The SIMIND Monte-Carlo Program

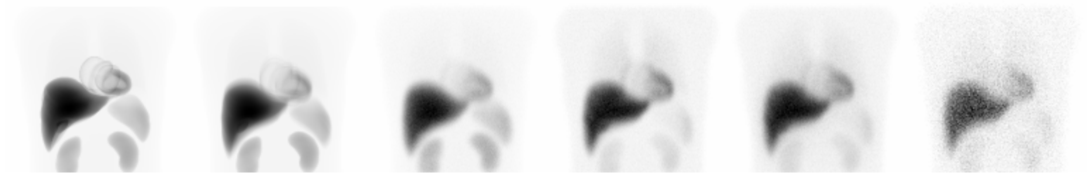
Version 7.0

Manual version created 13/01/2023

Michael Ljungberg, PhD
Professor
Medical Radiation Physics, Lund
Lund University
SE-221 85 Lund, Sweden
michael.ljungberg@med.lu.se

The Monte Carlo simulation code, **`simind`**, describes a standard clinical SPECT camera, and it can easily be modified for nearly any type of calculation or measurement encountered in SPECT imaging. **`simind`** has been developed by Professor Michael Ljungberg (Medical Radiation Physics, Department of Clinical Sciences, Lund, Lund University, Sweden).

The entire code is written in FORTRAN, including versions that are fully operational on Linux systems (x86), Mac systems running MacOS, and Windows systems (x86). The majority of the main code structure is similar for all operating systems; however, in the case of a unique operating system, additional information on the code pertaining to the specific system is provided.



The advantage of a Monte-Carlo program is that it facilitates a detailed study of the radiation transport which allows for inverstigating the reasons for the degradation of images captured by nuclear medicine scintillation camera systems. For example, the images above show a series of simulations. The first image corresponds to an imaging situation without patient motion and with perfect camera resolution. The second image includes patient movements (respiration and heartbeat). The third image is an image with normal SPECT system resolution for patient movements. The next two images show degradation due to photon attenuation and photon attenuation plus scatter contribution, respectively. In the last image, a realistic noise level has been added.

The main references for the program are (1-4).

INTRODUCTION

The **simind** system has two main programs, namely **change** and **simind.** The **change** program provides a way of defining the system to be simulated and writing data to external data files.

The actual Monte Carlo simulation is performed by the **simind** program, which reads input files created by **change** and outputs results to the screen or to different data files. Thus, several input files can be prepared and loaded into a command file for submission to a batch queue, which is a convenient way of working because Monte Carlo simulations by default are time-consuming processes.

This manual provides a detailed explanation of the structure and purpose of **simind** and all related subroutines that are important for the user to understand. In addition, it provides technical instructions for programming the code for specific applications.

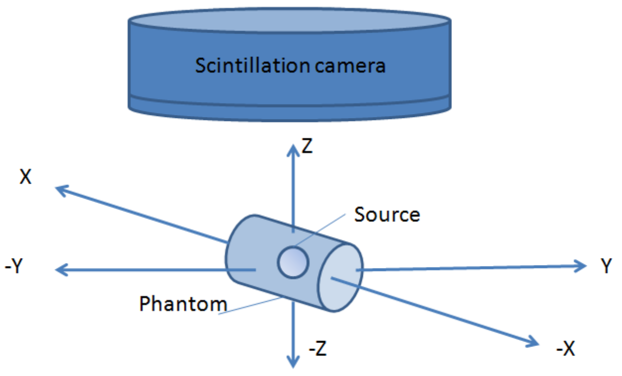
The primitive outlines of the topics addressed herein include an in-depth description of **simind**'s subroutine files, the manipulation of gamma camera parameters, the implementation of phantoms, the general operation of **simind**, and instructions for acquiring planar and SPECT images.

The **change** program enables the user to easily define the desired imaging system. The **change** includes a series of menus that prompt the user to input parameters specific to the description of the system. These parameters are then written to a data file used in **simind**.

A **simind** file name assumes a base name and a three-character extension separated by a period. The output file may have the same base file name as the input file but will usually differ in terms of the file extension. The output file can easily be redirected to other file names.

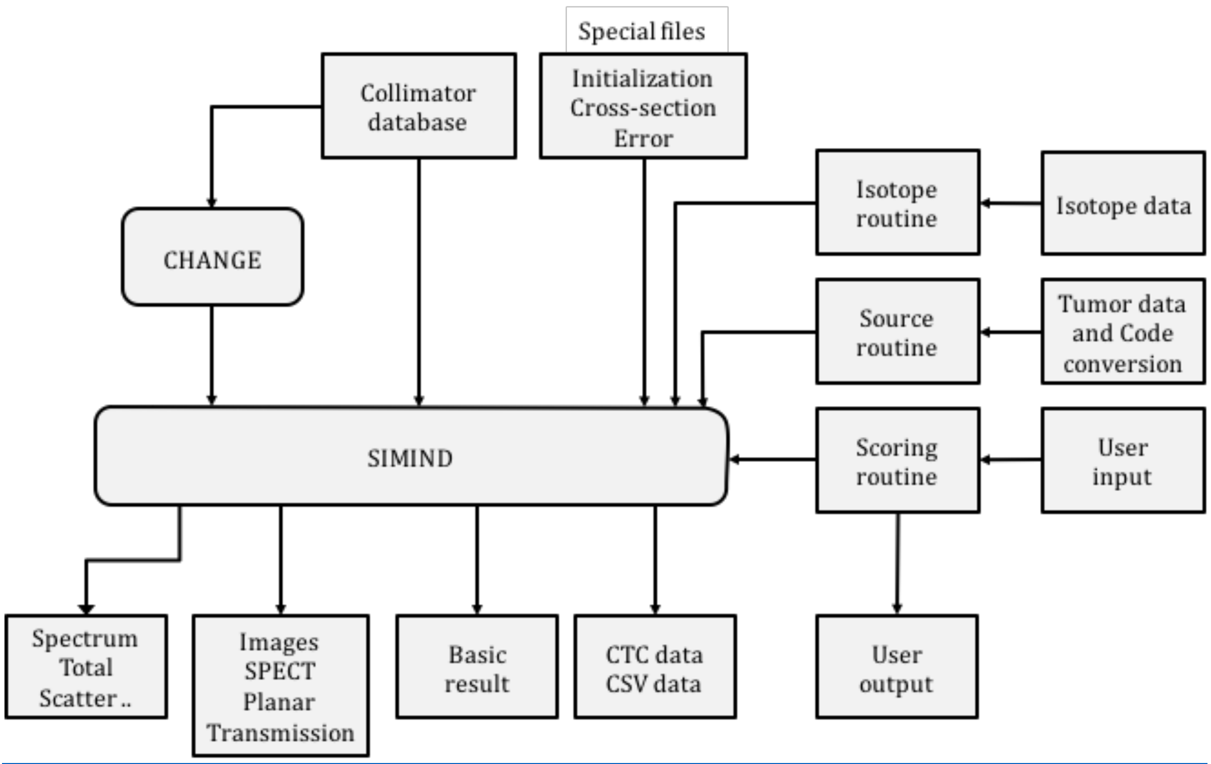
A default file, namely **simind.smc**, is always created when invoking the **change** program. This file may be used when performing tests that do not need to be stored in separate files. Each time that changes are made in **change,** the modifications are written to **simind.smc**. Even when exporting data to smc files with unique names, the content in **simind.smc** is updated. Thus, this file should be regarded as a temporary file.

A file name can be entered after the program name **change** at the operating system command prompt. If this file already exists, it will be used when reading data into **change.** All modifications will be saved to this file and to **simind.smc** at exit. If only the command **change** is entered, data can be exported/imported by Options 5 and 6 in the main menu. The coordinate system is defined in the figure below.

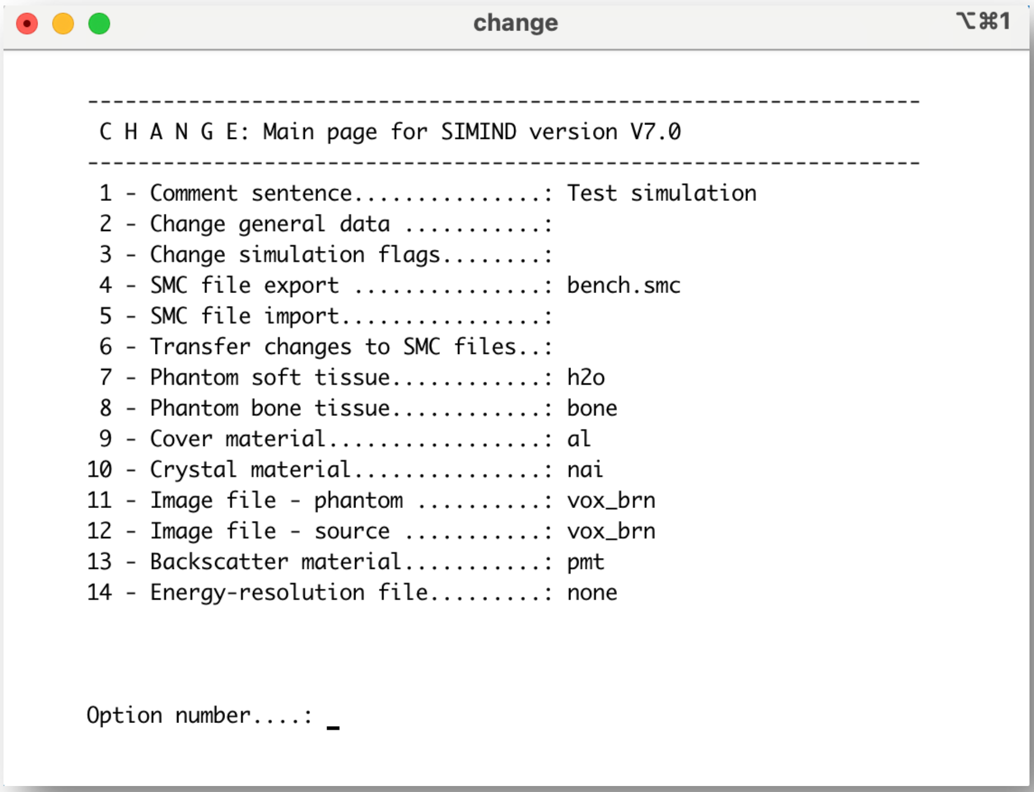


When simulating voxel-based phantoms, the first density/activity image is located towards +X and the last one is located towards –X. When simulating SPECT, the camera rotates in the ZY plane either clockwise or counter-clockwise (index 30).

The flow-chart below describing the link between the change program and simind and the files that are either used or produced in a simulation project.



Main page in Change



```
 ----------------------------------------------------------------
  C H A N G E: Main page for SIMIND version V7.0
 ----------------------------------------------------------------
  1 - Comment sentence...............: Test simulation
  2 - Change general data ...........:
  3 - Change simulation flags........:
  4 - SMC file export ...............: bench.smc
  5 - SMC file import................:
  6 - Transfer changes to SMC files..:
  7 - Phantom soft tissue............: h2o
  8 - Phantom bone tissue............: bone
  9 - Cover material.................: al
 10 - Crystal material...............: nai
 11 - Image file - phantom ..........: vox_brn
 12 - Image file - source ...........: vox_brn
 13 - Backscatter material...........: pmt
 14 - Energy-resolution file.........: none



 Option number....: _
```

Main-1: Not in use at the moment.

Main-2: This option allows the user to read the menus and **`change`** individual values. An index value is prompted, and after selecting the appropriate one, the value of that particular index is given. The menu is immediately updated. A carriage return, given as a value, returns a null value.

Main-3: Menu showing the possible simulation flags that can be used to control the program and that allow the user to invoke or evoke certain types of simulation features. These flags are Boolean and are set/reset by typing the corresponding flag number.

Main-4: Allows the user to export current **`change`** data to a named SMC file. To transfer changes to multiple files, please refer to Option 8 within this section. A carriage return given at the file name prompter returns to the main menu without action.

Main-5: Allows the user to import data from a *.smc file previously stored. A carriage return given at the file name prompter returns to the main menu without action.

Main-6: Allows the user to delete all information (parameter values) from **`change`** (current smc file) and thus define the simulated system from scratch. All detector parameters are set to zero and the simulation flags are set to false.

Main-7: Allows the user to insert a line of comments about the individual simulation that will then appear in the result file (*.res) written by **`simind`**.

Main-8: Allows the user to transfer current changes from an SMC file to other previously created SMC files. By simply invoking the wildcard command, the user can control data transfer from one SMC file to any other SMC file as required. The program prompts for [Y,N,A,Q]. "Y" means that changes will be transferred to the current file, "N" means no transfer, "A" means transfer without further confirmation (ALL), and "Q" means simply quit the option and return to the main menu. The counter that keeps track of the changes is reset

- by entering the program **`change`**,
- by clearing all data by Value 8, or
- after a transfer by Value 10.

Main-9: Tables of energy-dependent cross-sectional values for the photoelectric, Compton, coherent, and pair production interactions unique to the phantom. This is used for both an analytical phantom and a voxel-based photon. The typical phantom cross section is water, given by the file **h2o.cr3**.

Main-10: Tables of energy-dependent cross-sectional values for the photoelectric, Compton, coherent, and pair production interactions unique to the phantom. The typical phantom cross section is water, given by the file **h2o.cr3**. This cross-section table is used only when simulating voxel-based phantoms. The cross section is used for those voxels that have a density value higher than a defined threshold. This threshold is defined in the **`simind.ini`** file located in the **smc_dir** folder. A typical cross-section table is **bone.cr3**, but it could also be the same as the selection in **Main-9.**

Main-11: Tables of energy-dependent cross-sectional values for the photoelectric, Compton, coherent, and pair production interactions unique to the cover material incorporated into **`simind`**. In the cover, interactions occur but no scoring of deposit energy is made. A typical cross-section material for this compartment is **al.cr3.**

Main-12: Tables of energy-dependent cross-sectional values for the photoelectric, Compton, coherent, and pair production interactions unique to the detector crystal material incorporated into **`simind`**. The crystal of the detector is typically sodium iodide, and it is denoted by the file name, **nai.cr3**.

Main-13: Allows the user to input a name for a file that stores density maps for simulations of a non-homogeneous phantom. The file extension default is *.smi.

