

NAME

psic – PSI-circle geometry modes

DESCRIPTION

The *psic* version of *spec* operates a type of six-circle diffractometer that has two circles to orient the detector and four circles to orient the sample. Unlike the traditional six-circle configuration, the sample circles are not coupled to the motions of the detector circles.

The angle calculations and operating modes for the psi-circle diffractometer are based on those presented in a paper by H. You of Argonne National Lab in *J. Appl. Cryst.*, **32**, 614-623 (1999). (See <http://journals.iucr.org/j/issues/1999/04/00/hn0093>)

CONVENTIONS

The psi-circle motor names and mnemonics are as follows:

- Delta del – Rotates detector arm
- Eta eta – Holds chi and phi circles
- Chi chi – Sample tilt
- Phi phi – Sample rotation
- Mu mu – Rotates sample circles around vertical
- Nu nu – Rotates detector arm around vertical axis
- K-eta keta – For *kappa* geometry
- Kappa kap – For *kappa* geometry
- K-phi kphi – For *kappa* geometry
- Det Rot daz – Optional azimuthal detector slit rotation

The mnemonics are compiled into the C code, so must match those given above, while the motor names can be anything.

The keta, kap and kphi motors are associated with the kappa variation of the psi-circle diffractometer. In such a configuration, these motors are the real motors, while the eta, chi and phi are pseudo-motors (indicated by controller-type NONE in the configuration file).

The last motor, daz, corresponds to an optional azimuthal rotation of the detector used to maintain a specific angular acceptance. See the section on detector rotation below.

PSI-CIRCLE MODES

The psi-circle modes represent various constraints that can be made on the extra three degrees of freedom available in the transformation from reciprocal space to diffractometer angles. Use the *setmode* macro to select a psi-circle mode.

The psi-circle geometry code uses a different formalism for specifying operating modes than previous *spec* geometries. Instead of just one value to select a mode, up to five small numbers select the mode. The following table shows what the five numbers represent.

	<i>g_mode1</i>	<i>g_mode2</i>	<i>g_mode3</i>	<i>g_mode4</i>	<i>g_mode5</i>	
0omega-fixed....		X	0
1	Delta-fixed	Alpha=BetaEta-fixed.....			1
2	Nu-fixed	Alpha-fixedMu-fixed.....			2
3	Qaz-fixed	Beta-fixedChi-fixed.....			3
4	Naz-fixed	Psi-fixedPhi-fixed.....			4
5	Zone	X	Eta=Del/2	X	X	5
6	X	X	Mu=Nu/2	X	X	6

The first row indicates the names used to refer to the five mode parameters. The first and last columns are the possible values for those parameters. The X positions are not used. Some modes allow zero for *g_mode1* and/or *g_mode2*.

In general, the first two columns are related to detector angles and pseudo-angles, while the last three columns are used to set sample circles. The implemented modes are:

`setmode 5` – Selects *zone* mode. In *zone* mode, the scattering vector is confined to a plane specified by two reciprocal lattice vectors. Values for `chi` and `phi` are chosen to keep Q in that plane. See the `sz`, `cz` and `mz` macros below.

`setmode 0 0 s1 s2 s3` – Selects a mode where three sample circles are fixed, as specified by the values for `s1` `s2` and `s3`, which must all be different.

`setmode d1 d2 s1` – Requires nonzero values for the first three mode parameters. The `s1` mode can be any sample circle, or the special values 5 or 6 to set $\eta = \delta/2$ or $\mu = \nu/2$, respectively.

`setmode 0 d2 s1 s2` – Requires a nonzero `g_mode2` if `g_mode1` is zero. (Currently, the `mu-fixed` + `phi-fixed` modes aren't working.)

`setmode d1 0 s1 s2` – Requires a nonzero `g_mode1` if `g_mode2` is zero. (Currently, the `eta-fixed` + `chi-fixed`, the `eta-fixed` + `phi-fixed` and the `naz-fixed` modes aren't working.)

