

SACALC3 v1.4

Please note.

The program and any source code are without warranty of any kind whatsoever. The entire risk regarding the quality and performance of the software is yours. The author releases this program as freeware for use on condition that he will not be liable for any damages, injuries or losses as a consequence of using the program or source code, however caused.

The author has checked the executable program for viruses and other types of malicious code, using up-to-date antivirus software at the time of compilation. However, the author recommends you carry out your own checks as there is no guarantee the program is free from such malicious code.

What the program does

SACALC3 is a program for a personal computer that calculates the solid angle subtended by a right cylinder detector, a cuboid detector, or a spherical detector to a circular or rectangular, plane, thick or point source at any position and orientation to the detector. The program also calculates the number of hits on the detector sides and the average path length through the detector volume (assuming no scattering or absorption). The current performance of personal computers makes it realistic to model the order of 10^9 simulations of radiation emission and achieve accuracies of solid angle estimates typically better than 0.03 %.

The average solid angle is for calculating the fraction of activity emitted from a source incident on the detector volume. The advantage of the Monte Carlo method is that it allows the calculation of average solid angle for source-detector geometries that are difficult to analyse by analytical methods.

Detail is given in 'A Monte Carlo method to calculate the average solid angle subtended by a right cylinder to a source that is circular or rectangular, plane or thick, at any position and orientation. Radiation Protection Dosimetry 2006 vol 118: 459-474. (Note in this paper there is an error in tables 8 & 9. The cuboid source initial positions from the centre of the detector should read 5, 5.5, 6 and 7, not 3, 3.5, 4 and 5.)

The input data can be entered directly on screen. For batch processing, the input data can be read from a file and the results saved to an output file; see the Data options on the program menu. The data input file is a text file with the data in a specific order; an example is provided with the program.

How the program works

The program selects a random point in the source volume, then generates an emission at that point by a random three-dimensional vector and tests to see if it would 'hit' the detector volume. This is carried out a large number of times, and the proportion of hits to emissions is used to calculate the average solid angle. The random number generator used is the Mersenne Twister. The generator period is huge and will not be exhausted at current processor speeds, so avoiding a systematic error from the repetition of the random number stream.

The average path length through the cylinder is calculated from the sum of the straight-line distances between the intersections, and dividing the sum by the number of emissions that intersect the cylinder.

For non-parallel geometries, the Euler angles are used to define the orientation of the source to the detector. The rotations are performed around a floating co-ordinate frame, in particular the sequence commonly named XY'Z'', In the XY'Z'' sequence, one co-ordinate frame rotates by Euler angles into co-incidence with another, the frames having a common origin. In this program, the angles that need to be entered are those needed to rotate the source framework into coincidence with the detector (not the other way around).

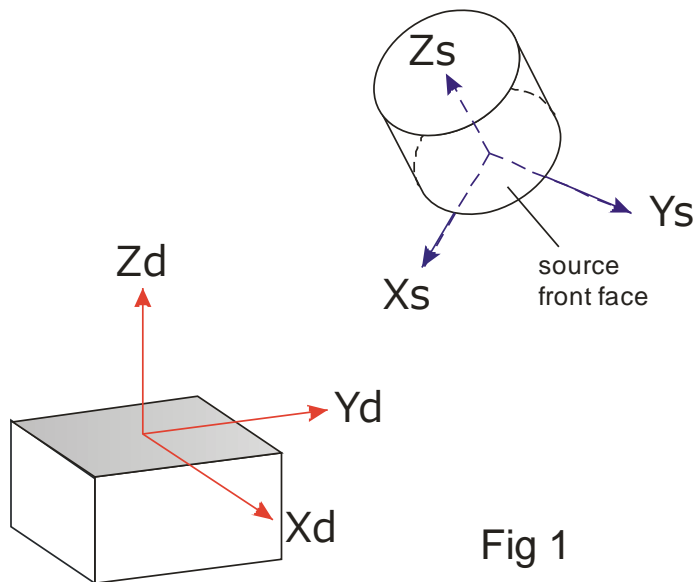


Fig 1

See figure 1 for an example. The detector co-ordinate framework is X_d, Y_d, Z_d and must be chosen so the Z_d axis is normal to, and out of, the detector top. If the detector is cuboid, X_d is chosen to be parallel with one edge of the top

The source co-ordinate framework, X_s, Y_s, Z_s , must be chosen so that the Z_s is normal to the front face and into the source. If the source is a cuboid, X_s is chosen to be parallel with one edge of the front face.