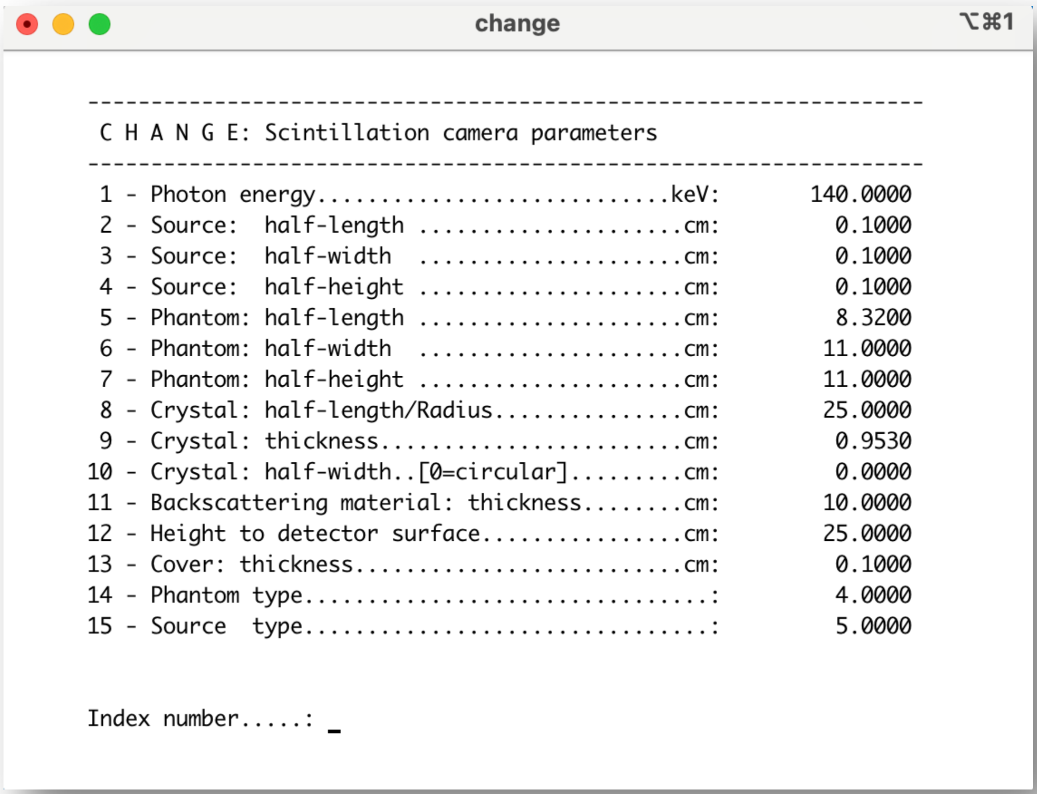
Note: If you are using images created from MCAT or NCAT/XCAT software, you should only give the base name of the file. **`simind`** adds the characters 'atn_av'. It is assumed that the MCAT/NCAT/XCAT files have the extension 'bin'.

Main-14: Allows the user to input a name for a file that stores source maps for simulation of a nonhomogeneous source. Similarly, the file extension default is *.smi. The file name is given to **`simind`** by the data file.

Note: If you are using images created from MCAT or NCAT/XCAT software, you should only give the base name of the file. **`simind`** adds the characters 'act_av'. It is assumed that the MCAT/NCAT/XCAT files have the extension 'bin'.

Main-15: Allows the user to input a file name for the backscattering media. This simulates the backscattering of photons from light guides and photo-multiplier tubes (PMTs). However, the simulation is based on a solid material whose size is equal to that of the crystal but with a user-selected thickness. Index-11 in MENU 2 gives the thickness. The number of scatter orders is set in the **`simind.ini`** file.

Index Page 1



```
------------------------------------------------------------------
 C H A N G E: Scintillation camera parameters
------------------------------------------------------------------
 1 - Photon energy...........................keV:      140.0000
 2 - Source:  half-length ....................cm:        0.1000
 3 - Source:  half-width  ....................cm:        0.1000
 4 - Source:  half-height ....................cm:        0.1000
 5 - Phantom: half-length ....................cm:        8.3200
 6 - Phantom: half-width  ....................cm:       11.0000
 7 - Phantom: half-height ....................cm:       11.0000
 8 - Crystal: half-length/Radius..............cm:       25.0000
 9 - Crystal: thickness.......................cm:        0.9530
10 - Crystal: half-width..[0=circular]........cm:        0.0000
11 - Backscattering material: thickness.......cm:       10.0000
12 - Height to detector surface...............cm:       25.0000
13 - Cover: thickness.........................cm:        0.1000
14 - Phantom type...............................:        4.0000
15 - Source  type...............................:        5.0000


Index number.....: _
```

Index-1: The energy in keV of the photon is given here. **`simind`** simulates only the transport of photons and disregards the transport of secondary electrons. The energy imparted by the electrons is assumed to be locally absorbed at the interaction site. Even if the cross-section tables include photon energies of up to several MeV, the user should be aware of the possibility of inaccuracy of the results when simulating small crystal sizes and high photon energies.

If the value is negative, a call to the **isotope** routine is made with the purpose of defining the initial photon energy from this routine instead of using the value defined by Index-1. However, the absolute value of Index-1 is used to define the location and width of the energy window and other parameters [keV].

Index-2: The x-dimensions of the standard source shapes are defined by this parameter. Please note that the numerical values entered for the corresponding dimensions are half of the total dimensions. Standard source geometries can be point, ellipsoidal, rectangular, or cylindrical (elliptical) in both the horizontal and the vertical directions. The choice for the actual shape of the sources is given in Index-15 described below [cm].

If voxel-based source maps are simulated, then this value defines the half-length of all slices defined by the phantom. For example, assuming 128 slices and a value of 40 for Index-2, the slice thickness will be 2*40/128 = 0.63 cm.

Index-3: The y-dimensions of the standard source shapes are defined. Please note that the numerical values entered for the corresponding dimensions are half of the total dimensions. Standard source geometries can be point, ellipsoidal, rectangular, or cylindrical (elliptical) in both the horizontal and the vertical directions. The choice for the actual shape of the sources is given in Index-15 described below [cm].

For voxel phantoms, this value is not used. Index-31 is used to define the voxel sides of the phantom (not the slice thickness).

Index-4: The z-dimensions of the standard source shapes are defined. Please note that the numerical values entered for the corresponding dimensions are half of the total dimensions. Standard source geometries can be point, ellipsoidal, rectangular, or cylindrical (elliptical) in both the horizontal and the vertical directions. The choice for the actual shape of the sources is given in Index-15 described below [cm].

For voxel phantoms, this value is not used. Index-31 is used to define the voxel sides of the phantom (not the slice thickness).

Index-5: The x-dimension of the standard phantoms is defined. The dimensions of the phantoms are defined similarly to the source distribution described above in that all dimension values are just half of their total values [cm].

If voxel-based phantoms are simulated, the values define the half-length of all slices defined by the phantom. For example, assuming 128 slices and a value of 40 for Index-2, the slice thickness will be 2*40/128 = 0.63 cm.

Index-6: The y-dimension of the standard phantoms is defined. The dimensions of the phantoms are defined similarly to the source distribution described above in that all dimension values are just half of their total values. For voxel phantoms, this value is not used. Index-31 is used to define the voxel sides of the phantom (not the slice thickness) [cm].

Index-7: The z-dimension of the standard phantoms is defined. The dimensions of the phantoms are defined similarly to the source distribution described above in that all dimension values are just half of their total values. For voxel phantoms, this value is not used. Index-31 is used to define the voxel sides of the phantom (not the slice thickness) [cm].

Index-8: The length of the cylindrical crystal in the x-direction. If Index-10 is set to zero, then the detector is assumed to be of a cylindrical shape with a radius equal to the value defined Index-8 [cm].

Index-9: The thickness of the scintillation crystal is given here [cm].

Index-10: The width of the detector. If this index is equal to zero, then the detector is assumed to be of a cylindrical shape with a radius equal to the value defined by Index-8 [cm].

Index-11: This value determines the thickness of the volume behind the scintillation crystal that simulates the light guides and PM tubes [cm]. The number of scatter-orders in the backscattering media is defined in the **`simind.ini`** file.

Index-12: The distance from the origin of the coordinate system (see Figure 1) to the lowest part of the detector. Note that, depending on the defined simulation flags, this can be the collimator lower surface, the protection layer, or the scintillation crystal itself [cm].

If this is set to a negative number when using analytical phantoms, the distance will be equal to Index-5 plus the absolute value of this negative number. This is convenient when simulating various phantoms using the same phantom surface to camera distance.

Index-13: A protective cover that has the same shape as the scintillation crystal can be simulated. However, no imparted energy is scored within the cover. The material can also be different and is defined by the file name of the cross-section table in the main menu. Index-13 then defines the thickness of this protective cover [cm].

Note: In this version of **`simind`**, the protective cover works only for photons that are impinging on the lower side of the detector.

Index-14: Integer flag indicating the type of phantom that is to be simulated. Available standard phantoms are

2 = Rectangular phantom,
3 = Vertical cylindrical phantom, and
4 = Horizontal cylindrical phantom.

Note: The cylindrical phantom can be elliptical if either Index-5 and -6 or Index-6 and -7 (depending on the direction of the phantom) are chosen not to be equal.

If the flag value is negative, the phantom is simulated using binary density maps. The file name storing the particular maps are defined in Option 14 of the main menu and should have the following extensions:

-1 = An density distribution map (16-bit integers,1000*density, extension: *.dmi)
-2 = The Zubal Voxel-Man density distribution (8-bit, extension: *.dat)
-3 = The Zubal brain density distribution (8-bit, extension: *.dat)
-4 = The Zubal whole-body Voxel-Man density distribution (8-bit, extension: *.dat)
-5 = Code-based Zubal phantom (8-bit, extension: *.dat)
-6 = Code-based NCAT/XCAT phantom (32-bit float, extension: *.bin)
-7 = Generic NCAT/XCAT phantom (32-bit float, extension: *.bin)
-8 = Generic MCAT phantom (32-bit float, extension: *.bin)

The non-homogeneous phantom is further defined by Index-31 to -38, which are described below.

Note: Because the coordinate system for MCAT/NCAT axially is rotated by 180 degrees compared to **`simind`**, these images are flipped vertically when reading into **`simind`**.

Note: NCAT and MCAT add the label "atn_av" to the base name. This is also done by **`simind`**, so you should only give the base name for the phantom in Index-14 in the main menu.

Note: When simulating time frames generated by NCAT or MCAT, refer to the information concerning the switches /SF and /DF under the section "Running **`simind`**".

Index-15: Integer flag indicating the type of source distribution that is to be simulated. The available standard source shapes are as follows:

1 = Ellipsoidal source
2 = Rectangular source
3 = Vertical cylindrical source
4 = Horizontal cylindrical source
5 = Point source (x,y,z = 0,0,0)
6 = Simple cardiac phantom (refer to the information about this below)
7 = Multiple spheres phantom (old jaszak program)

If a horizontal cylinder is selected, then Index-2 is the axial source length and Index-3 and -4 define the other axes, respectively. If a vertical cylinder is chosen, then Index-4 defines the axial source length and Index-2 and -3 define the remaining axes.

Note: The cylindrical source distribution can be elliptical if either Index-2 and -3 or Index-3 and -4 (depending on the direction of the source) are chosen not to be identical.

If the flag value is negative, a call is made to a user-written source routine. This source routine should then provide the coordinates of a sampled decay position through the parameter list.

This procedure is described further in the manual. If the default source map routine is used, it supports the following phantoms.

-1 = An source distribution map (16-bit integer, extension: *.smi)
-2 = The Zubal Voxel-Man source distribution (8-bit, extension: *.dat)
-3 = The Zubal brain source distribution (8-bit, extension: *.dat)
-4 = The Zubal whole-body Voxel-Man source distribution (8-bit, extension: *.dat)
-5 = Code-based Zubal phantom (8-bit, extension: *.bin)
-6 = Code-based NCAT phantom (32-bit float, extension: *.bin)
-7 = Generic NCAT phantom (32-bit float, extension: *.bin)
-8 = Generic MCAT phantom (32-bit float, extension: *.bin)

Note: The difference between -7 and -8 is that the NCAT/XCAT images are rotated by 180 degrees relative to the MCAT images.

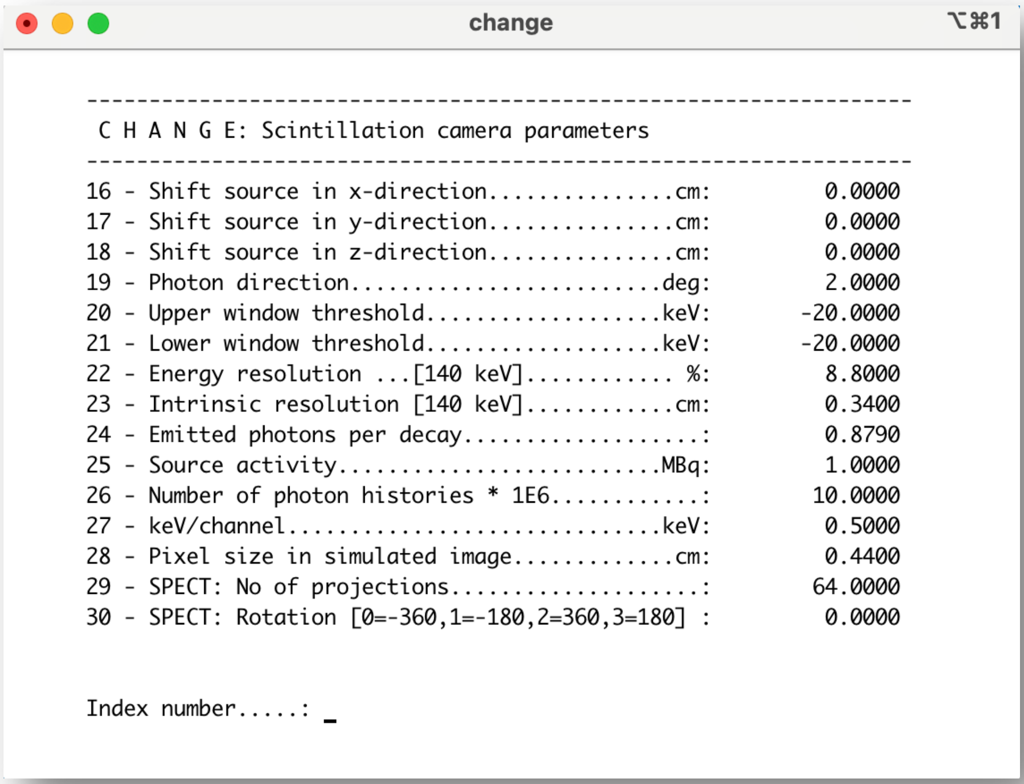
Note: NCAT and MCAT add the label "act_av" to the base name. This is also done by **`simind`**, so you should only give the base name for the phantom in Index-14 in the main menu.

Note: When simulating time-frames generated by NCAT or MCAT, refer to the information concerning the switches /SF and /DF under the section "Running **`simind`**"

Note: When using voxel-based images as a source, the value of Index-26 is not used. Instead, the sum of all pixel values in the source maps is used as photons/projections. This value can be increased by an integer factor n using the switch /NN:n

Note: For Index-2 to Index-8, the voxel size for the source maps follows the voxel size for the density map and the slice thickness, given in Index-31 and calculated by the number of density images and physical axial phantom length, respectively. However, you can set switches /PX: and /TH to override these defaults. Refer to the documentation for the source map provided later in the manual.

Index Page 2



```
 -----------------------------------------------------------------
  C H A N G E: Scintillation camera parameters
 -----------------------------------------------------------------
 16 - Shift source in x-direction..............cm:        0.0000
 17 - Shift source in y-direction..............cm:        0.0000
 18 - Shift source in z-direction..............cm:        0.0000
 19 - Photon direction........................deg:        2.0000
 20 - Upper window threshold..................keV:      -20.0000
 21 - Lower window threshold..................keV:      -20.0000
 22 - Energy resolution ...[140 keV]........... %:        8.8000
 23 - Intrinsic resolution [140 keV]...........cm:        0.3400
 24 - Emitted photons per decay..................:        0.8790
 25 - Source activity.........................MBq:        1.0000
 26 - Number of photon histories * 1E6...........:       10.0000
 27 - keV/channel.............................keV:        0.5000
 28 - Pixel size in simulated image............cm:        0.4400
 29 - SPECT: No of projections...................:       64.0000
 30 - SPECT: Rotation [0=-360,1=-180,2=360,3=180] :        0.0000


 Index number.....: _
```

Index-16: A translation of the x-coordinate of the sampled decay location that has been generated by the standard source geometries or the source routines supplied can be made here. Normally, a decay point is sampled symmetrically around the origin of the coordinate system. Nevertheless, it may be required to move the source, e.g. a point source location, either closer to the surface or to a larger depth.