`setmode d1 0 0 0` – Selects `omega-fixed` mode, where `omega` is defined as the angle of the scattering vector Q with respect to the plane of the `chi` circle. Requires a nonzero `g_mode1`.

CUT POINTS

Cut points affect the direction the diffractometer circles turn to get from one position to the next. For example, if a cut point is at zero, the corresponding circle will only move through angles of 0 to 360 degrees. Thus, to get from 355 ($=-5$) to 5 degrees, the circle will turn 350 degrees. If a cut point is at -180 , the circle will move through angles from -180 to 180. Thus the same motion from -5 to 5 will require only 10 degrees of movement. Use the `cuts` macro to select cut points for all the diffractometer angles.

SECTORS

The calculation of `HKL` from the six diffractometer circles involves sums and products of the sines and the cosines of each angle. If one takes each angle x and substitutes $-x$, $180 + x$ and $180 - x$ (which only affects the sign of the sines and cosines), one can find sixteen combinations where the sign changes cancel out, resulting in identical `HKL` values for any set of diffractometer angles. Some of these alternative angle settings may be better suited to constraints of a particular experiment or experimental configuration. For historical reasons, `spec` refers to these alternative angle settings as *sectors*.

If one or more of the diffractometer angles is set to a multiple of $\pi/2$, it is possible that additional combinations of angles, other than the sixteen combinations mentioned above, become available. As of `spec` release 5.06.04-1, the `psic` geometry code does an exhaustive search through all possible transformations in order to find any additional available sectors. In addition, besides ruling out the transformations that violate the frozen-angle constraints or wind up at a different `HKL`, the new code also rules out angle combinations that are outside the software motor limits.

`spec` provides two methods of choosing sectors. One can have one of the 16 legacy transformations always performed (although the resulting angles may well violate the fixed conditions associated with the current geometry mode). Or one can have `spec` run through all the transformations, and after rejecting those that violate the frozen conditions or motor limits, select the one that best conforms to a built-in ranking scheme.

The sixteen transformations that always result in identical `HKL` are as follows:

	del	eta	chi	phi	nu	mu
1
2	.	-x	180-x	180+x	.	180+x
3	.	180-x	180+x	.	.	180+x
4	.	180+x	-x	180+x	.	.
5	-x	.	-x	180+x	-x	180-x
6	-x	-x	180+x	.	-x	-x
7	-x	180-x	180-x	180+x	-x	-x
8	-x	180+x	.	.	-x	180-x
9	180-x	.	.	.	180+x	.
10	180-x	-x	180-x	180+x	180+x	180+x
11	180-x	180-x	180+x	.	180+x	180+x
12	180-x	180+x	-x	180+x	180+x	.
13	180+x	.	-x	180+x	180-x	180-x
14	180+x	-x	180+x	.	180-x	-x
15	180+x	180-x	180-x	180+x	180-x	-x
16	180+x	180+x	.	.	180-x	180-x

Note, for sector transformations on the detector circles when used with the naz- and qaz-fixed modes and the constraints $\text{eta}=\text{del}/2$ or $\text{mu}=\text{nu}/2$, the eta or mu angles will be modified to maintain the constraint and the remaining sample circles will be recalculated to keep the same reciprocal space coordinates (as of spec release 5.05.04-01).

To have one of the above transformations always performed, use the `sector` macro (or assign a value to the `g_sect` symbol). Set the sector to zero to have `spec` use the ranking method and examine a greater number of possible sector transformations.

So far, three ranking schemes are implemented. The first is for a pseudo-vertical orientation, the second for a pseudo-horizontal orientation and the third to accommodate users at ESRF's ID01 beamline. Each ranking a

1 Pseudo-vert:	0 <= del	chi < 180	-90 <= nu	mu eta < 90
2 Pseudo-horz:	0 <= nu	chi < 180	-90 <= del	mu eta < 90
3 ESRF ID1:	0 <= del	nu chi < 180	-90 <= mu	eta < 90

For the kappa configuration, chi is kept between two times the kappa angle (`g_kappa`) and that angle minus 180 degrees.