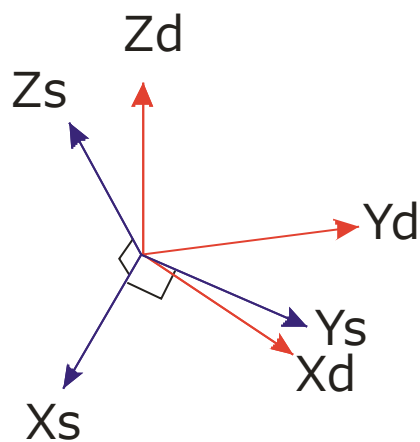


Fig 2

Figure 2 shows the source co-ordinate framework X_s, Y_s, Z_s , and the detector co-ordinate framework X_d, Y_d, Z_d , translated into coincidence. This is described by signed displacements from the origin of X_d, Y_d and Z_d to the origin of X_s, Y_s, Z_s .

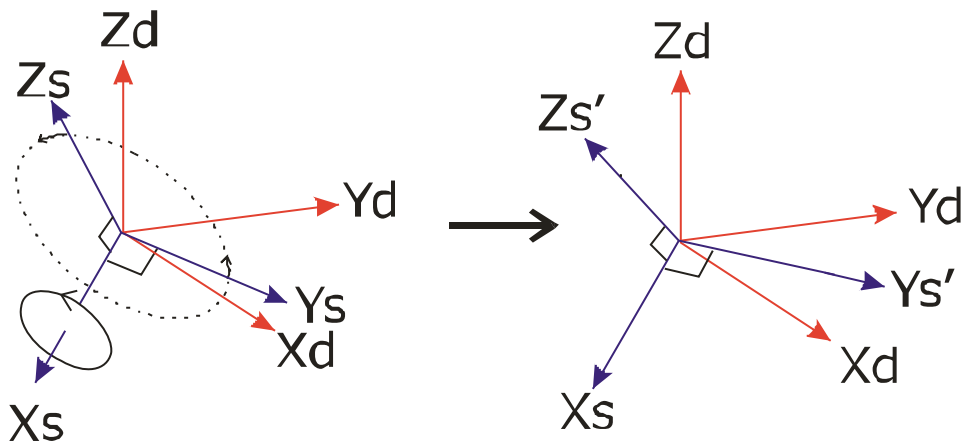


Fig 3

The first rotation is around the source X_s axis. See figure 3. The source rotates so that the Y_s axis moves into the X_d, Y_d plane. The new source Y axis is now labelled $Y_{s'}$. The direction of the rotation is determined looking into the positive X_s axis; a counter-clockwise rotation is positive, a clockwise rotation is negative. In the example figure 3, the rotation is positive. Note that Z_s has also moved to a new position, labelled $Z_{s'}$.

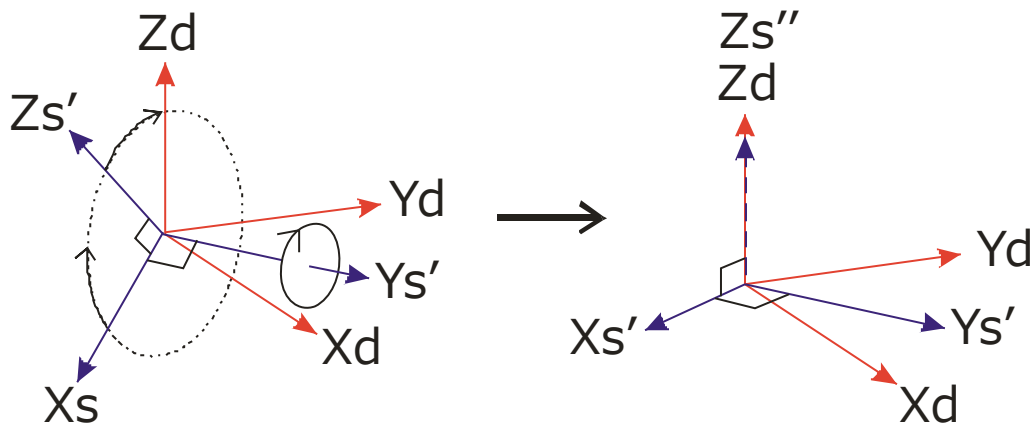


Fig 4

The second rotation is around the source $Y_{s'}$ axis. See figure 4. The source rotates so that the $Z_{s'}$ moves into alignment with Z_d . The direction of the rotation is determined looking into the positive $Y_{s'}$ axis towards the origin; a counter-clockwise rotation is positive, a clockwise rotation is negative. In the example figure 4, the rotation is negative. Note that X_s has also moved to a new position, labelled $X_{s'}$.