Index-17: A translation of the y-coordinate of the sampled decay location that has been generated by the standard source geometries or the source routines supplied can be made here. Normally, a decay point is sampled symmetrically around the origin of the coordinate system. Nevertheless, it may be required to move the source, e.g. a point source location, either closer to the surface or to a larger depth.

Index-18: A translation of the z-coordinate of the sampled decay location that has been generated by the standard source geometries or the source routines supplied can be made here. Normally, a decay point is sampled symmetrically around the origin of the coordinate system. Nevertheless, it may be required to move the source, e.g. a point source location, either closer to the surface or to a larger depth [cm].

Note: A positive value of z moves the source towards the camera.

Index-19: This integer flag controls the solid angle in which the photon is to be emitted. The calculation of the solid angle depends on the actual value of Index-19.

1) If the value is 2, then the solid angle is calculated using the following rules in descending priority:

a) If a collimator is used, the solid angle will be calculated from the hole sizes and the effective collimator thickness.

b) If a phantom is simulated, then the solid angle is calculated from the phantom dimension, given by Index-5, -6, and -7, and the detector dimensions such that a photon will always have a chance to strike any part of the detector from any point within the phantom.
c) If no phantom is simulated, the solid angle is calculated from the source dimension, given by Index-2, -3, and -4, and the detector dimensions such that a photon will always have a chance to strike any part of the detector from any point within the source distribution.

2) If the value is 3, the solid angle is calculated using the phantom dimension and camera dimension regardless of whether a collimator has been selected. This is convenient in simulations where penetration is important (Index-53 = 1) and when it is necessary to simulate photons impinging on the whole camera surface.

3) If the value is less than or equal to zero, then the polar angle that defines the solid angle will be calculated from the absolute value given in this index [degrees].

Note: The calculations 1 and 2 above use the values defined in **`change`.** Therefore, the proper value for the solid angle may not be calculated if an external source routine is used. Hence, the user should give an absolute value of the solid angle to be sure that the whole detector is covered. This restriction is only valid when simulating an ordinary detector without collimators.

Note: If a value of zero is given to simulate a narrow-beam situation, then some basic parameters in the result file will not be calculated. The value cannot be equal to or greater than -90. In such cases, the maximum angle is set to 89.99 degrees.

Index-20: This index refers to the upper energy window threshold. Instead of using the absolute values of the upper threshold in keV, a relative window definition can be given by using a negative sign. For example, a value of -20 given to either Index-20 or -21 will be regarded as a 20% (+-10%) energy window centred on the primary photon energy, given by Index-1 [either keV or %].

Index-21: This index refers to the lower energy window threshold. Instead of using the absolute values of the lower threshold in keV, a relative window definition can be given by using a negative sign. For example, a value of -20 given to either Index-20 or -21 will be regarded as a 20% (+-10%) energy window centred on the primary photon energy, given by Index-1 [either keV or %].

Index-22: The energy resolution of the detector is modelled from an on-line convolution of the imparted energy from each photon history using three different ways. If a positive value is given, the energy resolution is modelled by an energy-dependent Gaussian function that varies with $1/\sqrt{E}$. The reference point for the energy resolution is given as the percentage full-width at half-maximum (FWHM) at 140 keV [%]. If negative, then a fixed flat Gaussian energy resolution is modelled. If the value is zero, then the energy resolution is modelled from a fitted function. The simulation Flag-12 can turn off the energy resolution.

Index-23: This index refers to the intrinsic spatial resolution of the crystal. The simulation of this parameter is based on the same concept as that of the energy resolution described above. The spatial resolution is provided as the FWHM at 140 keV and will blur the x- and y-coordinates by a Gaussian function [cm].

Index-24: The number of emitted photons per decay is used to calculate basic detector parameters, such as the sensitivity (cps/MBq) of the detector system. This value can also be defined from the spectrum file. The priority is as follows:

1) If there is a line in the spectrum file starting with the word GAMMA = value, then the emitted photons per MBq will be set to 'value'.
2) If a spectrum file without an explicit GAMMA is defined in the file, then the GAMMA is calculated from the sum of abundances for all photon energies.

3) If no spectrum file is used, the GAMMA obtained from Index-24 in **`change`** is used.

Option 1 above will be useful if only a part of an energy spectrum is to be simulated while retaining the proper value of GAMMA.

Index-25: The activity of the source distribution is used to obtain correct values of the sensitivity and different calculated count rates, as well as correct values for the pixels in the projection data. It does not decide the number of photons that are simulated [MBq].

Note: The corresponding acquisition time per projection is assumed to be 1 s. If the numerical values in the projection data should reflect a different acquisition time, then use Index-25 such that the value reflects the product of activity times the acquisition time.

Index-26: This index represents the number of photon histories that are simulated per projection angle. If a SPECT study is to be simulated, the total number of compiled photon histories will be this value times the number of projection angles. The value is given in thousands of histories. Note that doubling this value does not increase the count level in the final projection nor the energy pulse-height distribution because all data are normalized to the defined activity and acquisition time of 1 s. However, it improves the statistical uncertainty.

Note: This value does not affect the number of histories when simulating voxel-based phantoms. In these cases, the sum of all voxel values will determine the number of histories per projection. This value can be increased by the multiplier, defined by the switch /NN.

Index-27: **`simind`** has a limited number of energy channels for calculating the energy pulse-height distribution. The default value is 512 channels, but it can be set to a different value by Index-80. Index-27 then represents the number of keV/channel. If the defined upper energy window (given by Index-20) is greater than the maximum energy defined by the number of bins times the keV/channel in the spectra, then **`simind`** will increment the keV/channels to a proper value that covers the upper energy threshold. Nevertheless, this check will not decrement the keV/channel value [keV/channel].

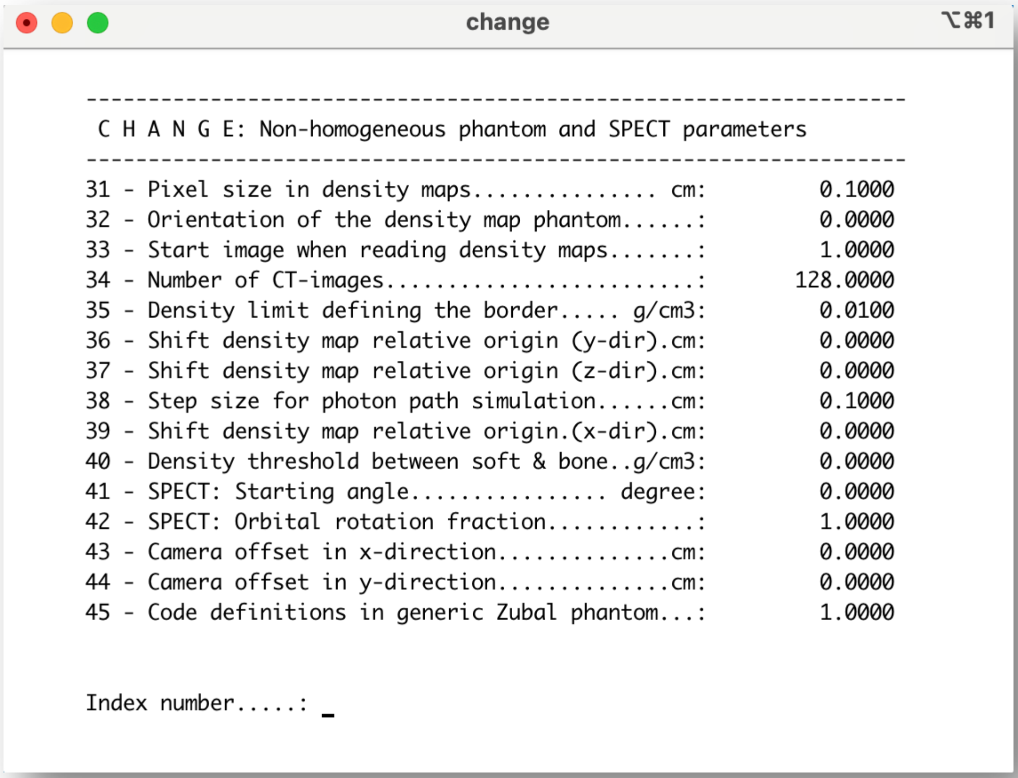
Index-28: The pixel size is used to define the area on the crystal that is represented by the projection image. The matrix is always centred on the cylindrical axis of the crystal and no translation can be made. The actual dimension of the projection matrix is given by Index-76 and -77. However, the created image can be zoomed in or out arbitrarily by giving a smaller/larger value of the pixel size. An error message will appear if an image with zero pixel size is simulated [cm].

Index-29: This value defines the number of SPECT projections to be simulated. Each projection is stored in a SPECT file or in the current directory.

Index-30: This option defines the SPECT rotation. The value is the rotation angle and the sign defines the rotation direction: negative implies counter-clockwise and positive implies clockwise. To be comparable with older versions of **`simind`**, the following translations remain [degrees].

0 = 360 degrees counter-clockwise
1 = 180 degrees counter-clockwise
2 = 360 degrees clockwise
3 = 180 degrees clockwise

Index Page 3



```
-------------------------------------------------------------------
 C H A N G E: Non-homogeneous phantom and SPECT parameters
-------------------------------------------------------------------
31 - Pixel size in density maps.............. cm:       0.1000
32 - Orientation of the density map phantom......:       0.0000
33 - Start image when reading density maps.......:       1.0000
34 - Number of CT-images.........................:     128.0000
35 - Density limit defining the border..... g/cm3:       0.0100
36 - Shift density map relative origin (y-dir).cm:       0.0000
37 - Shift density map relative origin (z-dir).cm:       0.0000
38 - Step size for photon path simulation......cm:       0.1000
39 - Shift density map relative origin.(x-dir).cm:       0.0000
40 - Density threshold between soft & bone..g/cm3:       0.0000
41 - SPECT: Starting angle............... degree:       0.0000
42 - SPECT: Orbital rotation fraction...........:       1.0000
43 - Camera offset in x-direction.............cm:       0.0000
44 - Camera offset in y-direction.............cm:       0.0000
45 - Code definitions in generic Zubal phantom...:       1.0000


Index number.....: _
```

The non-homogeneous phantom simulation is based on using a set of integer matrices that describes the density distribution in a section of the phantom. The file format is described in the section "FILE EXTENSION and FILE FORMATS". Each cell in the density maps must store the density times a factor of 1000. The half-length of the phantom is defined by Index-5. The choices available in this menu for setting up the non-homogeneous phantom and the different SPECT parameters include the following:

Index-31: This value represents the pixel size of the density maps being used to simulate the non-standard phantoms. This value has nothing to do with the pixel size used to form the images and projection matrices. Instead, the value is used to scale the distance from a relative cell position in the 'matrix' space to a physical distance in the coordinate system [cm].

Index-32: Integer value that refers to the direction of the phantom that is defined by the density maps.

0 = density map where *(i,j)* corresponds to direction y,z (Transversal slices)
1 = density map where *(i,j)* corresponds to direction x,y (Sagittal slices)
2 = density map where *(i,j)* corresponds to direction x,z (Coronal slices)

Index-33: This index defines the image number in the density map file that will be read as the first image. If the value is less than zero, then the block number (number of blocks of 512 bytes) in the binary file that defines the first density map is defined. Block zero indicates the beginning of the file. This can be used to skip a file header.

Index-34: Value equal to the number of density maps that will be read into **`simind`**. The memory for these maps is allocated at runtime.

Index-35: Value for the density limit that defines the border. If the density image is slightly noisy, a threshold can be used to properly define the border of the phantom. The ray-tracing technique assumes that the border of the phantom has been reached if a density value that is lower than the given threshold is found [gcm^{-3}].

Index-36: This index refers to the column location (I dir) of the pixel cell in the density map that corresponds to the origin of the coordinate system. Normally, the cell with index *0,0* should be at the centre. However, if a translation has been performed in the maps, then the shift of the centre pixel cell from the origin can be accounted for [cm].

Index-37: This index refers to the row location (J dir) of the pixel cell in the density map that corresponds to the origin of the coordinate system. Normally, the cell with index *0,0* should be at the centre. However, if a translation has been performed in the maps, then the shift of the centre pixel cell from the origin can be accounted for [cm].

Index-38: This value is the step size that is used to ray-trace the photon path through the density maps. This value should be less than the pixel size of the density map (Index-31). However, the actual computational time may depend on this value, so do not select a low value that is too small [cm].

Index-39: This index refers to the location (K dir) of the slice in the density map that corresponds to the origin of the coordinate system. This allows the user to shift the centre of the phantom in the x-direction relative to the origin of the system [cm].

Index-40: The density threshold that defines the **`change`** from the cross-section file for soft tissue (Index-9 in the main menu) to the cross-section file for bone (Index-10 in the main menu). If the value is set to zero, then the default value, defined in the system file **`simind.ini`** (located in the **smc_dir** directory), is used.

Index-41: This index refers to the starting angle of the detector in a SPECT simulation. A positive value indicates a clockwise rotation, while a negative value indicates a counter-clockwise rotation. A value of zero means that the detector is centred on the z-axis of the coordinate system [degrees].

Index-42: This value controls the non-circular orbit of the SPECT camera. This orbit can operate in different ways. A positive value determines a parameter called ”orbital rotation fraction” and is used to simulate a non-circular SPECT study based on a major and minor axis. This fraction defines the maximum distance from the origin to the lowest part of the detector in the y-direction relative to the distance given in Index-12. For example, a fraction of 1.5 yields a maximum radius of rotation along the y-axis of 15 cm if Index-12 is set to 10 cm. This value can be less than unity but not less than or equal to zero.

If the value is negative, the distance from the origin to the detector surface is calculated for each angle using the density map. This mimics a SPECT system with photosensitive detection of the outline. The absolute value of Index-42 will define the distance (air gap) between the surface of the phantom and the detector. The distances are stored for each projection angle in a file whose base name is the same as the input name with the extension *.cor. Note that it is important for the density value that defines the border (Index-35) to be higher than 0.0; otherwise, the distances calculated will correspond to the density matrix borders and not to the object inside the density matrix. The *.cor file is a simple ASCII text file, and it can then be used in a reconstruction program that can compensate for the distance-dependent spatial resolution.

Index-43: This index allows a shift of the camera head relative to the phantom. This can be useful if a large phantom is simulated with a source distribution outside the field-of-view of the camera [cm].

