RADDOSE−3D Command Reference

version 1.2 – 18 December 2014

Contents

RADDOSE-3D can take input from one or more files and/or from standard input (STDIN). Any input will be processed by the `InputParser` class and the `RADDOSE-3D` ANTLR parser. This section describes the syntax of accepted input. Advanced users of `RADDOSE-3D` can create their own input method that need not rely on the `InputParser` class or the `RADDOSE-3D` ANTLR parser. This feature will not be covered in this reference.

The simplest use case of `RADDOSE-3D` will involve only one file describing the entire experiment. In some instances it may be desired to split up the input into a number of files, e.g. one file describing the crystal, one automatically updated file describing the current beam on the beamline, and one file chosen from a set of possible wedge strategies. Each file can contain an arbitrary number (including none) of `Crystal`, `Beam` and `Wedge` block (henceforth called *blocks*). However, splitting up blocks across multiple files is not allowed.

The parser will read the input sequentially, and, when multiple sources are given, one source after the other in the specified order. While the parser may accept `Crystal`, `Beam` and `Wedge` blocks in any order, the exposure of a wedge can only take place if both the crystal and the beam have been set either in an earlier file or before the `Wedge` block within the same file.

1 General syntax considerations

Any keywords specified below are case-insensitive. Upper (**CRYSTAL**), lower (**crystal**) and mixed case (**CrYsTaL**) are equivalent.

The characters **#**, **!** and the character sequence **//** denote the start of a comment. Any text from that position until the end of the current line is ignored.

Tabular and newline characters are treated as white space. They can therefore by freely used to format the file for increased readability.

The order of statements within a `Crystal`, `Beam` and `Wedge` block generally is not relevant. There are two exceptions to this rule: The leading keyword (**CRYSTAL**, **BEAM**, **WEDGE**) **must** be the first keyword of the block. If a keyword is repeated within the same block, then the latter will always override the former.

Every block must be self-contained, e.g. the energy set for the previous `Beam` is not remembered when setting up the following `Beam`, and must be repeated.

Numeric values can be given in scientific notation ($2.0e2 = 2e+2 = 200$), negative values may not have a space between the sign ('-') and the value ($-1.9e-1 = -.19 = -0.19$).

2 Crystal block

A `Crystal` block must begin with the keyword **CRYSTAL**. At least the **TYPE** and **DIMENSION** must be specified. Depending on the chosen **TYPE** further declarations may be required.

2.1 TYPE

With the keyword **TYPE** the underlying crystal implementation is chosen. Currently three distinct crystal implementations exist:

TYPE CUBOID defines a solid crystal with a cuboid shape.

TYPE SPHERICAL defines a solid crystal with a spherical shape.

TYPE POLYHEDRON defines an arbitrary crystal shape as a polyhedron. The wire frame type (**WIREFRAMETYPE** keyword - section section 2.2) and file containing the model (**MODELFILE** keyword - section section 2.3) must be specified.

2.2 WIREFRAMETYPE

WIREFRAMETYPE Specifies the type of wire frame model used to model a crystal of polyhedron type. Currently only .obj (geometry definition) files can be used:

WIREFRAMETYPE OBJ specifies the wire frame model to be in .obj format.

2.3 MODELFILE

MODELFILE Specifies the location of the file that contains the wire frame model of the polyhedron crystal. Currently only .obj (geometry definition) files can be read. The models and the .obj files can be generated using the free and open source 3D animation software **BLENDER**. (NOTE: if you export a *Wavefront (.obj)* file in **BLENDER**, then select the option *Triangulate Faces* before you export. RADDOSE-3D works with triangular faces for the polygons.):

MODELFILE *SOMEMODELFILE.OBJ* specifies the location of the .obj file to be imported.

2.4 DIMENSION

DIMENSION specifies the size of the crystal. Dimensions are given in micrometres (µm). The keyword **DIMENSION** can take either one or three parameters:

DIMENSION *D* with a single number (see section 1) as parameter is used for specifying the crystal dimensions for spherical crystals. The parameter sets the crystal diameter. This syntax cannot be used for cuboid crystals.

DIMENSION *X Y Z* with three numbers as parameters X, Y and Z is used to set the dimensions for cuboid crystals (**TYPE CUBOID**). X defines the length of the crystal orthogonal to both the beam and the goniometer at $L=P=0$, (see below) Y defines the length along the goniometer axis at $L=P=0$ and Z defines the length along the beam axis.

If three parameters are given for a spherical crystal (**TYPE SPHERICAL**) the value for X sets the diameter of the crystal while the values of Y and Z are ignored.

2.5 PIXELSPERMICRON

PIXELSPERMICRON *F* specifies the resolution of the voxel grid used to represent the crystal in voxels/µm. Defaults to 0.5 voxels/µm.

