uncertainty on the restraint value
- Field 7 : S parameter value

- Example

```
3 CA 3 C      -2.127479  0.195  1.00
4 CA 4 C      1.182786  0.349  1.00
5 CA 5 C      3.222452  0.245  1.00
```

### 3. print subcommand output format for the SCSA command

#### a. human output format

- Format description

Each line of this file comprise a set of Fields described below :

- Field 1 : identifier of the segment containing the first atom of the orientational restraint (this identifier is specified in the pdb file describing the molecule structure)
- Field 2 : number of the residue containing the first atom of the orientational restraint (this number is specified in the pdb file describing the molecule structure)
- Field 3 : name of the residue containing the first atom of the orientational restraint (this name is specified in the pdb file describing the molecule structure)
- Field 4 : name of the first atom of the orientational restraint (this name is specified in the pdb file describing the molecule structure)
- Field 5 : identifier of the segment containing the second atom of the orientational restraint (this identifier is specified in the pdb file describing the molecule structure)
- Field 6 : number of the residue containing the second atom of the orientational restraint (this number is specified in the pdb file describing the molecule structure)

- Field 7 : name of the residue containing the second atom of the orientational restraint (this name is specified in the pdb file describing the molecule structure)
- Field 8 : name of the second atom of the orientational restraint (this name is specified in the pdb file describing the molecule structure)
- Field 9 : identifier of the segment containing the third atom of the orientational restraint (this identifier is specified in the pdb file describing the molecule structure)
- Field 10 : number of the residue containing the third atom of the orientational restraint (this number is specified in the pdb file describing the molecule structure)
- Field 11 : name of the residue containing the third atom of the orientational restraint (this name is specified in the pdb file describing the molecule structure)
- Field 12 : name of the third atom of the orientational restraint (this name is specified in the pdb file describing the molecule structure)
- Field 13 : identifier of the segment containing the fourth atom of the orientational restraint (this identifier is specified in the pdb file describing the molecule structure)
- Field 14 : number of the residue containing the fourth atom of the orientational restraint (this number is specified in the pdb file describing the molecule structure)
- Field 15 : name of the residue containing the fourth atom of the orientational restraint (this name is specified in the pdb file describing the molecule structure)
- Field 16 : name of the fourth atom of the orientational restraint (this name is specified in the pdb file describing the molecule structure)
- Field 17 : calculated value of the orientational restraint
- Field 18 : experimental value of the relational restraint
- Field 19 : absolute value of the difference between the calculated value and the experimental value for the orientational restraint

When several orientational restraints classes are displayed, each class will be separated from the others by a line with the syntax described below :

*type\_class class class\_restraint\_name*

where *type\_class* will be equal to *sca*

## **b. module output format**

This format corresponds to the input format of the Module program developed by the Blackledge laboratory.

○ Format description

A description of the line Fields of this file is given below :

- Field 1 : number of the residue containing the first atom of the orientational restraint (this number is specified in the pdb file describing the molecule structure)
- Field 2 : name of the first atom of the orientational restraint (this name is specified in the pdb file describing the molecule structure)
- Field 3 : number of the residue containing the second atom of the orientational restraint (this number is specified in the pdb file describing the molecule structure)
- Field 4 : name of the second atom of the orientational restraint (this name is specified in the pdb file describing the molecule structure)
- Field 5 : number of the residue containing the third atom of the orientational restraint (this number is specified in the pdb file describing the molecule structure)
- Field 6 : name of the third atom of the orientational restraint (this name is specified in the pdb file describing the molecule structure)
- Field 7 : number of the residue containing the fourth atom of the orientational restraint (this number is specified in the pdb file describing the molecule structure)
- Field 8 : name of the fourth atom of the orientational restraint (this name is specified in the pdb file describing the molecule structure)
- Field 5 : number of the residue containing the third atom of the orientational restraint (this number is specified in the pdb file describing the molecule structure)
- Field 9 : calculated value of the orientational restraint
- Field 10 : uncertainty on the restraint value

○ Example

```
3 CA 3 C 5 N 6 H      -2.127479 0.195
4 CA 4 C 7 N 9 H      1.182786 0.349
5 CA 5 C 4 N 10 H     3.222452 0.245
```