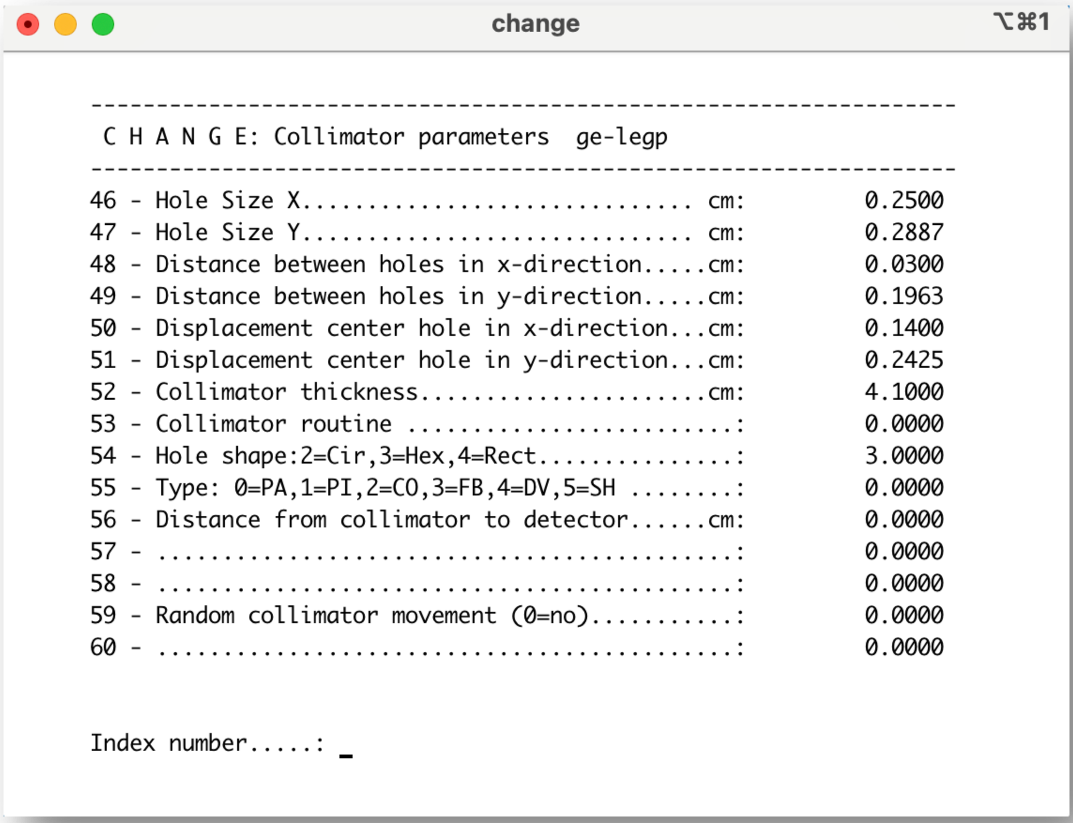
Index-44: This index specifies a shift of the camera in the y-direction relative to the phantom. If it is zero, the camera centre and the phantom centre are aligned to each other [cm].

Index-45: A special code-based phantom format is supported when simulating a voxel-based phantom. Instead of defining the attenuation and activity directly in the images, these values are defined in a table. This facilitates a setup of multiple source distributions for a given phantom because only one binary voxel phantom is needed. These phantoms should be either in the 8-bit mode (Index-14, -15 = -5) or as float (32-bit values) arrays (Index-14, -15 = -6). Each voxel should have a value from 0 to 255, where the number identifies particular structures/organs. The definition of the organ, its code, and corresponding densities and relative activity concentrations are given in a *.zub file (refer to vox_man.zub in the **smc_dir** directory as an example). In a *.zub text file, there are different sections that are valid for different phantoms. You may also develop your own phantom and add a code section to the text file as long as it follows the format.

The supported phantoms for this code-based feature are the Zubal torso phantom **vox_man1.dat** (Index-14, -15 = -1), the Zubal brain phantom **vox_brn.dat**, and the whole-body version of the Zubal torso phantom **vox_man3.dat.** Further, a segmented RSD Torso rsd.dat is available. Because these phantoms have the same code for different regions, you must select the proper code page in the *.zub file.

This is a number indicated in the text file. If you create your own code-based phantom in a byte format, then you can add your own section in the *.zub file if you follow the format of the earlier phantom and select a unique number. The main advantage of this feature is that you can easily **change** the densities and activity per voxel by editing a simple text file instead of creating large image files.

Index Page 4



```
-----------------------------------------------------------------
 C H A N G E: Collimator parameters  ge-legp
-----------------------------------------------------------------
46 - Hole Size X............................ cm:        0.2500
47 - Hole Size Y............................ cm:        0.2887
48 - Distance between holes in x-direction.....cm:       0.0300
49 - Distance between holes in y-direction.....cm:       0.1963
50 - Displacement center hole in x-direction...cm:       0.1400
51 - Displacement center hole in y-direction...cm:       0.2425
52 - Collimator thickness.....................cm:        4.1000
53 - Collimator routine ........................:       0.0000
54 - Hole shape:2=Cir,3=Hex,4=Rect...............:       3.0000
55 - Type: 0=PA,1=PI,2=CO,3=FB,4=DV,5=SH ........:       0.0000
56 - Distance from collimator to detector......cm:       0.0000
57 - .........................................:       0.0000
58 - .........................................:       0.0000
59 - Random collimator movement (0=no)..........:       0.0000
60 - .........................................:       0.0000


Index number.....: _
```

Included in `change` is a large database that covers most of the commercial collimators available for both SPECT and planar imaging. The collimator parameters are stored in a file, called collim.col, in the special SMC directory. The contents of the file are listed in Appendix A.

A code name must be given at the prompt to include a specific collimator. For example, if a General Electric Low-Energy General-Purpose collimator is to be simulated, the following command can be entered:

```
    Index_Number…> *ge-legp
```

The values for that particular collimator will then be updated. When simulating a Standard collimator that is not included in the database, simply input the inner hole diameter (Index 46 - Hole Size X), the septa thickness (Index 48- Distance between holes in x-direction), the collimator thickness (Index 52: Collimator Thicknes), and the type of collimator (Index 54 Hole Shape: 2= Cir, 3= Hex, 4=Rect). If this procedure is followed, then the command given at the prompt is:

```
    Index_Number…> *standard
```

At this point, the new collimator parameters will be calculated.

Note: This calculation assumes a 'normal' collimator, i.e. four holes in each corner and one in the middle if a hexagonal or elliptical/circular hole shape has been chosen.

Index-46: This parameter is the length of the collimator hole in the x-direction [cm]. For hexagonal holes, this value defines the inner square-to-square hole diameter commonly referred to in many collimator data-sheets as an inscribed circle or a flat-to-flat distance [cm].

Index-47: This parameter is the length of the collimator hole in the y-direction [cm].

Note: For a hex-hole collimator, this value is normally not given by the user. Instead, it is calculated using the command *standard (refer to the above text).

Index-48: This parameter is the distance between two collimator holes in the x-direction. In the data-sheet, this distance is often stated as the septal thickness [cm].

Index-49: This parameter is the distance between two collimator holes in the y-direction [cm]. This value is normally not given by the user. Instead, it is calculated using the command *standard (refer to the above text).

Index-50: This parameter is the distance of the shift in the x-direction from the centre to adjacent holes. This parameter holds only for circular or hexagonal collimator holes [cm]. For a hex-hole collimator, this value is normally not given by the user. Instead, it is calculated using the command *standard (refer to the above text).

Index-51: This parameter is the distance of the shift in the y-direction from the centre to adjacent holes. This parameter holds only for circular or hexagonal collimator holes [cm]. This value is normally not given by the user. Instead, it is calculated using the command *standard (refer to the above text).

Index-52: This parameter is the physical thickness of the collimator [cm].

Index-53: The type of collimator routine used.

0 = the earlier default collimator with analytical weighting is used (developed by Eric C. Frey et al.).
1 = the parallel hole collimator with hex or rectangular holes that allow for penetration and scattering is used (used before in the SIMINDC and SCATTWINC programs).

Index-54: This parameter denotes the geometric shape of the collimator holes. The possible options for this flag at present are 3 (hexagonal holes) and 4 (rectangular holes). This option is only used in collimator 1 (Index-53).

Index-55: This parameter defines the type of collimator used when **Index-53 is set to 0**. The possible options are

0 = Parallel-hole collimator (Index-51 = 0,1)
1 = Pinhole collimator (Index-51 = 1)
2 = Converging collimator (Index-51 = 0)
3 = Fan-beam collimator (Index-51 = 0)
4 = Diverging collimator (Index-51 = 0)
5 = Slant-hole collimator (Index-51 = 1)

Index-56: Collimator-dependent variable 1.

a) Parallel-hole collimators: This will define a possible air gap between the collimator and the cover of the detector.
b) Pinhole collimator (Index-53 = 1): This value defines the air gap between the pinhole and the cover of the detector.

Index-57: Collimator-dependent variable 2.

a) When simulating fan-beam, cone-beam, or diverging collimators with collimator Index-53 = 0, this value defines the focal distance.

Index-58: Collimator-dependent variable 3. It will be used in future releases depending on the choice of the collimators.

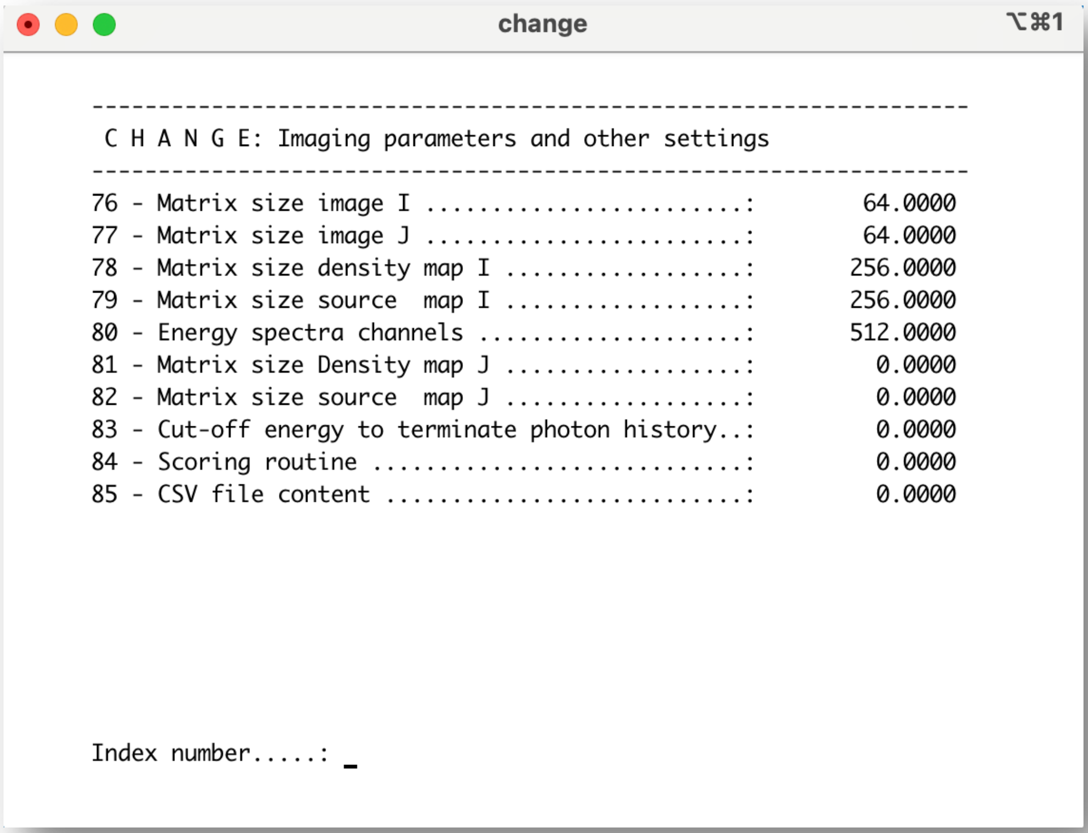
Index-59: If this is 1, then the parallel-hole collimator that handles septa penetration (Index-53 = 1) will be arbitrarily moved (one hole diameter + one septa) to avoid visible hole patterns in the image.

Index-60: Collimator-dependent variable 4. It will be used in future releases depending on the choice of the collimators.

Index Page 5

This was before the radionuclide transmission page, that now has been removed. The page and the numbers 61-75 will be reserved for future developments.

Index Page 6



```
 -------------------------------------------------------------------
  C H A N G E: Imaging parameters and other settings
 -------------------------------------------------------------------
 76 - Matrix size image I ........................:       64.0000
 77 - Matrix size image J ........................:       64.0000
 78 - Matrix size density map I ..................:      256.0000
 79 - Matrix size source  map I ..................:      256.0000
 80 - Energy spectra channels ....................:      512.0000
 81 - Matrix size Density map J ..................:        0.0000
 82 - Matrix size source  map J ..................:        0.0000
 83 - Cut-off energy to terminate photon history..:        0.0000
 84 - Scoring routine ............................:        0.0000
 85 - CSV file content ...........................:        0.0000







 Index number.....: _
```

Index-76: The size of the image matrix in terms of the number of columns. The first column is 1.

Index-77: The size of the image matrix in terms of the number of rows. The first row is 1.

Index-78: Matrix size of the density map in the I direction (column).

Index-79: Matrix size of the source map in the I direction (column).

Note: The Zubal phantom (Index-14 and -15 = -2, -3, -4) has specific hard-coded values equal to 128×128 (Index-2), 256×256 (Index-3), and 192×96 (Index-4). This is because these are known phantoms located in the smc_dir folder.

Index-80: Number of channels in the energy pulse-height distributions. The maximum energy will then be this value times the keV/channel defined by Index-27.

Index-81: Matrix size of the density map in the J direction (row). If this value is zero, then it will be set equal to the value of Index-78.

Index-82: Matrix size of the source map in the J direction (row). If this value is zero, then it will be set equal to the value of Index-79.

Index-83: Cut-off energy used to terminate the photon histories below the lower energy threshold when simulating scatter **in the phantom**. This value can be used in two ways.

1) If the value is positive, then the cut-off value is the actual value in keV.
2) If the value is negative, then the cut-off value is defined as the lower energy window threshold minus the value of Index-83.

When using the scat twin scoring routine (Index-84 = 1) together with a negative number of the cut-off energy, the lowest value of the energy window settings defined in the *.win file will be used as the cut-off value.

Note: The program will be terminated if the cut-off value is higher than the lowest threshold of the energy window.

Note: Because of the energy resolution, an event with a lower energy than the lower energy window value may still enter the energy window because of the Gaussian blurring. Therefore, you need to be careful about the section of the value so that you do not miss these events.

Note: The maximum scatter order is set to 10 automatically when using a cut-off value.

Index-84: Standard scoring routine compiled and linked to the program that can be used. Refer to the separate links for additional information about each of the routines.

0 = A dummy scoring routine
1 = The ”`scattwin`” scoring routine
2 = The ”`list mode`” scoring routine
3 = The ”`forced collimation`” scoring routine
4 = The "`penetrate`" scoring routine

Note: Refer to the "Scoring Routines" chapter for additional information.

Index-85: This option allows for certain results to be written to a text file in a comma-separated value (CSV) format. The advantage is that it is then easy to import results into, e.g. an Excel sheet. There are different ways of writing data, and the output is described below. Please review the page on "*The RES file*" for further information on each of the parameters stated below. In general, the file name will be *output.csv*. However, if a third file is given at the command line, then this file name will be used. For example, by giving the command "***simind*** *input output/85:1 csvname*", a file named csvname.csv will be created.