2.6 ANGLEP

ANGLEP *F* sets the angle in the plane of the loop between the crystal Y axis and the goniometer axis. The angle is to be given in degrees, but without the degree symbol (°). The default P ('plane') angle is 0°.

The rotation angle to be applied to the crystal in the plane of the loop (right handed rotation about Z axis applied to all voxels, as shown in figure 1).
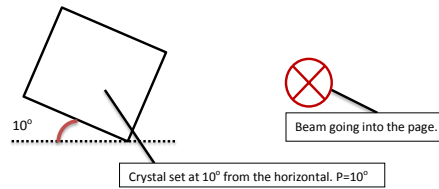

10°

Beam going into the page.

Crystal set at 10° from the horizontal. P=10°

Figure 1: Schematic of **ANGLEP**. Figure courtesy of John Bremridge.

2.7 ANGLEL

ANGLEL *F* sets the loop angle between the plane of the crystal loop and the goniometer axis. The angle is to be given in degrees, but without the degree symbol (°). The default L ('loop') angle is 0°.

The rotation angle to be applied to the angle of the crystal in the loop (right handed rotation about X axis applied to all voxels, as shown in figure 2).
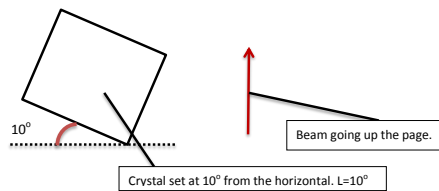

10°

Beam going up the page.

Crystal set at 10° from the horizontal. L=10°

Figure 2: Schematic of **ANGLEL**. Figure courtesy of John Bremridge.

2.8 ABSCOEFCALC

This keyword specifies whether the program should use average absorption and attenuation coefficients, or whether it should calculate them from input crystal parameters.

ABSCOEFCALC AVERAGE
ABSCOEFCALC DUMMY
    These two commands are equivalent. Each will cause RADDOSE-3D to assume an absorption coefficient of 0.237 mm^{-1} and an attenuation coefficient of 0.281 mm^{-1}. These values are representative of an average crystal at an incident X-ray beam energy of 12.4 keV (1Å). Please see Section 3 in the main paper for more details. Crystal composition keywords will have no effect.

ABSCOEFCALC RD
ABSCOEFCALC RDV2
    These two commands are equivalent. RADDOSE-3D will call a previous version of RADDOSE to estimate absorption and attenuation coefficients.

ABSCOEFCALC RDV3
    This command will use the current RADDOSE-3D code to calculate the absorption and attenuation coefficients using the crystal composition specified by the user. (NOTE: If the crystal composition is specified then the **RDV3** keyword is preferred over the other **ABSCOEFCALC** keywords).
    The composition of the crystal has to be described using the keywords **UNITCELL**, **NUMMONOMERS**, **NUMRESIDUES**, **NUMRNA**, **NUMDNA**, **PROTEINHEAVYATOMS**, **SOLVENTHEAVYCONC** and **SOLVENTFRACTION**. The use of these keywords is described in the sections 2.10–2.17 below.

ABSCOEFCALC EXP
    This command should be used if the crystal composition from a Protein Data Bank (PDB) entry is to be used instead of a user specied crystal composition. The current RADDOSE-3D code is then used to calculate the absorption and attenuation coefficients using the crystal composition from the PDB entry. The PDB entry must be specified using the (**PDB** keyword - section section 2.9).

2.9 PDB

PDB *CODE*
    Where **CODE** is the four letter code of the PDB entry to be downloaded. From the PDB entry RADDOSE-3D will calculate **UNITCELL**, **NUMMONOMERS**, **NUMRESIDUES**, **NUMRNA**, **NUMDNA**, **PROTEINHEAVYATOMS** and **SOLVENTFRACTION**, and hence these keywords are not needed. It will NOT calculate **SOLVENTHEAVYCONC** so if there are any heavy atoms in the solvent then they will have to be specified.

2.10 UNITCELL

This keyword only has an effect when the absorption and attenuation coefficients are estimated using a legacy version of RADDOSE (see section 2.8).

UNITCELL *A B C*
UNITCELL *A B C* α β γ
    Dimensions and angles of the unit cell a, b, c, α, β, γ

The first three numbers specify the unit cell size in Angstroms. The second three numbers optionally specify the unit cell angles alpha, beta and gamma.

The (optional) angles are to be given in degrees, but without the degree symbol (°). If no angles are specified `RADDOSE-3D` assumes default angles of 90°.

2.11 **NUMMONOMERS**

This keyword only has an effect when the absorption and attenuation coefficients are estimated using a legacy version of RADDOSE (see section 2.8).