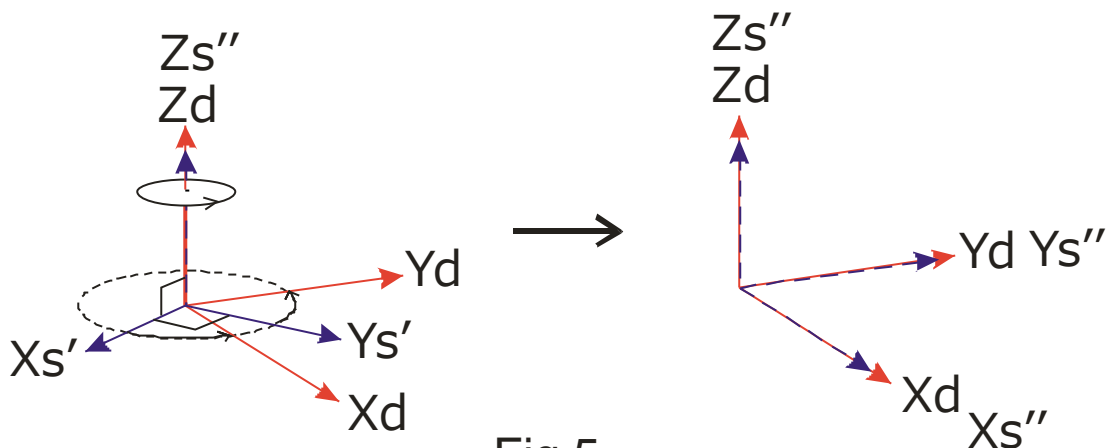
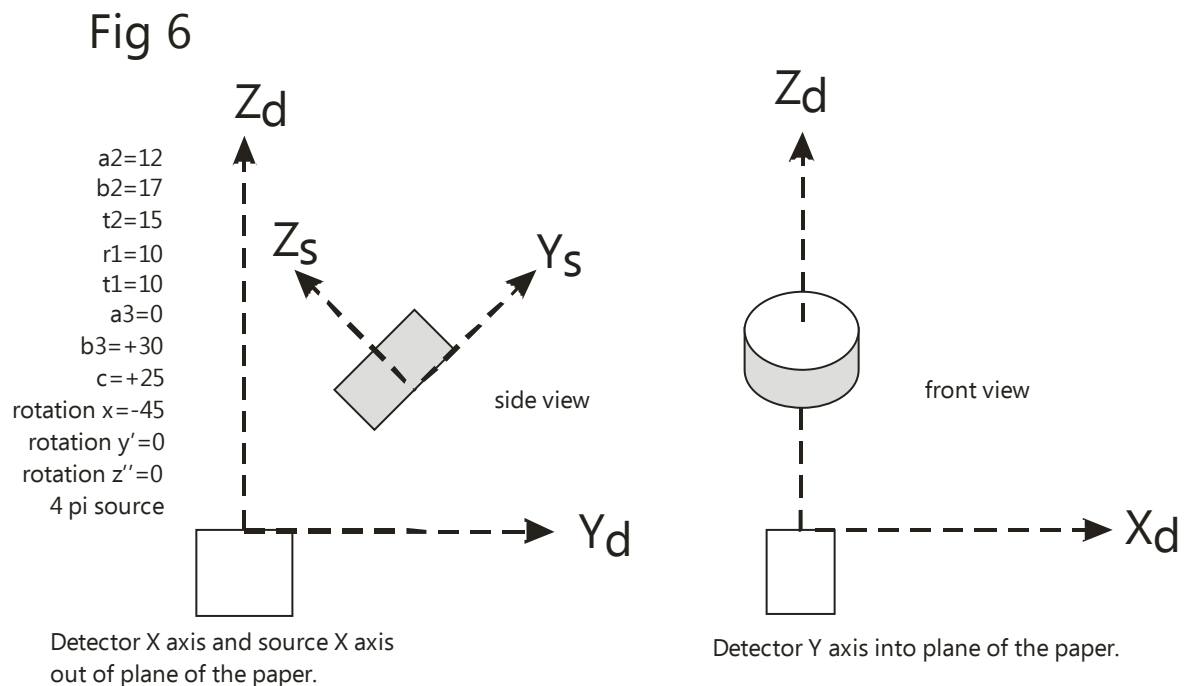


Fig 5

The final rotation is about Zs'' so that Xs' and Ys' align with Xd and Yd respectively. See figure 5. The direction of the rotation is determined looking into the positive Zs'' axis towards the origin; a counter-clockwise rotation is positive, a clockwise rotation is negative. In the example figure 5, the rotation is positive.