- **value = 1**: 'ResFile','ComReg','FulReg', 'PilReg','Foto1','Compt1','Effici(W)','SD','Effici(D)', 'Cps/MBq','Cpm/mCi','Peak/C1','Peak/C2','Peak/Tot', 'FWHM','FWTM'

- **value = 2**: 'ResFile','ScattPrim', 'ScattTot','ScatterOrder from 1 to ISCUT','FWHM','FWTM'

- **value = 3**: 'File','HV0','HoleDiam','CollZ','CollDep1','CollDep2','CollDep3','CollDep4', 'CpsMBq', 'Geom (D)','Geom (W)','Pen (D)','Pen (W)','Scatt (D)','Scatt (W)'

- **value = 4**: Scatter/Total (W), ScatterWeight(W), PrimaryWeight(W) for each projection angle. In this option a **new** file is always created!

- **value = 5**: The activities and concentrations for the voxel-based phantom are written.

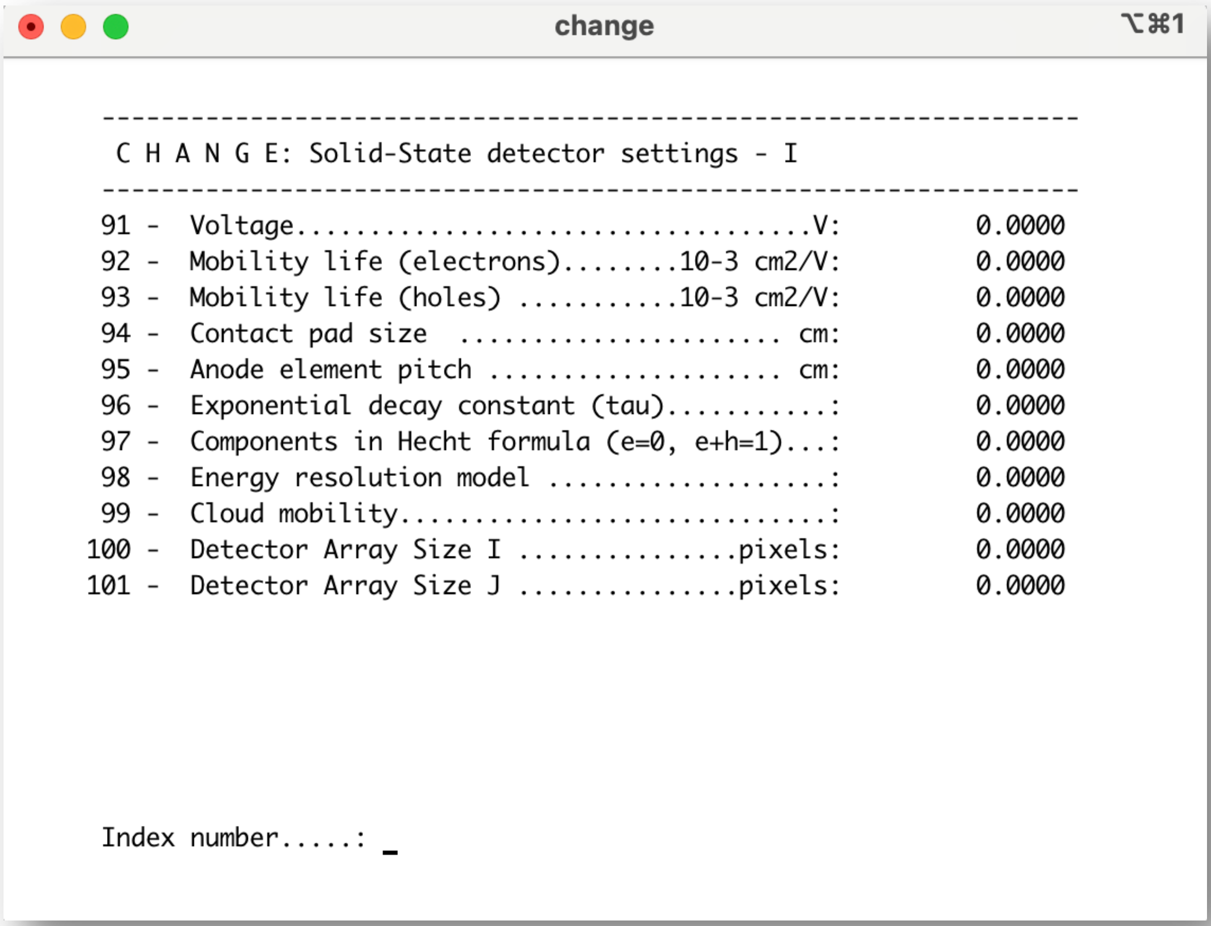
Note: The results of the FWHM and FWTM are only relevant if a point source located at the centre (0,0,0) has been simulated.

Note: If you use this option on the command line with a negative value, i.e. /85:-1, then a blank line is added to the CSV file. This is useful to separate certain data from others.

Note: If no CSV file is present, then a new file is created. Otherwise, a line is added to the present content. If you use this option in a script, be sure to add a command that deletes the CSV file before starting the simulation series.

Note: Send Michael an email if you have a specific combination of results that you would like to be sent to a CSV file that is not covered by the above options.

Index Page 7



```
 ----------------------------------------------------------------
  C H A N G E: Solid-State detector settings - I
 ----------------------------------------------------------------
 91 -  Voltage.................................V:        0.0000
 92 -  Mobility life (electrons)........10-3 cm2/V:       0.0000
 93 -  Mobility life (holes) ..........10-3 cm2/V:       0.0000
 94 -  Contact pad size  ...................... cm:       0.0000
 95 -  Anode element pitch .................... cm:       0.0000
 96 -  Exponential decay constant (tau)..........:       0.0000
 97 -  Components in Hecht formula (e=0, e+h=1)...:       0.0000
 98 -  Energy resolution model ...................:       0.0000
 99 -  Cloud mobility.............................:       0.0000
100 -  Detector Array Size I ..............pixels:       0.0000
101 -  Detector Array Size J ..............pixels:       0.0000




 Index number.....: _
```

It is possible to perform a Monte Carlo simulation of solid-state cadmium zinc telluride (CZT) in this version of **simind**. The detection of an event in these types of detectors is determined by released charges (electron/hole pairs) instead of scintillation light. Each image pixel has its own corresponding detector. There are several properties that differ from those of the common Anger-type scintillation camera detection of radiation. These are described in a paper that also includes several validation studies of the energy spectrum, system sensitivity, and spatial resolution. For additional information, please refer to reference (5).

Index-91: The voltage between the anode and the cathode (V). A typical value is 600 V.

Index-92: This value is the mobility life for the electrons. The parameter is expressed in units of 10^{-3}cm^2/V. A typical value would be 5.

Index-93: This is the mobility life for the holes, and it is expressed in units of 10^{-3}cm^2/V. The mobility life is generally lower than that for electrons, and a typical value would be 0.4.

Index-94: This is size the anode contact pad, expressed in cm. It is assumed that this pad I square.

Index-95: This is the pitch of each detector element and include index 94 and the gap between the elements.

Index-96: Tau is an exponential decay term that facilitates the simplification of Ramo's formulation of induced charge (6-8).

Index-97: The Hecht formula simulating signal loss due to recombination. If the value is 1, then the Hecht formula for both electrons and holes is used. If the value is zero, then

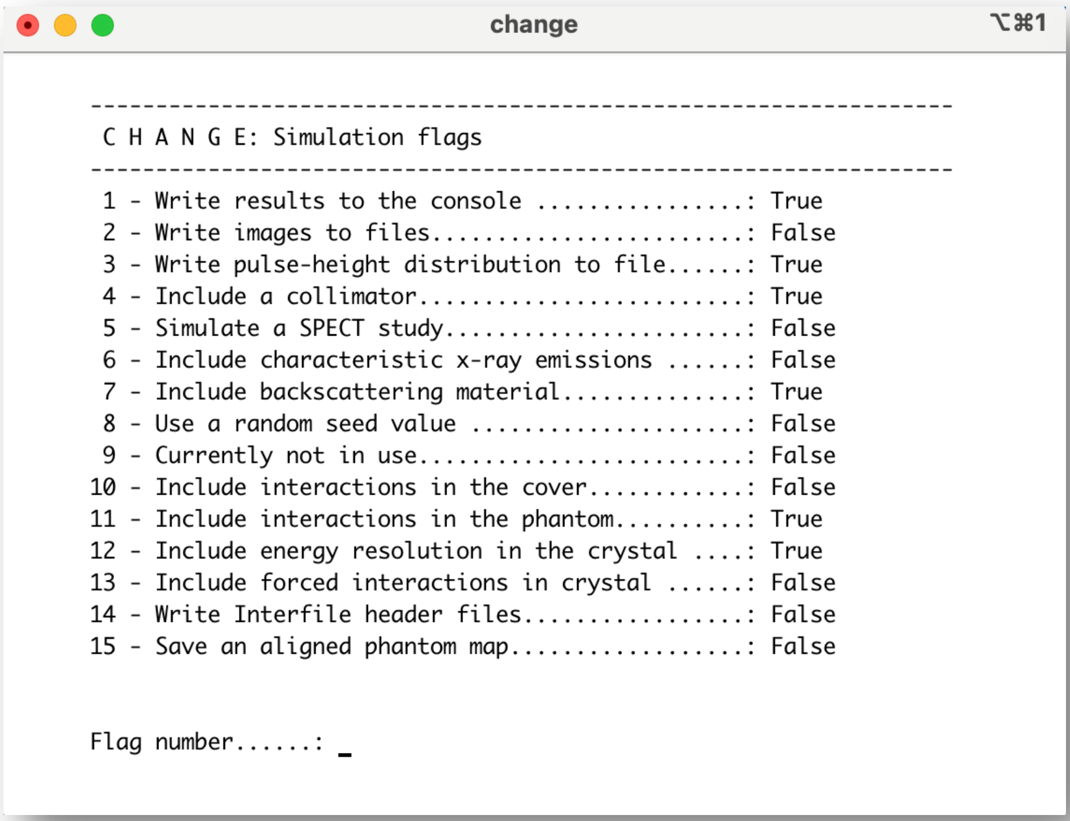
Index-98: Some different energy resolution can be modelled. If the value is 0, then the energy resolution model described by Chen (9) is used. If the value is 1, then the Chen model is used, but it can be modified by the energy resolution parameters provided in the **simind.ini** file. If the value is 2, then a Gaussian model is used. For additional information, please refer to reference (5).

Index-99: The cloud of charge also has lateral mobility when drifting towards the anode. This drift is modelled as a Gaussian function (10). The value entered should be the diffusion constant.

Index-100: Size in I direction of the whole detector (e.g. for full-sized CT cam**eras)**

Index-101: Size in J direction of the whole detector (e.g. for full-sized CT cam**eras)**

Simulation Flags



```
-----------------------------------------------------------------
 C H A N G E: Simulation flags
-----------------------------------------------------------------
 1 - Write results to the console ...............: True
 2 - Write images to files.......................: False
 3 - Write pulse-height distribution to file......: True
 4 - Include a collimator........................: True
 5 - Simulate a SPECT study......................: False
 6 - Include characteristic x-ray emissions ......: False
 7 - Include backscattering material.............: True
 8 - Use a random seed value .....................: False
 9 - Currently not in use........................: False
10 - Include interactions in the cover...........: False
11 - Include interactions in the phantom..........: True
12 - Include energy resolution in the crystal ....: True
13 - Include forced interactions in crystal ......: False
14 - Write Interfile header files................: False
15 - Save an aligned phantom map.................: False


Flag number......: _
```

Option 3 in the main menu is a menu consisting of logical flags that control various operational aspects of `simind`. These flags are Boolean and are set/reset by giving the appropriate flag number. The flags are defined as follows:

Flag-1: Determines whether on-line printout and results will be written to the terminal during the actual simulation. This flag should be turned off when submitting a command procedure to a batch queue.

Flag-2: Determines whether a binary image file (*.bim) that will store a calculated planar image should be created. This flag does not affect the actual calculation of the matrix. The file format for the image is 32-bit float.

Flag-3: Determines whether a binary energy pulse-height distribution file (*.bis) should be created. As for Flag-2, this does not affect the calculation of the different detector parameters that are derived from the energy spectra. The file format for the spectrum file is 32-bit float.

Flag-4: Includes simulation of a scintillation camera collimator. The values in Index-45 to -53 are not in use although they are defined.

Flag-5: Determines if a SPECT simulation will be performed. If this flag is set, then the total number of photon histories in the SPECT simulation will be the number of photon histories per projection multiplied by the number of projection angles. Invoking Flag-5 will set Flag-2 to false because a special SPECT file with the extension *.a00 is produced instead. The file format for the projection images is 32-bit float.

Flag-6: Includes the simulation of characteristic k-α X-ray photons emitted from a photoelectric absorption site in the crystal, the collimator, and the protective cover. However, whether interactions in these different compartments are to be actually simulated depends on what is set in the **`simind.ini`** file located in the smc_dir folder.

Flag-7: Includes a volume behind the crystal to simulate scattering in light guides and PMT. The thickness is given by Index-11. The scatter order is 3 by default, and it can be changed by editing the **`simind.ini`** file located in the smc_dir folder.

Flag-8: Controls the sequence of the random number generator. If it is set to false, then the same initial seed will be used in every simulation. Then, no statistical differences will be observed between different simulations of identical imaging systems. This function can be useful when tracing errors. However, for a normal procedure, the flag should be set to true.

Flag-9: Currently not in use

Flag-10: This option includes the simulation of the protective cover. The cover is defined by its cross-section file (in the main menu of **`change`**) and the thickness of the layer (Index-13).

Flag-11: This option takes into account the simulation of the phantom. The dimensions of the phantom are given by Index-5 to -7 or, if Index-14 is negative, by Index-31 to -38. Be sure to have a valid file name for the non-homogeneous phantom file when setting Index-14 to a negative number.

Flag-12: This flag incorporates the simulation of both the limited energy resolution and the intrinsic spatial resolution in the simulation. The actual values for the system are defined by Index-22 and -23. If this flag is reset, then the energy pulse-height spectra will reflect the imparted energy in the scintillation crystal and not counts from simulated scintillation light that may be affected by statistical fluctuations.

Flag-13: This option forces the photons to interact in the crystal by invoking a 'forced direction' variance reduction technique. It is useful when simulating high-energy photons that are impinging on thin crystals or when simulating a low-density detector. The calculated detector parameter is not affected by this option because the weight of each photon is adjusted for the forced interaction.

Flag-14: If this flag is true, then the files storing the image and the SPECT projection data will have a header of the Interfile version 3.3 format. This format has been developed to establish a mode of communication between different nuclear medicine units. The specific **`simind`** Interfile parameters have a prefix of ;# and are thus treated as comments by an external system.

Flag-15: If this flag is true, then a file of the phantom that is re-sampled to the pixel size and slice thickness for the corresponding SPECT image is stored. The base name of the file is the same as that of the density map file (Index-14 in the main menu). The contents of the file are in units of density times 1000. The extension is *.ict for the actual images and *.hct for the interfile header. Note that these files are only created if voxel-based phantoms are used and Flag-11, which allows for interaction in the phantom, is set to true.