NUMMONOMERS *I* specifies the number of monomers in the unit cell. Only integer numbers *I* should be used. This number should not be confused with the number of monomers in the asymmetric unit.

2.12 **NUMRESIDUES**

This keyword only has an effect when the absorption and attenuation coefficients are estimated using a legacy version of RADDOSE (see section 2.8).

NUMRESIDUES *I* specifies the number of amino acid residues per monomer. Only integer numbers *I* should be used. Using this keyword the number and types of atoms are calculated according to the formula

$$\text{amino acid} = 5C + 1.35N + 1.5O + 8H$$

Sulfur atoms, e.g. from CYS and MET residues, should be added explicitly with the **PROTEINHEAVYATOMS** keyword.

The default value for *I* is 0.

2.13 **NUMRNA**

This keyword only has an effect when the absorption and attenuation coefficients are estimated using a legacy version of RADDOSE (see section 2.8).

NUMRNA *I* specifies the number of RNA nucleotides per monomer. Only integer numbers *I* should be used. Using this keyword the number and types of atoms are calculated assuming an average nucleotide content defined as

$$\text{mean nucleotide} = 11.25H + 9.5C + 3.75N + 7O + 1P$$

If a more accurate estimate is required, individual atoms may be entered explicitly with the **PROTEINHEAVYATOMS** keyword.

The default value for *I* is 0.

2.14 **NUMDNA**

This keyword only has an effect when the absorption and attenuation coefficients are estimated using a legacy version of RADDOSE (see section 2.8).

NUMDNA *I*

specifies the number of DNA deoxynucleotides per monomer. Only integer numbers *I* should be used. Using this keyword the number and types of atoms are calculated assuming an average deoxynucleotide content defined as

$$\text{mean nucleotide} = 11.75H + 9.75C + 4N + 6O + 1P$$

If a more accurate estimate is required, individual atoms may be entered explicitly with the **PROTEINHEAVYATOMS** keyword.

The default value for *I* is 0.

2.15   **PROTEINHEAVYATOMS**

This keyword only has an effect when the absorption and attenuation coefficients are estimated using a legacy version of RADDOSE (see section 2.8).

PROTEINHEAVYATOMS *El I (El I (El I ..))* defines a list of atoms to add to the protein part of the absorption. Each species is defined by a two character string *El* for the elemental symbol, and an integer number *I* of atoms of that species per monomer.

The command **PROTEINHEAVYATOMS S 10 Se 2** would add 10 sulfur and 2 selenium atoms per monomer.

2.16   **SOLVENTHEAVYCONC**

This keyword only has an effect when the absorption and attenuation coefficients are estimated using a legacy version of RADDOSE (see section 2.8).

SOLVENTHEAVYCONC *El I (El I (El I ..))* defines the concentration of elements (not including water) in the solvent in millimoles per litre. Oxygen and lighter elements should not be specified.

The command **SOLVENTHEAVYCONC Na 1000 Cl 1000** specifies 1M sodium chloride in the solvent.

2.17   **SOLVENTFRACTION**

This keyword only has an effect when the absorption and attenuation coefficients are estimated using a legacy version of RADDOSE (see section 2.8).

SOLVENTFRACTION *F*

The fraction of the unit cell that is occupied by solvent. If not given explicitly, this value is estimated from **NUMRESIDUES**, **NUMRNA** and **NUMDNA** using 1.35 g/ml for protein, and 2.0 g/ml for RNA and DNA.

3   **Beam** block

A Beam block must begin with the keyword **BEAM**. At least the **TYPE** must be specified. Depending on the chosen **TYPE**, further declarations may be required.

3.1   **TYPE**

With the keyword **TYPE**, the underlying beam implementation is chosen. Currently three distinct beam implementations exist:

TYPE TOPHAT defines a beam with uniform flux.

TYPE GAUSSIAN defines a beam with a 2-dimensional Gaussian flux profile. The full-width half-maximum must be specified with the **FWHM** keyword (see section 3.3).

TYPE EXPERIMENTALPGM defines an arbitrary beam profile as defined in a given portable graymap file (.pgm) specified with the **FILE** keyword (see section 3.5). The pixel size of the image file also has to be specified with the **PIXELSIZE** keyword (see section 3.6).

3.2 FLUX

FLUX *F* specifies the total beam flux in photons per second. The flux parameter F can be specified in scientific notation (e.g. **1.3e12**).

3.3 FWHM

FWHM *X Y*

The FWHM of the beam (vertical), (horizontal). Not needed if a Top-Hat beam is used. This defines the X and Y FWHM of the beam respectively in the RADDOSE coordinate system.

3.4 ENERGY

ENERGY *F*
ENERGY *F* KEV

specifies the incident photon energy in keV. The optional keyword **KEV** can be appended for human readability of the input file.