For ranking sector transformations, `spec` currently uses a somewhat *ad hoc* weighting scheme. Weights for the del, nu, mu, eta and chi transformations are assigned values of 16, 8, 4, 2 and 1, respectively, if the angle falls in the preferred range. The transformation with the highest ranking (that doesn't violate the constraints of the geometry mode) is then selected.

The ranking scheme is selected by assigning values to `g_prefer`. A value of one selects the vertical orientation ranking scheme, a value of two selects the horizontal ranking scheme and a value of three selects the ESRF ID1 scheme.

Users are encouraged to suggest additional ranking schemes that meet particular needs, and CSS will endeavor to include them in future releases.

If both `g_sect` and `g_prefer` are zero, then the sector 1 angles (no transformations) are used.

See the `sectors` and the `ShowSect` macros below for options on displaying the possible transformations.

GLOBALS AND MACROS

Q[] – a built-in array which holds the following psic-circle parameters:

```

def H 'Q[0]' – Reciprocal space coordinate.
def K 'Q[1]' – Reciprocal space coordinate.
def L 'Q[2]' – Reciprocal space coordinate.
def LAMBDA 'Q[3]' – Wavelength of X rays.
def ALPHA 'Q[4]' – The angle between the reference vector and the x-z plane.
    Also, the incident angle when the reference vector is the surface normal.
def BETA 'Q[5]' – Exit angle when the reference vector is the surface normal.
def OMEGA 'Q[6]' – The angle between Q and the plane of the chi circle.
def TTH 'Q[7]' – Twice the Bragg angle. Also, the angle between Q and the x-z
    plane.
def PSI 'Q[8]' – Azimuthal angle of reference vector with respect to Q and the
    scattering plane.
def TAU 'Q[9]' – Longitudinal angle of reference vector with respect to Q and the
    scattering plane.
def QAZ 'Q[10]' – Azimuthal angle of Q, i.e., the angle between Q and the y-z
    plane.
def NAZ 'Q[11]' – Azimuthal angle of reference vector, i.e., the angle between the
    reference vector and the y-z plane.
def SIGMA_AZ 'Q[12]' – Angle used in alternate method of specifying reference
    vector.
def TAU_AZ 'Q[13]' – Angle used in alternate method of specifying reference vec-
    tor.
def F_ALPHA 'Q[14]' – Frozen value of ALPHA for alpha-fixed mode.
def F_BETA 'Q[15]' – Frozen value of BETA for beta-fixed mode.
def F_OMEGA 'Q[16]' – Frozen value of OMEGA for omega-fixed mode.
def F_PSI 'Q[17]' – Frozen value of PSI for psi-fixed mode.
def F_NAZ 'Q[18]' – Frozen value of NAZ for naz-fixed mode.
def F_QAZ 'Q[19]' – Frozen value of QAZ for qaz-fixed mode.
def F_DEL 'Q[20]' – Frozen value of A[del] for delta-fixed mode.
def F_ETA 'Q[21]' – Frozen value of A[eta] for eta-fixed mode.
def F_CHI 'Q[22]' – Frozen value of A[chi] for chi-fixed mode.
def F_PHI 'Q[23]' – Frozen value of A[phi] for phi-fixed mode.
def F_MU 'Q[24]' – Frozen value of A[mu] for mu-fixed mode.
def F_NU 'Q[25]' – Frozen value of A[nu] for nu-fixed mode.
def F_CHI_Z 'Q[26]' – Value calculated for A[chi] in zone mode.
def F_PHI_Z 'Q[27]' – Value calculated for A[phi] in zone mode.
def CUT_DEL 'Q[28]' – Cut point for del circle.
def CUT_ETA 'Q[29]' – Cut point for eta circle.
def CUT_CHI 'Q[30]' – Cut point for chi circle.
def CUT_PHI 'Q[31]' – Cut point for phi circle.
def CUT_MU 'Q[32]' – Cut point for mu circle.
def CUT_NU 'Q[33]' – Cut point for nu circle.
def CUT_KETA 'Q[34]' – Cut point for keta circle.
def CUT_KAP 'Q[35]' – Cut point for kap circle.
def CUT_KPHI 'Q[36]' – Cut point for kphi circle.