SCORING ROUTINES

The following scoring routines are now included in **`simind`**. For further details, refer to the specific information for the particular routine by clicking the appropriate link to the left.

- 0 = dummy routine.
- 1 = A routine that provides multiple energy window simulations (Scattwin).
- 2 = A list mode scoring routine (Listmode).
- 3 = Forced collimation routine.
- 4 = Separation of events into components (Penetrate).

The ”scattwin” scoring routine

This routine has been developed as an aid to perform multiple window simulations easily without the need for programming. Because Fortran-90 and later compilers allow for dynamical allocatable arrays, multiple windows can easily be defined at runtime. At runtime, **`simind`** looks for a file whose name is the same as that of the input file but with the extension *.**win**. If this cannot be found, then the program looks for the file **scattwin.win**. The structure of the file is relatively simple, as shown in the following example:

```
126.0,154.0,0
92.0,124.0,0
124.0,126.0,0
154.0,156.0,0
124.0,126.0,-4
```

Each row represents a window where the first number is the lower threshold in keV and the second number is the upper threshold. The third number is a flag indicating two different situations.

- If the number is negative, '-n', then this window will add data from the window defined in row 'n' in the window file. This automatically sums two windows. In the example above, data in window 4 is added to the last window.
- If the number is greater than 0, then the scatter images saved for this energy window only contain events from photons scattered with order 'n' . You can then, e.g. investigate the scatter image in an energy window for each order using the following *.win file:

```
126.0,154.0,1
126.0,154.0,2
126.0,154.0,3
126.0,154.0,4
126.0,154.0,5
```

The files created by **scattwin** have the format

```
out_air_w1.a00
out_tot_w1.a00
out_sca_w1.a00
```

where air, tot, and sca stand for 'air','total', and 'scatter', respectively, and 'w1' represents the first energy window. Primary images can be obtained by subtracting the scatter images from the total images. The number in the extension stands for the window number, i.e. the row number in the *.win file.

For easy use of the simind-generated scatter images and for the castor reconstruction program, it is possible to write two keywords in the *.win file depending on whether you select the DW method or the TEW method.

The DEW method: The scatter is estimated on the basis of the previous energy window. For example, assume this energy window file

```
126.0,154.0,0
92.0,124.0.74,dew
```

This now works in such that the scatter file for the second energy window 92-124 is scaled to match the energy window in the previous setting, i.e. the scaling factor will be [154-126]/[124-92]. In addition, the data will be scaled optionally by the constant 0.5. Thus, the total scaling will be [154-126]/[124-92]*0.74. The scaled version will be stored in the file *sca_w2.a00. It is possible to add a new set of window settings and obtain two different scaled scatter files in the same simulation.

The TEW method: This works as follows. Assume an energy window file as in this example:

```
126.0,154.0,0
154.0,158.0,0
122.0,126.0,1.0,tew
```

Now, the scaling assumes that the primary photopeak window is the second previous window setting, i.e. in this case, the first line. The scatter data in the third window will then be scaled by the factor [Data$_2$*W$_2$ + Data$_3$*W$_3$]*0.5 * W$_1$, where W$_1$ = [154-126], W$_2$ = [158-154], and W$_3$ = [126-122]. Here, there is no optional scaling, i.e. the factor is 1.0 but there could have been a value. The triple energy window data will be stored in *_sca_w3.a00.

The BIS file saved by this routine contains spectra for the total photons (spectra 1), primary unscatter spectrum (spectra 2) and scattered photons (spectra 3) and the for each of the scatter orders selected (1,2,3,4 ......) The spectra can be extracted using the bis program, e.g.

```
bis test spectra1/spe:1 # Extracted spectra = the total events
bis test spectra2/spe:2 # Extracted spectra = the primary events
bis test spectra3/spe:3 # Extracted spectra = the scatter events
bis test spectra4/spe:4 # Extracted spectra = the first scatter order
```

The results for each of the energy windows is printed in the middle section of the result file.

Switches

/FW	This switch can be used to define the window file name. For example, if a window file has the name "windows.win", then this file is used if the command is followed by the switch "/FW:windows". Note that no extension ".win" is needed here.

The "listmode" scoring routine

This routine saves information about the photon history in a binary list mode file. The file will have the extension *.lmf. To save disk space, each record of a history in the binary file will have nine 16-bit integer values, one 64-bit float value, and one 8-bit value in the order indicated in the table. To preserve the decimal precision, the floating values of the initial parameters (e.g. ,x,y,z, ..) have been multiplied with a scaling factor before being truncated to integers. Therefore, you should divide the values by the corresponding scaling factors, found in the table below, before using them.

/OU = 0 (default)

1 = x0 * 100 2 = y0 * 100 3 = z0 * 100	integer, 16-bit integer, 16-bit integer, 16-bit	X0,Y0,Z0 is the sampled emission point
4 = xphant * 100 5 = yphant * 100 6 = zphant * 100	integer, 16-bit integer, 16-bit integer, 16-bit	This is the last interaction point in the phantom before leaving the phantom.
7 = xcrystal * 100 8 = ycrystal * 100 9 = zcrystal * 100	integer, 16-bit integer, 16-bit integer, 16-bit	The X,Y,Z position of the centroid of interactions in the crystal. This location can be convolved with the spatial intrinsic resolution if this flag has been selected.
10 = energy in crystal * 10	integer, 16-bit	Units are keV.
11 = photon weight	float,64-bit	Used to add to the energy spectrum and image matrices.
12 = scatter order	integer, 8-bit	If (2) = 0, this is a primary photon that has not been scattered in the phantom.

/OU = 11

1 = xcrystal * 100 2 = ycrystal * 100 3 = zcrystal * 100	integer, 16-bit integer, 16-bit integer, 16-bit	The X,Y,Z position of the centroid of interactions in the crystal. This location can be convolved with the spatial intrinsic resolution if this flag has been selected.
4 = energy in crystal * 10	integer, 16-bit	Units are keV.
5 = photon weight	float,64-bit	Used to add to the energy spectrum and image matrices.

Note: If simulating SPECT, a negative sign of the weight ndicates that a new projection has started and the event record is the first in this projection . Please be aware of this in your reformatting program.

Switches

/OU	If this switch is given, where n is a given number, then different list mode formats can be stored. For the present version, only one format is defined, which is given below.

The ”forced collimation” scoring routine

This is an option of including a forced collimation in the simulation to speed up the simulation. Owing to technical reasons, this option cannot be made as an ordinary collimator routine. Instead, it must be a scoring routine. This means that if the forced collimator option is selected, in the present version, it cannot be combined with other scoring routines, such as `scattwin`.

The principle of the method is that multiple images are created on the basis of the distance between the lower collimator surface and the last interaction point. When the simulation of a certain projection is completed, each of these multiple images is convolved with a Gaussian function with sigma calculated from the distance between the plane for those interaction points and the camera surface. Subsequently, the images are added to a final image.

The following assumptions have been made:

- The ordinary collimator is turned off and the direction of the photon after the last interaction point is directed perpendicular to the crystal surface. The attenuation is then calculated on the basis of the distance from the last interaction point to the surface of the phantom. This means that the difference in attenuation as a function of the azimuthal angle close to the surfaces is not taken into consideration.
- The shape of the Gaussian will be symmetric.
- No backscatter from the volume behind the crystal is included.
- No events from penetration or scatter in the collimator are included.

To run the program, select Index-84 = 3. The default distance between the planes is 1.0 cm and 64 planes are generated. This can be changed by the switch /DP (distance between planes), so if /DP:0.5 is given, then 64/0.5 = 128 planes are used. Finally, the images will be saved to *.bim or *.a00 files.

Note: Because this technically is a scoring routine, it is not possible to combine this feature with the ’Scattwin’, ’Penetrate’, or ‘List Model’ routines in this version.

The ”penetrate” scoring routine

This scoring routine separates the components of an image or spectrum. Further, 18 different images and energy spectra are stored. The file names for the images are stored with the extension *.b01 to *.b18. Different energy spectra are stored in consecutive order in the *.bis file. The scoring routine is run by selecting Index-84=4. The different image files and spectra are separated according to the following table and where events are coming from

*.b01=	All type of interactions.
*.b18=	photons **without** scattering and attenuation in the phantom.
*.b19 =	photons **without** scattering and attenuation in the phantom and geometrically collimated for primary photon energy (indicated by a negative energy in the *.isd file).

The following files are **with no** backscatter from the compartment behind the crystal and show events coming from.

*.b02=	Geometrical collimated **primary attenuated photons from** the phantom.
*.b03=	Penetration of a septa **from primary attenuated photons from** the phantom.
*.b04=	Scatter from the collimator **from primary attenuated photons from** the
*.b05=	phantom.
*.b06=	X-rays from the collimator **from primary attenuated photons from** the
*.b07=	phantom.
*.b08=	Geometrical collimated **from scattered photons from** the phantom.
*.b09 =	Penetration of a septa **from scattered photons from** the phantom.
	Scatter from the collimator **from scattered photons from** the phantom
	X-rays from the collimator **from scattered photons from** the phantom.

The following are **with** backscatter from the compartment behind the crystal.

*.b10 =	Geometrical collimated **from primary attenuated photons from** the phantom.
*.b11 =	Penetration of a septa **from primary attenuated photons from** the phantom.
*.b12 =	Scatter from the collimator **from primary attenuated photons from** the phantom.
*.b13 =	X-rays from the collimator **from primary attenuated photons from** the phantom.
*.b14 =	Geometrical collimated **from scattered photons from** the phantom.
*.b15 =	Penetration of a septa **from scattered photons from** the phantom.
*.b16 =	Scatter from the collimator **from scattered photons from** the phantom.
*.b17 =	X-rays from the collimator **from scattered photons from** the phantom.

The content in the energy spectrum file follows the order of appearance described above, but different spectra are stored in a single file. To access the *.bis file, you need to open the file and read in to a 32-bit float vector dimension with two dimensions according to the spectra (NrChannels,18). The value of NrChannels is defined by Index-80. You can then select the proper energy spectra by using the same numbers as above. For example, the distribution of events from photons scattered in the collimator and in the phantom but without backscattering will be plotted as spectra(*,8).

SOURCE ROUTINES

The following source routines are now included in **`simind`** in addition to the geometrical shapes defined by Index-1 to -5, as described above.

- Index-15 < 0 -> image-based source routine linked by default. The magnitude of the value determines the type of standard image file. For additional information, refer to the next sub-chapter.
- Index-15 = 6 -> myocardial source routine.
- Index-15 = 7 -> multiple source routine.

The ”image-based sources” source routine

This simulates a source distribution in three dimensions by using the information in a file that contains integer matrices when generating the location for the decays in the phantom. With this procedure, very geometrically complex source distributions can be simulated in a relatively easy manner. The dimension of each map is defined by Index-79 and -82 in the **`change`** program. The number of images is unlimited. The sources are normally centred on the origin of the coordinate system but can be shifted by Index-16 to -18 in the **`change`** program. The routine simulates decays randomly within the voxel volume corresponding to the location of the source map voxel. The number of decays from that location is determined from the contents of the source map voxel.

A pixel size (/PX described below) is needed for conversion to physical coordinates because the location to a matrix cell is referred by relative indices (i,j,k). Note that this pixel size has no connection to the pixel size that defines the projections being simulated (Index-28). Depending on the desired direction of the source, the half-axial source length (that determines the slice thickness) is defined by Index-2-4. The routine reads the number of images stored in the input file. The name of the file is defined by Value 14 in the main menu of **`change.`** The default file extension is *.**smi**, but it may be different depending on the selected type of phantom.

A start block can be defined to avoid file headers, and a limited number of images can be defined. The routine scans through the source maps and sums up all voxels values. The sum will then be equal to the number of simulated photon histories. Thus, when simulating voxel-based source distribution, Index-26 in **`change`** is not used. To use the ”Image-Based Routine”, Index-15 in **`change`** needs to be negative.

A tumour file can be read into the source map routine and the count is incremented according to the definition in the tumour file. One tumour is defined by one row in the file as floating-point values separated by a space.

- The first three values (1-3) define the centre of the tumour in pixel units.
- The next values (4-6) refer to the centre location of the tumour also in pixel units.
- Value 7 is the voxel value and is a relative value associated with the main source distribution.
- Value 8 is the density of the tumour. If the value is 0, then the density will be given by the original values in the voxels that the sphere occupies.
- Value 9 and 10 determine the distribution of activity in the tumour. If both values are 0, then it will be a uniform distributed activity. Otherwise, the shape of the activity distribution will change according to a Butterworth filter, where 9 is the cut-off and 10 is the order (steepness of the curve).

Note: Index-26 is not used when simulating source maps. Instead, the number of photon histories is determined by calculating the sum of the voxel values in the source map. This ensures that the proper distribution is sampled. If the number of histories is too low to obtain good statistics in the final projections, then the switch /NN (described below) can be used to easily increase the number of photon histories by multiplication with a factor.

Switches

/PX	Pixel size of the source maps. This switch is needed for proper function [cm].
/DI	The general direction of the source map defined by an integer (0,1 or 2) in the same way as the non-homogeneous phantom (Index-32 in **change**). No switch means a direction when indices i,j correspond to y,z.
/TH	The value sets the slice thickness for the images [cm].
/SB	Start block (512 bytes) when reading source maps from the file. No value means starting from beginning of the file.
/1S	This switch defines the position in the file of the first image to be used. If omitted, the first image is the first image in the file.
/NN	Multiplier that scales the number of counts in the source maps with an integer. This is employed to allow longer simulations with better statistics using the same source map file. Note that the magnitude of the final event in the images does not depend on this value. The idea is that the count levels in the images should reflect the activity level in MBq as defined by Index-25 and measured with an acquisition time of 1 s regardless of the number of photons simulated. Thus, a larger value of NN improves the statistics in the image but the total sum of events in the image should remain the same.
/IF	An input tumour file is read in with a file with extension *.inp. The format is the same as that for the multiple source routine **with the addition** of an optional density value (times 1000 according to the same logic as that of the densities in the *.zub files) as the number 7 in the sequence and two values that define the distribution of the activity in the tumour. If the number 7 is absent or has a value equal to 0.0, then the tumour density will be the same as that of the organ it occupies. Note that if a tumour infiltrates two organs, the density of the tumour will have two values if it is not explicitly defined. There are two types of tumours. Tumour type 1 has the centre in the pixel defined by value 4,5,6 in the *.inp file. If value 1,2,3 is equal to zero, then the tumour will be a single pixel. Tumour type 2 has the centre at the surface of the pixel that is between i,j,x and i-1,j-1,k-1. The minimum size for this tumour is 8 pixels. Note the following options:

1. /IF = The tumour type as defined in the **`simind.ini`** is used for all tumours defined in the *.inp file.
2. /IF:2 = Tumour type 2 is used for all tumours in the *.inp file.
3. /IF:-2 = Tumour type 2 is used for all tumours in the *.inp file but the background activity is set to zero.
4. /IF:2.3 = Tumour type 2 is used with the Zubal background but only the tumour corresponding to row 2 in the inp file is used.

The ”myocardiac” source routine

This is a source routine that simulates a heart consisting of a combination of a half sphere shell and a cylindrical shell, both with the same thickness. The source distribution can very accurately be matched to the physical cardiac insert manufactured by Data Spectrum Inc. A defect can be inserted in the half cylinder at an arbitrary location. The boundaries of the heart are defined by Index-2,3,4 in the **`change`** program.