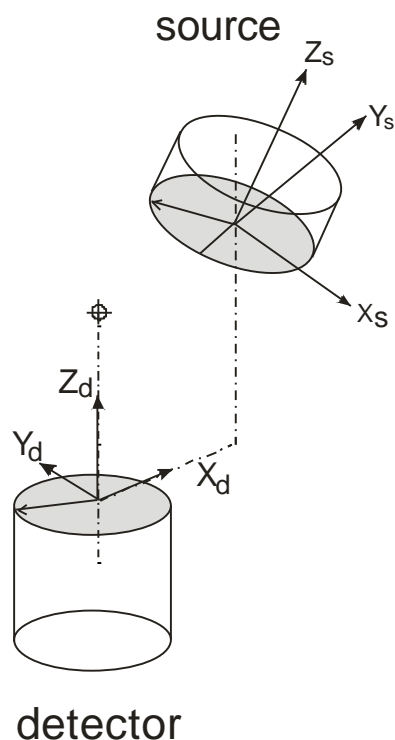
NOTE. The source co-ordinate framework axes are through the centre of the front face of the source, not the centre of the source. Similarly, the detector co-ordinate framework axes are through the centre of the front face of the detector face, not the centre of the detector.

Figure 6 (drawn to scale) shows a specific numerical example. A cylindrical source, radius 10, thickness 10 and cuboid detector 12 wide, 17 deep and 15 high. The centre of the source face is distance +30 along the Yd axis, +25 along the Zd axis and 0 along the Xd axis. The angle between the source Zs angle and detector Zd axis is 45 degrees in the way shown.



The input data is shown on the left of the diagram.

If the source and detector are both cylindrical, the X_d axis is chosen so that only translations of the source along the X_d and Z_d axes are required to bring the origins into coincidence. See figure 7.



Errors

The results from the modelling are treated as binomially distributed. The higher the number of simulated emissions, the lower the calculation error. If the average probability of a hit on the detector face is p , and the number of emissions is n , the estimated errors are:

p	n	Likely error
0.01	1.0E+07	<1%
0.1	1.0E+07	<0.3%
0.5	1.0E+07	<0.1%
0.01	1.0E+08	<0.3%
0.1	1.0E+08	<0.1%
0.5	1.0E+08	<0.03%
0.01	1.0E+09	<0.1%
0.1	1.0E+09	<0.03%
0.5	1.0E+09	<0.01%
0.01	1.0E+10	<0.03%
0.1	1.0E+10	<0.01%
0.5	1.0E+10	<0.003%

The programme allows the user to select the number of simulated emissions; choose the number to achieve the desired accuracy within the time taken by the program.

NOTE: The program uses double-precision floating point arithmetic. To check if a source is on the cylinder surface, and to avoid problems from cumulative round off errors, two points are assumed co-incident if the differences between their x , y and z co-ordinates are less than $1E-12$. Sources that are located within $1E-12$ units from the cylinder surface are likely to be treated as if they are on the cylinder surface.

Examples of input and output data

Examples: A ZP1481 GM tube (useful diameter 17mm, length 40 mm) is placed coaxially 15 mm above a 20 mm diameter 5 mm thick circular radioactive x-ray source. The source is tilted 25 degrees on its Y axis. The source is a four-pi source, emissions are from all directions from the source, not just the front face. If the radioactive material in the source is uniformly distributed, and a calculation of average solid angle with an error of <1% is required, 10^7 emissions need to be simulated. The data entries will be: source shape checked as cylindrical, detector checked as cylindrical, 8.5 entered in detector radius, 40 entered into detector thickness, 10 entered in source radius, 5 entered into source thickness, 15 entered in displacement c, 0 in displacement d, +25 is entered in the rotation about Y' axis. The four-pi option is checked, and 100000000 (or 1E7) entered into number of emissions. The Start simulation button is then clicked.