3.5 FILE

FILE *SOMEPGMFILE.PGM*

specifies the location of the .pgm file that describes the beam profile. NOTE: the .pgm file should be in ASCII format. The pixel size of the image file also has to be specified with the **PIXELSIZE** keyword (see section 3.6).

3.6 PIXELSIZE

PIXELSIZE *X Y*.

X and ***Y*** specify the horizontal and vertical size of the pixels in micrometers in the .pgm file respectively.

3.7 COLLIMATION

COLLIMATION RECTANGULAR *X Y*

specifies the horizontal and vertical collimation of the beam. Delimits where the beam has non-zero intensity. This is defined by the slits. For an uncollimated Gaussian beam, set to $\approx 3\times$ FWHM.

4 Wedge block

A Wedge block must begin with the keyword **WEDGE**.

WEDGE *A B*

A and B define the start and end angle of the rotation in degrees (°). At 0° the front face of the crystal (X-Y plane) is normal to the beam. Rotation is right handed about the Y axis, as shown in figure 3).
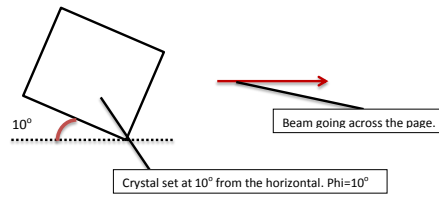
Figure 3: Schematic of angles for **WEDGE**. Figure courtesy of John Bremridge.

4.1 EXPOSURETIME

EXPOSURETIME *F*

specifies the total exposure time for this wedge in seconds.

4.2 ANGULARRESOLUTION

ANGULARRESOLUTION *F*

specifies the angular step size used for wedge iterations in degrees (°). Defaults to 2°.

4.3 STARTOFFSET

STARTOFFSET *X Y Z*

offset translation in µm applied to the crystal relative to the origin (defined as the intersection of the beam and the aligned goniometer axis) for the starting position of the wedge. Defaults to 0 0 0.

4.4 TRANSLATEPERDEGREE

TRANSLATEPERDEGREE *X Y Z*

translation of the goniometer during exposure in µm/° for helical scanning, leading to improvements in dose distribution. Defaults to 0 0 0.

4.5 ROTAXBEAMOFFSET

ROTAXBEAMOFFSET *F*

the offset in µm along X (vertical in most set-ups) between the beam axis and the rotation axis. Used to create 'offset' scanning for improvements in dose distribution. Defaults to 0 µm.

5 **RADDOSE−3D** for SAXS

RADDOSE−3D can be used to give dose estimates for SAXS experiments. Given the protein concentration all that is required is to calculate the number of molecules in a given volume. This is equivalent to finding the number of monomers in a unit cell volume for a crystal. An example of the required calculation is given for (mouse) glucose isomerase:

Glucose isomerase (mouse) has 558 amino acids. The atomic mass for each amino acid can be used to accurately find the total atomic mass but for demonstration purposes we will use the fact that an amino acid is roughly 110 daltons. This gives a total molecular mass for the protein as $558 \times 110 = 61380$ daltons or $61380\,grams/mole$. Using the protein concentration in the SAXS experiment, here we will use a concentration of $2\,grams/litre$ the molarity of the solution can be calculated:

$$molarity = \frac{2\,grams/mole}{61380\,grams/litre} = 3.26 \times 10^{-5}\,M \tag{1}$$

Now it is necessary to choose the dimensions for an arbitrary volume such that the volume will contain at least 1 molecule. For simplicity a 1000Å^3 volume is chosen. Converting this volume to litres yields $1000\text{Å}^3 = 1 \times 10^{-18}\,L$. Multiplying this volume by the molarity of the solution calculate earlier gives the number of moles - $3.26 \times 10^{-5}\,M \times 1 \times 10^{-18}\,L = 3.26 \times 10^{-23}moles$.

Finally the number of molecules in the chosen volume is given by multiplying the number of moles by Avogadro's number ($\approx 6.02 \times 10^{23}\,moles^{-1}$). Hence in this example, the number of molecules in the chosen volume is $3.26 \times 10^{-23} \times 6.02 \times 10^{23} = 19.63$. Since the input for the number of monomers is required to be an integer this value is rounded to 20 molecules per unit chose volume.

Therefore in the RADDOSE−3D input file the input line for the **NUMMONOMERS** would be "**NUMMONOMERS** 20" and the input for the unit cell would be **UNITCELL** would be "**UNITCELL** 1000 1000 1000".

Generally in a SAXS experiment the protein solution is passed through a cylindrical capillary. This shape can only be explicitly modelled in RADDOSE-3D as a **POLYHEDRON** object. This can be modelled and exported using **BLENDER**.