Switches

/A1	Shift of the heart in the xy-direction (default value 122 degrees)
/A2	Shift of the heart in the yz-direction (default value 52 degrees)
/A3	Shift of the heart in the zx-direction (default value 0 degrees)
/L1	Location of defect 0 = ant, 180 = inferior, -999 = No defect
/L2	Angular size of the defect (± degrees) (default value 30 degrees)
/L3	Start from Base (default value 0.6 cm)
/L4	Extent of defect i axis direction (default value 2.0 cm)
/L5	Transgression in % (default value 1.0 = full transgression)
/L6	Activity ratio in defect (default value 1.0 = same as surrounding activity)
/M1	Thickness of the myocardial wall (default value 1.0 cm)
/M2	Thickness of the plastic wall (default value 0.2 cm)
/M3	Total length of the chamber (default value 8.0 cm)
/M4	Total diameter of the chamber (default value 6.1 cm)

The ”multiple spheres” source routine

This source routine simulates a Jaszczak cylindrical uniform source phantom (Data Spectrum, Inc., Chapel Hill, North Carolina). The sources must be defined in an input text file with the extension *.inp. The routine first searches for a file named `input.inp`, where **input** is the base name of the input smc file created by **`change`**. If this does not exist, then the routine attempts to open a default input file `jaszak.inp`. If this is not found, then **`simind`** is terminated.

The input file: This file is an ASCII file where each row defines a sphere. Float 32-bit values separated by a comma are used. The values at the first three positions define the size of the source. The values at positions 4-6 are the location in the coordinate system for the centre of the source. Note that this is not the centre of gravity but the starting point for each of the axes that define the ellipsoid. The value at position 7 in the row is a relative value of the desired activity concentration (MBq/cc). The program normalizes the decays in each source so that the number of photon histories per projection provided by Index-26 in **`change`** can be simulated. However, if these values are **negative**, then they are treated as absolute activity concentrations in units of MBq/cc. The total activity (Index-25) is then recalculated to match the sum of the activities for all spheres and, if defined, also together with the background activity.

The value at position 8 determines whether the source shape will be

- a sphere (0.0)
- a cylindrical horizontal rod (1.0)
- a rectangular horizontal rod (2.0)
- a hexagonal horizontal rod (3.0)
- a cylindrical vertical rod (4.0)
- a cone (5.0)

As an example, consider the following two lines in the input file.

```
1.0,1.0,1.0,0.0,0.0,0.0,1.5,0.0
2.0,2.0,2.0,3.0,0.0,0.0,3.0,1.0
```

The first source is spherical with a radius of 1.0 cm and is centred at the origin. The relative activity concentration is 1.5. The second source is cylindrical with a radius of 2 cm and the source is shifted 3 cm towards the positive x-axis. The relative activity concentration in this source is twice as much as that in the first source.

Switches

/BG	This switch defines the background activity concentration in the same units as the concentration defined for sources in the input file.
/HO	This switch generates hot spheres, i.e. no background activity and only activity in the sources defined in the input file.
/CO	This switch results in a cold sphere simulation where a uniform distribution of decays is simulated by information from Index -2 to -4 and -15 in **`change`** but with the exclusion of the decays within the spheres, as defined by the input file. The activity concentration has no meaning here.
/IF	A file name for the input file with the different source definitions can be given. The extension *.inp is assumed.
/DI	Change direction of the whole source distribution. Values can be 0,1 or 2

If the

RUNNING SIMIND

A **`simind`** command at the operating system prompt consists of the following parts:

- The program name, **`simind.`**
- An input file (optional).
- An output file (optional).
- Control switches (optional).

The example below considers a Windows operating system with possible differences from other operating systems in the prompt and the upper case of the characters

The input file in the command line above, `input.smc`, defines a file created by **`change`** that contains the data for the particular simulation. No extensions are necessary because **`change`** always creates files with the *`.smc` extension. The output file(s) created by **`simind`** stores basic results from the simulation, a binary image, and an energy pulse-height spectrum. If not explicitly given, the output base file name, `output`, will be the same as the input file name, but different files are separated by different extensions. If no input file name is given, the default file, **`simind.smc`**, will be used and the base name of the output file will also be **`simind`**.

Example of the program running on a PC

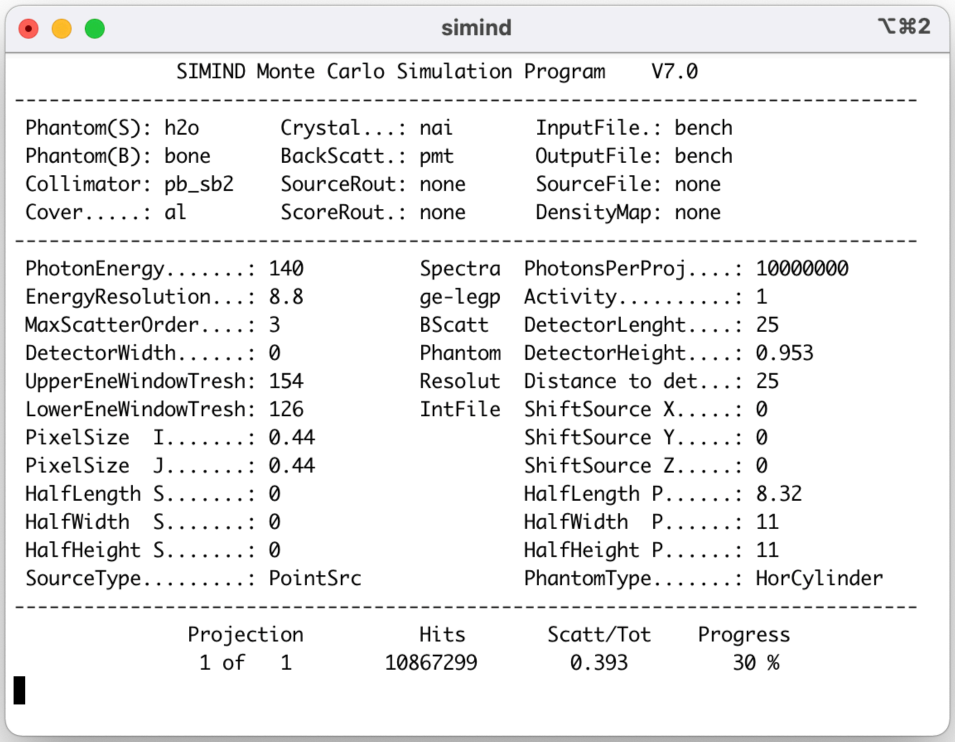
The user can control the input parameters created originally by **change** at the command level. This means that a single data file (e.g. `input.smc`) can be used for multiple simulations. Each value given by an index or by a flag in **change** can be overwritten by a matching control switch. By giving a two-character alphanumeric corresponding value, this new value takes precedence over the value defined by **change.** Additional characters may be given for clarity but only the first two are used. For example, overwritten the value in the input file `input.smc` to change the number of histories (Index-26 in **change**) to 10,000 is accomplished by the following command:

```
simind input output/26:10
```

The character ":" or "=" can be used to separate data from the switch name. Note that Index-1 to -9 must be assigned 01-09 in order to fulfil the minimum two-character definition of a switch. The two switches "TR" and "FA" (true and false) are used to control the simulation flags defined in **change.** The following command can be entered to reset Flag-1 (indicating no output to the screen) and set simulation Flag-5 (invoke a SPECT simulation):

```
simind input output/fa:1/tr:5
```

The control switches may be placed arbitrarily together with the program name, the input file, or the output file. Input files, output files, and program names are separated by spaces. Trailing spaces are removed.



```
               SIMIND Monte Carlo Simulation Program    V7.0
------------------------------------------------------------------------------
 Phantom(S): h2o       Crystal...: nai       InputFile.: bench
 Phantom(B): bone      BackScatt.: pmt       OutputFile: bench
 Collimator: pb_sb2    SourceRout: none      SourceFile: none
 Cover.....: al        ScoreRout.: none      DensityMap: none
------------------------------------------------------------------------------
 PhotonEnergy.......: 140          Spectra  PhotonsPerProj....: 10000000
 EnergyResolution...: 8.8          ge-legp  Activity..........: 1
 MaxScatterOrder....: 3            BScatt   DetectorLenght....: 25
 DetectorWidth......: 0            Phantom  DetectorHeight....: 0.953
 UpperEneWindowTresh: 154          Resolut  Distance to det...: 25
 LowerEneWindowTresh: 126          IntFile  ShiftSource X.....: 0
 PixelSize  I.......: 0.44                  ShiftSource Y.....: 0
 PixelSize  J.......: 0.44                  ShiftSource Z.....: 0
 HalfLength S.......: 0                     HalfLength P......: 8.32
 HalfWidth  S.......: 0                     HalfWidth  P......: 11
 HalfHeight S.......: 0                     HalfHeight P......: 11
 SourceType.........: PointSrc              PhantomType.......: HorCylinder
------------------------------------------------------------------------------
               Projection         Hits       Scatt/Tot     Progress
                1 of    1       10867299       0.393          30 %
█
```

Limitations for Linux/UNIX/MacOS

There will be a conflict between the slashes used by **simind** and the slashes used to define a certain directory in Linux/UNIX/MacOS. **simind** cannot distinguish between a slash that points to a switch and a slash that is part of a UNIX file directory. To overcome this problem, the directory name can be specified using two backslashes '\\'. **simind** will, after being decoded for control switches, change these backslashes to ordinary slashes '/' so that the result files can be stored in other directories.

It is also not possible to add control switches directly to the program name in Linux/UNIX. The operating system assumes that this will be the program name. Thus, control switches on a Linux/UNIX/MacOS machine should be either added after the input or output file or separated by spaces. All file names are converted to lower case to simplify the coding and comparability for different operating systems.

Basic runtime-switches for **simind**

/CC	Collimator code in a format according to the description in the **change** program. Note that you should not include the '*', i.e. the character to be used as a prefix in the **change** program. For example, **simind** input/cc:ge-legp gives this collimator in the simulation regarding what is given in the smc file [character].
/DF	This switch is used when simulating a heartbeat and/or respiratory movements using MCAT/NCAT/XCAT phantom. If the base name is **test**, then by adding the switch /DF:3, **simind** will try and open a file named **test_atn_3.bin.** If this switch is not given, then it will open the file **test_atn_avg.bin**, which is the standard name for the average density image.
/ES	An energy offset relative to Index-1 and expressed in keV for the centre energy of the photo peak window. A negative value means a lower value relative to Index-1. If this shift is not given, then the centre value of the energy window is defined by Index-1.
/FE	The name for the energy resolution for file can be given as the parameter. Give the name without the extension 'erf' [character]. If only /FE is given it is assumed that the file name definied in the main menu is used. This switch also overrides the value in index 22.
/FZ	The name for the Zubal file can be given as the parameter. Give the name without the extension 'zub' [character].
/FI	The name for the Input file can be given as the parameter. Give the name without the extension 'isd' [character].
/FD	The base name for the density map (*.dmi) or the NCAT attenuation map (*.bin) [character].
/FS	The base name for the source map (*.smi) or the NCAT activity map (*.bin) [character].
/I2	Image files are stored as 16-bit integer matrices. By default, a scaling factor is calculated from the first projection so that the maximum in that projection will be 1000. If a data value is gives after /I2, then that value will be used as a scaling factor. For example, /I2:150 scales the data by a value of 150. If the value is negative, Poisson noise is also added to the projection data.
/IN	With this switch, you can change any value in the **simind.ini** file at the execution level. The switch contains two parameters. The first is the ini number, which is given in the **simind.ini** file. The second value is the actual value that is to be changed. For example, /IN:x4,8x sets the maximum scatter order in the backscatter layer to 8. Here, 'x' is used as brackets because real brackets do not work in the UNIX environment.
/LO	The number of photon histories before printout of the on-going status line. Requires a data value. The default value is 1000.

/LF	A sampling technique based on linear sampling of the polar angle for the photon direction is used. Note that this is only useful when simulating the collimator routine that allows for penetration (Index-53 = 1).
/MP	When running **simind** with MPI, this switch allows a parallel run of all projection angles at the same time. However, this works only if the number of MPI nodes equals the number of projections.
/OR	Parameter that changes the orientation of the density map. Value 2 transposes the maps, value 3 swaps the maps such that Index-(1,1) is at the lower left, and value 4 both transposes and swaps.
/PR	Start simulation at projection number given by the switch. Default is projection 1. If the value is negative, then this indicates the stop projection number. The order of positive and negative numbers is of no importance.

/PU	In some cases, it is more convenient to express the centre of the source in terms of pixel units instead of cm and let the computer calculate the physical centre position. By using the /PU (= pixel units) switch, one can give the shift of the source in pixel units according to
	`simind input/16:38/17:128/18:64/PU`
	From the phantom length and the pixel size for the density maps, the pixel size is calculated as the true source shift in cm. If the command looks like
	`simind input/16:38/17:128/18:64/PU:0.25`
	then the position is scaled down by a factor of four. This is used when the pixel unit is valid, e.g. a 512 matrix, but the actual density map used in the simulation is only a 128 matrix.

/QF	Quit the simulation if an earlier result file exists. This flag is useful if a batch simulation has been going on and the computer has been restarted following a power failure. This may then prevent recalculation of already existing data files. Note that for this to work as intended, there should not be any *.res before the start of the simulation.
/RR	The random number generator is based on seeds that are stored in *.num files in the smc_dir directory. When **`simind`** starts, it reads the seed, and when the simulation ends, the last seed is stored in the files for the next run. However, when running on clusters or machines with multiple cores, multiple runs may read the same seed and thus duplicate the results. If different integer values are set for this /RR switch, random numbers are created at the very beginning of the run on the basis of this value. This ensures that when the actual simulation begins, the random numbers will be different. The most convenient way to use this switch is of course when defining the run in a script.
/SC	The maximum number of scatter orders allowed in the phantom. Requires a data value. A large number requires a longer computational time because more scattered photons will be followed in the phantom. In some cases (e.g. when simulating images obtained in a narrow energy window), a lower value may be sufficient because the energy window rejects the lower energy photon with a low probability. Therefore, the user must be careful so that the proper number of scatter orders is simulated for the particular simulation. It may be necessary to perform some 'trial-and-error' simulations here to fully understand how /SC works. The default value is 3 and this value can be changed in the **`simind.ini`** file. If the scatter order is a negative value, only the scatter distribution is simulated and stored in the images and spectrum.
/SF	This switch is used when simulating a heartbeat and/or respiratory movements using MCAT/NCAT/XCAT phantoms. Because each segment from MCAT/NCAT/XCAT has its own image file with a particular number, this switch makes a patch to the file name of the density file. For example, suppose that you have generated 8 segments of a beating

	heart using the base name **test**. The output from MCAT/NCAT/XCAT software gives you files named `test_act_1.bin, test_act_2.bin,....test_act_8.bin`. If you add the switch /SF:3, then you will use the source map `test_act_3.bin` even if the input file says test. If this switch is not given, then it will open the file `test_act_avg.bin`, which is the standard name for the average source image from MCAT/NCAT/XCAT.
/TS	A time shift can be added to the date and time description in the interfile header. The initial time and date is set in the **simind.ini** file and the value of the /TS switch is the extension of this time expressed in hours. If the value is negative, then the activity value (Index-25) will be reduced by a factor corresponding to the physical decay of the time shift. This is done only if the half-life value (expressed in units of hours) is provided in the *.isd file used. This option does not work for single photon energies. All the *.isd files included in the **smc_dir** have information about the half-life. This switch is also used when simulating dynamic imaging. For more information see section 'Dynamical Imaging'
/UA	Set the density equal to the data buffer or if not present equal to 1.0 (= water). This makes the phantom equal to uniform attenuation regardless of the initial voxel value in the phantom.
/WB	Whole-body simulation of anterior and posterior views. Note that you might need to adjust the matrix size and detector length (Index-8). When this switch is given, the following parameters are set. Flag 15 - Interfile = .true. Flag 5 - SPECT simulation = .true. Flag 11 - Phantom flag = .true. Number of projections = 2 Rotation type = 0 (360 degrees) Matrix size = n*512 x n*192 where n is the value of the WB Switch Pixel size = initial pixel size/n (defined by Index-28 in **change**) In the simind.ini file, there is a flag T/F that control the format of the aligned attenuation map defined by Flag-15. If the flag in simind.ini is set to T, then one planar scout file with the dimension [Index-76 x index-77] is created as an aligned attenuation map. The numerical values are the projected my*d. If the flag is set to F, then the aligned file is defined as transversal slices with the number of slices equal to Index-77 and the matrix size (square) defined from Index-76. The value in the simind.ini file can be changed using a negative sign in the /WB switch. For example, if the value in **simind.ini** is set to T, then /wb:-2 will **change** it to F (or vice versa).
/Xn	'n' specifies the cross sections (1-6) that can be changed. /X1 to /X5 refer to the values 9 (soft), 10 (bone), 11(cover), 12(crystal), and 15 (backscatter), respectively, in the main **change** menu. /X6 is the material for the collimator routine (if Index-53 = 1). The default value for /X6 is defined in **simind.ini** [character].