```

G[] – a built-in array which holds the following psic-circle parameters:

```

def g_prefer 'G[0]' – Holds sector preference value.
def g_sect 'G[1]' – Holds sector mode.
def g_frz 'G[2]' – Nonzero when frozen mode is on.

```

```

def g_haz 'G[3]' - H of azimuthal reference vector.
def g_kaz 'G[4]' - K of azimuthal reference vector.
def g_laz 'G[5]' - L of azimuthal reference vector.
def g_zh0 'G[6]' - H of first zone-mode vector.
def g_zk0 'G[7]' - K of first zone-mode vector.
def g_zl0 'G[8]' - L of first zone-mode vector.
def g_zh1 'G[9]' - H of second zone-mode vector.
def g_zk1 'G[10]' - K of second zone-mode vector.
def g_zl1 'G[11]' - L of second zone-mode vector.
def g_kappa 'G[12]' - Kappa tilt angle.
def g_sigtau 'G[15]' - Nonzero if using angles SIGMA_AZ and TAU_AZ to specify
    azimuthal reference vector.
def g_mode1 'G[16]' - Holds a geometry mode parameter.
def g_mode2 'G[17]' - Holds a geometry mode parameter.
def g_mode3 'G[18]' - Holds a geometry mode parameter.
def g_mode4 'G[19]' - Holds a geometry mode parameter.
def g_mode5 'G[20]' - Holds a geometry mode parameter.
def g_use_daz 'G[21]' - If set and the detector-rotation motor daz is configured,
    enables detector-rotation calculations.

```

ca *H K L* - Calculates and display the motor positions and values for the pseudo-angles at the given reciprocal space position.

pa - Displays most of the geometry mode and orientation matrix parameters.

freeze [*values ...*] - Sets *frozen* mode and assigns frozen values using either the current angle settings or the optional arguments. In frozen mode, angles and pseudo-angles fixed by the selected geometry mode are assigned their frozen values before calculating motor positions for the non-fixed angles. If present, the arguments are assigned to the frozen parameters in the same order that the *g_mode* parameters fix values. With no arguments, the current positions of the fixed angles and/or pseudo-angles are used in the assignment.

unfreeze - Turns off *frozen* mode

setmode [*modes ...*] - Sets the up-to-five geometry mode parameters. If no arguments are given, the macro displays a table of possible modes, the current mode and queries for a new mode.

cuts [*angle cut*][*all_cuts ...*] - Sets the cut points. A single motor mnemonic and one cut point can be given as arguments, or all six (or nine in kappa-mode) cut points may be given as arguments in the order δ η χ ϕ μ ν (keta kappa kphi). With no arguments, the macro queries for all cut points.

sigtau [1|0] - Turns on or off the mode where the azimuthal reference vector is specified by two angles, conventionally referred to as σ and τ , or the mode where the azimuthal reference vector is specified by *HKL* values. A value of one enables the first of the above. The default method is the second of the above.

setaz [*sigma tau*][*H K L*] - Defines the azimuthal reference vector according to the method selected by the *sigtau* macro.

sector [*which*] - Selects the current sector transformation. Values from 1 to 16 correspond to specific transformations. A value of zero allows the ranking and preference scheme. See the Sectors discussion above.

prefer [*how*] - Selects the preferred sector calculation. A value of zero turns off the preference ranking. See the Sectors discussion above.

sectors [*H K L*] – Displays the possible motor positions corresponding to the different allowed sectors for a particular reciprocal space position. If none is given as an argument, the current HKL values are used.

