

Component file

```
.....
* Nistao, neutron ray-tracing package
* Copyright 1997-2002, All rights reserved.
* Risoe National Laboratory, Roskilde, Denmark
* Institut Louis Langevin, Grenoble, France
*
* Component: Source_flat
*
* $I
* Written by: Kim Leifmann
* Date: October 30, 1997
* Modified by: IL, October 4, 2001
* Modified by: Emmanuel Fuchs, October 30, 2001. Serious bug corrected.
* Version: $Revision: 1.22 $
* Origin: Risoe
* Release: McStas 1.6
*
* A circular neutron source with flat energy spectrum and arbitrary flux.
*
* $D
* The routine is a circular neutron source, which aims at a square target
* centered at the beam (in order to improve MC-acceptance rate). The angular
* divergence is then given by the dimensions of the target.
* The neutron energy is uniformly distributed between E0-dE and E0+dE.
*
* Example: Source_flat(radius=0.1, dist=2, xv=1, yh=1, E0=14, dE=2)
*
* $P
* radius: (m) Radius of circle in (x,y,0) plane where neutrons
* are generated.
* dist: (m) Distance to target along z axis
* xv: (m) Width(x) of target
* yh: (m) Height(y) of target
* E0: (meV) Mean energy of neutrons.
* dE: (meV) Energy spread of neutrons.
* Lambda0 (AA) Mean wavelength of neutrons
* dLambda (AA) Wavelength spread of neutrons
* flux: (1/(s*cm**2*st)) Energy integrated flux
*
* $E
*.....
DEFINE COMPONENT Source_flat
DEFINITION PARAMETERS { }
SETTING PARAMETERS (radius, dist, xv, yh, E0=0, dE=0, Lambda0=0, dLambda=0, flux=1)
OUTPUT PARAMETERS { }
STATE PARAMETERS (x, y, z, vx, vy, vz, t, s1, s2, p)
DECLASS
%I
double pmat, pdir;
%I
INITIALISE
%I
pmat=flux*PI*1e4*radius*radius/wget_ncount();
%I
TRACE
%I
double chi,E,Lambda,v,r, xf, yf, rf, dx, dy;
t=0;
s=0;
chi=2*PI*rand01(); /* Choose point on source */
r=sqrt(rand01()*radius); /* with uniform distribution */
x=r*cos(chi);
y=r*sin(chi);
[
randvec_target_rect(xf, yf, xf, yfdir,
0, 0, dist, xv, yh, ROT_A_CURRENG_COMP);
dx = xf-x;
dy = yf-y;
rf = sqrt(dx*dx+dy*dy+dist*dist);
p = pdir*pmat;
if (Lambda0==0) { /* Choose from uniform distribution */
E=E0+dE*randpal();
v=sqrt(E)*SE2V;
} else {
Lambda=Lambda0+dLambda*randpal();
v = SE2V*(2*PI/Lambda);
}
vx=v*dist/rf;
vy=v*dy/rf;
vz=v*dx/rf;
%I
MEDISPLAY
%I
magrify('');
circle('xy',0,0,0,radius);
%I
END
```

Written by developers
and possibly you!