Average Solid Angle Calculation

File Data About

Source type

☒ Cylinder or disc

☐ Cuboid, plane rectangle, or line

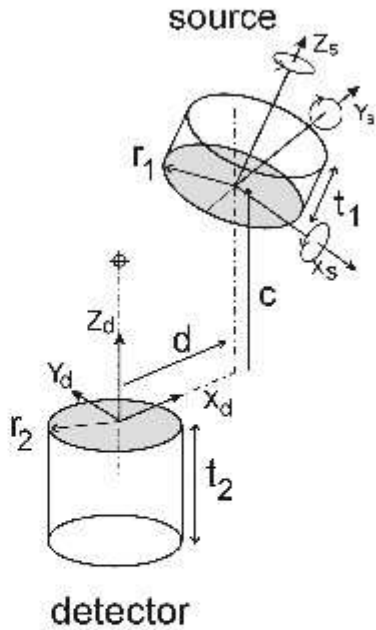
☐ Point

Detector type

☒ Cylinder, or disc window

☐ Cuboid, or rectangular window

☐ Sphere



Source dimensions | Detector dimensions | Detector-source displacement | Rotation angles | Source emissions

source radius, r1

source length, a1

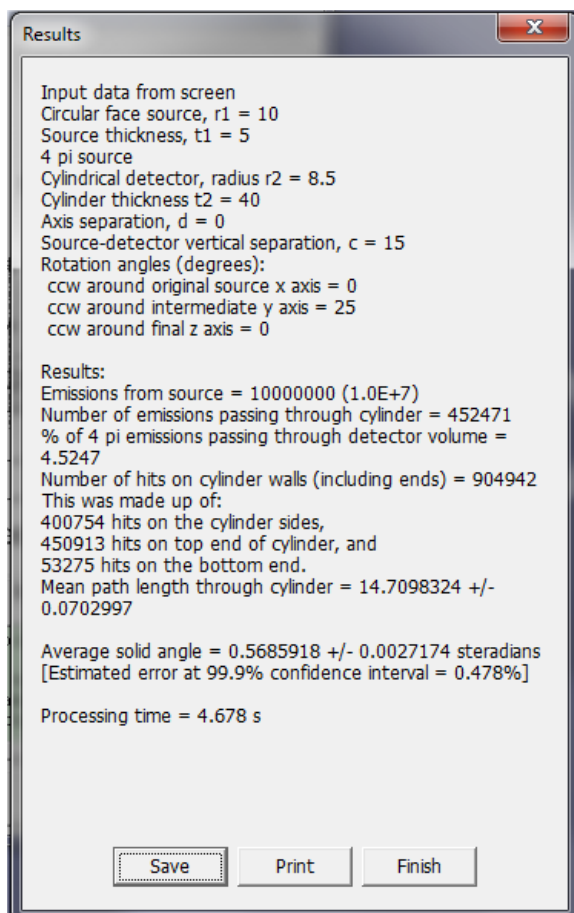
source length, b1

source thickness, t1

For a plane source, set the thickness t1 to 0. For a line source, select cuboid source and set t1 to 0 and a1 or b1 to 0.

Start simulation

The output is:



Programming language

The programme was compiled with Turbo Delphi 2006. On an Intel Core™ i7 Q740 1.73 GHz with 4 Gbytes RAM under OS Windows 7, the program took under 8 minutes to simulate 10^9 random events for a rotated thick source, giving an error of less than 0.032% where the average probability of a hit >0.1 .

Installation

SACALC3.EXE has been tested on Windows XP/Vista/7. It can run from the hard disk, from CD, or from a memory stick. Copy the program file SACALC3.EXE into file into a directory of your choice, e.g. C:\SACALC. To execute the program, click on START, then RUN, and type in the full pathname and program name, e.g. C:\SACALC\SACALC3.EXE, and click OK. Alternatively, create a shortcut on your desktop. To remove, delete the SACALC3.EXE. The program makes no intentional changes to the Windows registry, nor does it install DLLs.

Acknowledgements

The random number generator used is the Mersenne Twister developed by Matsumoto and Nishimura, Hiroshima University Department of Mathematics, and implemented by Pascal Unit written by A Diazgov.

I also wish to thank Mathieu Trocmé (mathieu.trocme@cea.fr) for reviewing the series of beta developments of SACALC3 and the many helpful suggestions for improvements which have been implemented into the final version.