ShowSect [*arg*] – Displays the possible sectors corresponding to the current HKL values. The bits in *arg* have the following meaning:

- 0x01 - Show the basic 16 sectors
- 0x02 - Show all 4096 sectors
- 0x04 - Show values for ω , φ and ν
- 0x08 - If kappa mode, show both real and pseudo angles
- 0x10 - Show ruled-out sectors
- 0x20 - Show sectors ruled out by limits

Sectors with duplicate values are never displayed. When bit 0x10 is set, sectors that are ruled out are displayed with a code that indicates the reason, as follows:

- F - Violates Frozen angle
- o - Violates a frozen pseudoangle Ω , φ or ν
- Q - Produces different HKL values
- L - Violates motor Limit

The sector numbers for the legacy 16 sectors are displayed as decimal values. The sector number of the remaining 4080 sectors are displayed in hexadecimal. If *arg* is missing, the default behavior is the same as if *arg*=0x02.

setkappa [*kappa_tilt_angle*] – For kappa-type diffractometers, tells the software the fixed value of the kappa tilt angle.

startgeo – Calls the several macros used to setup the geometry mode.

savegeo – Prints the values of all the geometry configuration parameters in assignment format. Can be used to save the parameters as a command file that can be read back later.

sz [*h0 k0 l0 h1 k1 l1*] – For *zone* mode, sets the two reciprocal space vectors z_0 and z_1 that define the plane of the scattering vector.

cz *a b* – For the two zone vectors z_0 and z_1 calculates and displays Q for the vector $a z_0 + b z_1$.

mz *a b* – For the two zone vectors z_0 and z_1 moves the diffractometer to the Q value corresponding to the vector $a z_0 + b z_1$.

on_daz – Enables detector rotation.

off_daz – Disables detector rotation.

ZEROS AND ROTATION SENSE

In order for the calculations in the *psi*-circle code to work correctly, you must set the rotation sense and zeros of the circles according to the conventions built into the code.

The laboratory coordinate system used in the code is right handed. The x axis points upward and the y axis points along the direction of the incident beam. When all circles are at zero, the z axis points outward from a sample mounted on the ϕ circle.

The rotation sense is defined with respect to the positive unit vectors of the coordinate system described above. For a right-handed rotation, if your thumb points along the unit vector that is in line with the rotation axis, your fingers will point in the positive rotation direction.

The rotation sense of an axis can be changed by changing the sign of the *steps-per-degree* parameter or the *sign-of-user-*-dial* parameter in the *config* file. You should set the sign of the first parameter so that the *spec* dial positions agree with the dial indicators of each circle.

Set the sign of the second parameter as necessary to make the spec user angles agree with the rotation sense described below.

The mu and nu circles are in the y - z plane, with the rotation axes along the x axis with a right-handed rotation convention. The zero of mu is such that the eta circle is on the negative side of the x - y plane with its rotation axis along the z axis. Likewise, the zero of nu is such that the delta circle is also on the negative side of the x - y plane with the rotation axis also along the z axis.

The delta, eta and phi circles all have a left-handed rotation sense. The delta circle is at zero when the detector arm is along the positive y direction. Eta is zero when the chi circle is in the x - z plane, with the chi axis along the y axis. The chi rotation is right-handed. The zero of chi puts the phi rotation axis along the z direction with the "surface normal" of the phi table pointed in the positive z direction. The zero of the phi circle is arbitrary.

In the kappa configuration, the eta, chi and phi real motors are replaced with keta, kap and kphi. The keta and kphi circles have the same left-handed rotation convention as eta and phi, while kap is right handed. The zero of keta is chosen so that the rotation axis of the kap motor lies in the positive y - z quadrant. The zero of kap places the phi rotation axis in the positive z direction. The zero of kphi is arbitrary.

AZIMUTHAL DETECTOR ROTATION

The optional motor daz (for detector azimuth) corresponds to an azimuthal rotation of the detector or detector slits that can be used to maintain a specific angular acceptance at the detector. The theory for this motion is described by E. Vlieg in *J. Appl. Cryst.*, **31**, 198-203 (1998). (See <http://journals.iucr.org/j/issues/1998/02/00/pe0028>)

The macros on_daz and off_daz are available to enable and disable the rotation. The automatic rotation only occurs when moving to reciprocal space positions. That is, the target position is calculated along with the other geometry motor positions by calcA.