Hidden flags

This table summarizes the 'hidden flags' in the **change** data. The negative sign of an index value is used here as a flag and, in some cases, can also be used as the actual value. A negative value of the following indices indicates:

1	Call to the **isotop** routine for the emission photons. The default routine linked is a spectra routine that reads information about energy and abundance from a text file. The default name for the file is spectra.isd. However, if there is a file that has the same

	base name as the input SMC file but with extension *.isd, then this file will be used. The user can also specify the name using the switch /IF:
14	Simulation of a non-homogeneous voxel-based phantom. There are three defined phantoms available (-2 to -4). Depending on the value, different types of voxel-based phantoms are used. **`simind`** is written such that the XCAT phantoms are easy to import but there is, owing to license reasons, no XCAT phantom in the **`simind`** distribution.
15	Call to the **source** routine. Depending on the value, different types of voxel-based phantoms are used.
19	Photons are emitted within a solid angle defined by the absolute value of the index. A value of 0 indicates a narrow-beam simulation. Note that when this option is selected, some of the results in the *.res files are not defined (e.g.the cps/MBq that goes to zero).
20,21	Treat the absolute value as a percentage FWHM of the energy resolution relative to the photon energy in Index-1 and calculate the thresholds of the energy window from this value.
22	If positive, then the energy resolution is simulated with an dependent proportional to $1/\sqrt{E}$. If negative then it is a
25	If the activity is set to zero, then no reduction in the value owing to decay is calculated when using this in combination with the TS switch. Otherwise, the value is changed on the basis of the half-life provided in the isotop file (*.isd) if this value is present in the file.
33	A negative value indicates the start block instead of the start image when reading from the density map.
41	Non-uniform rotation. The value determines the 'security distance', i.e. the minimal distance between the surface and the collimator.
85	If a negative sign is given, then a space is written to the CSV file after the data have been written. This is useful to separate the data row into groups. Use this in a switch value in a script file rather than defining it in the SMC file. Otherwise, a space is written for each simulation.

Dynamic simulation

There are several requirements that need to be fulfilled in order to simulate dynamical imaging.

1. This only works with an image-based phantom that is based on codes. For additional details about these phantoms, please refer to the information related to Index-14 and Index-15.
2. For each of the sources having an activity distribution that will change dynamically, a time-activity curve (TAC) needs to be available in a data file and in a specific file format, which is described below.
3. The actual time sequence is provided by the switch /TS. The value for this switch is a character string and its syntax depends on the type of dynamic simulation. Note4 tat the character x below shall be understood as a delimiter. The natural delimiters [,(,{ cannot be used since these have special meanings in the OS systems.

    a. /TS:A – This does nothing other than add a starting time to the interfile header file that is A hours longer that the reference time (injection time) that is specified in the simind.ini file. (Works the same for Planar/SPECT)

b. /TS:xAx Initiation of a sampling from the TAC files, that needs to be in the zub file. The value represents the administered activity. For planar **simind** mode, the sample represent the starting time of the measurement (Works the same for Planar/SPECT)

c. /TS:xA,Cx - Same as above but now together the time_shift, defined by the value C, is added to the starting time (Planar/SPECT)

d. /TS:xA_Bx - A serie of dynamic planar images where A is the frame time (seconds) and B is the number of images. (Planar only). An example is /TS:x10_100x = 100 frames with each having a time of 10 seconds.

e. /TS:xA1_B1_A2_B2x - A serie of dynamic planar images using groups of sequences. An example is /TS:x10_100_20_50_100_10x = 100 frames with each having a time of 10 seconds followed by 50 frames with acquisition time of 20 seconds followed by 10 frames with 100 seconds acquisition time. (Planar only)

f. /TS:xA_B,Cx - Same as (d) but with a time shift C before the start of the acquisition (Planar only)

g. /TS:xA1_B1_A2_B2,Cx - Same as (e) but with a time shift C before the start of the acquisition (Planar only)

The file format for the TAC is as follows:

1. 16-bit integer	Version number
2. 16-bit integer	Curve units 0=relative values, 1 = MBq/ml , 2 = MBq
	Time units 1 = seconds, 2 = minutes, 3 = hours, 4 = days
	Number of points
	Total voxels in the volume
3. 32-bit float array	an array of 'float' values for the time points and time steps indicated with a negative number.

The first value in the array of time points must be negative and the magnitude of the value reflects the time step for the TAC values that follow. If a negative number is found as the next number in the time point sequence, then the time step is changed to this value. Thus, it is possible to have a TAC file with different time bins. The actual value is interpolated to the actual time in the /TS switch (which is the mid-point in the time interval).

When an image for a new time point (defined by the /TS switch) is simulated, **`simind`** creates a new activity map from the information in the code file. To connect to the TAC file in order to obtain the proper activity distribution, the format for the code file (*.zub file) needs to be according to that of the following example (using the xcat phantom).

```
hrt_myoLV                  1  1040   1,hrt_myolv.tac
hrt_myoRV                  2  1035   1,hrt_myorv.tac
hrt_myoLA                  3  1053   1,hrt_myola.tac
hrt_myoRA                  4  1048   1,hrt_myora.tac
body                       9  1000   1,body.tac
muscle                    10  1050   1,muscle.tac
brain                     11  1040   1,brain.tac
sinus                     12  1000   1,sinus.tac
liver                     13  1060   1,liver.tac
gall_bladder              14  1026   1,gall_bladder.tac
l_lung                    15   260   1,l_lung.tac
r_lung                    16   260   1,r_lung.tac
```

```
esophagus                   17  1000   1,esophagus.tac
laryngopharynx_activity     18  1000   1,laryngopharynx_activity.tac
```

The difference here, as compared to the template file **vox_man.zub** located in the smc_dir folder, is the kinetic TAC file. The number that precedes the file name must be there, and it can be used as a scaling factor. The extension needs to be 'tac'. For those organs that do not have a TAC associated to it, a fixed activity value based on the number in the fourth column will be used (standard code usage).

MPI (Message Passing Interface) simulation

The MPI [46] is a protocol for message passing and allow a program to split itself into parallel version to run on a cluster (https://en.wikipedia.org/wiki/Message_Passing_Interface). The system is based on standardized calls between parallel process making it possible to with a program split itself into version and send/receive data dusing a run. In SIMIND, MPI parallelisation has been implemented to improve the statistics of the simulated images and energy spectrum for a given calculation time. The MPI system works as follows. SIMIND is replicated to run in parallel on separate nodes automatically, where one node is selected to be a primary node that controls the execution of the others at various positions in the code. When simulating SPECT, and when the first projection has been completed, all nodes are temporarily stopped. When the master node has verified that all nodes have reached that location in the code, the primary node collects specific data from all running nodes and adds them to its own data (e.g. projection images, energy spectrum, etc.). The summed data are, if necessary, stored to a file. The primary node then releases all nodes for the simulation of the next projection.

MPI nneds to have a special installation (see Installation procedure). There are different installations for Linux/Windows and for MacOS. For macOS versons, simind_Ämpi have been compiled with the openmpi library. For the Linux distribution, simind_mpi is compiled with the Intel MPI system. Windows-based simind_mpi are currently not available

Some information about scripts

Monte Carlo simulations are by default very time consuming and therefore it is very convenient to learn some basic things about how to use scripts in Windows and UNIX. A script can be very advanced and nearly like a small program, but in its simplest form, it is a file that contains a series of commands that could be given at the command line prompt. The reason for scripting in Monte Carlo simulations is to set up a series of simulations that will start without the need for keyboard commands. Usually, in an MC projection, one investigates some behaviour by varying one parameter. The reason **simind** has the structure for execution is that this is very convenient for scripts. For example, if one wants to investigate the spatial resolution in an image (FWHM) as a function of five different distances between the source and the camera, one could in principle create five **change** files (*.smc) where Index-12 is set to the five different distances. However, for a large project, this becomes difficult to maintain. Therefore, the ability to input changes in the *.smc file at the execution stage becomes important. Suppose that we want to study the distances 5, 10, 15, 20, and 30 cm. Then, we can write a script file that contains the following lines.

```
simind camera result1/12:5
simind camera result2/12:10
simind camera result3/12:15
simind camera result4/12:20
simind camera result5/12:30
```

In Windows, the script file must always have the extension `.bat` but in UNIX there is no such requirement. I prefer to use `.csh` because I use the tcsh shell.

Thus, if the file `run.bat` (or `run.csh`) contain the above lines, you can use the following commands to start the simulation:

```
c:\simind> run.bat          (Windows)
simind# csh run.csh         (Linux/MacOS)
```

Note that the name of the result file must be unique (result1, result2, etc.); otherwise, the files from a previous simulation will be overwritten. The switch /12 refers to Index-12 in **change**, so regardless of what it is set to in the camera.smc file, the value will be changed to the one given by the /12 switch. All indices in **change** have a corresponding switch.

An advantage of scripting is that it is possible to pass information from the command line into the script without editing the script itself. One can define command line parameter as numbers where the number matches the position of a sequence of characters in the command. For Windows, the command variables are then defined by a `%` character while in UNIX/Linux/MacOS, it is the `$` sign. Suppose that the script file (for Windows) looks like this instead.

```
simind camera result1%2%3/12:5%1
simind camera result2%2%3/12:10%1
simind camera result3%2%3/12:15%1
simind camera result4%2%3/12:20%1
simind camera result5%2%3/12:30%1
```

If the command now looks like this:

```
c:\simind> run.bat /01:200        (Windows)
$HOME/simind# csh run.csh /01:200 (Linux/Mac)
```

then `/01:200` will be inserted at the place where `%1` is, i.e. at the end of the command. This means that I now overwrite Index-1 in the **change** file and instead simulate a photon energy of 200 keV. Thus, we do not need to edit the run.bat file nor do we need to modify the camera.smc file. Now, we have a problem because the result files will be overwritten if we use this for different energies. Therefore, we modify the command to look like this:

```
c:\simind> run.bat /01:200 200kev     (Windows)
$HOME/simind# csh run.csh /01:200 200kev  (Linux/Mac)
```

Note that ”200kev” appears at the second position after the script file name and therefore the ”200kev” characters will be inserted at the position corresponding to `%2`. Note that `%3` is not used and therefore nothing is inserted at this position. The result name will have the format `result1_100kev.res`. However, parameter `%3` can be useful in the following example.

If one creates a new file called run1.bat, one can define commands in it that start the file run.bat for different scenarios.

```
call run.bat /01:100/fa:12 _100kev _nores
call run.bat /01:200/fa:12 _200kev _nores
call run.bat /01:300/fa:12 _300kev _nores

call run.bat /01:100/tr:12 _100kev _res
call run.bat /01:200/tr:12 _200kev _res
call run.bat /01:300/tr:12 _300kev _res
```

This example uses `run.bat` to simulate five distances and photon energies with and without energy resolutions. The third parameter is used to get unique file names and the first parameter is used both to set the photo energy and to turn on or off the energy resolution (Flag-12 in **change**). We now have a case where one script calls another script. In Windows, we use `call` for this purpose. In UNIX, we can instead use `csh` to call another script. The result name will have the format `result1_100kev_nores.res;` the underscore character makes it easier to understand the name of the result file.

Depending on whether it is Windows or UNIX or `sh,csh,tcsh,` or `bash` shells in UNIX, there is a syntax for performing a really complicated script involving loops, if-then-else statements, etc., but it is beyond the scope of this manual.

The “bim” program

The **bim** program is a simple utility program that performs certain calculations on the created projection image in the *.bim file. The program is invoked in the same manner as **`simind`**, i.e. giving an input file name and additional switches after the program name.

```
bim input output/switches
```

The default input file name is **`simind`.bim** and the output is **`simind`.prn** if no base name for the output file is given. BIM outputs a very simple 'image', based on keyboard characters, of the matrix together with calculated data and a two-row data of a profile through the matrix. The profile can then be plotted or included in a spreadsheet program for further analysis. Wildcards are allowed according to the rules described in Option 10 in the **`change`** description.

Switches

/HO	Horizontal profile (default).
/-F	Do not save result in file with extension 'prn'.
/VE	Vertical profile.
/-O	Do not write results to the screen.
/AD	Number of added lines centred around /LI.
/LI	Select a line number for the profile.
/SM	Smooth matrix with a filter.
/1R	Save only y-values in the prn file.
/2R	Save only x- and y-values in the prn file.
/FW	Calculate Full Width at % Maximum (def = 50).
/II	Read I*2 matrix.
/IA	Read I*2 matrix and remove sign.
/IU	Read a I*2 matrix unsigned (0-65535).
/OI	Write the matrix as a I*2 matrix with extension 'int'.
/OU	Write the matrix as a I*2 matrix norm to 65534.
/OR	Write a R*4 matrix with extension 'mtx'.
/+*-	Algebra with matrices.
/YP	Addition value to y-values (matrix).
/MA	Addition slice from start image.
/UT	Upper threshold limit (default = 100%).
/LT	Lower threshold limit (default = 0%)
/NP	Normalize slice peak to value.
/NA	Normalize slice to slice area times value.
/LG	Calculate logarithm log(mtx(i,j)+1).
/IM	Starting with image number.
/LO	Process a number of images.
/PL	Simple plot on the screen.
/SB	Start at block when reading a file.
/XP	Add value to x-values (Slice).
/YM	Multiply value to y-values (Matrix).
/NT	Normalize slice to mtx area times value.

/64	Output only matrix. No profile data.
/CA	Calculate region of interest.
/SW	Swap bytes in 16-bit integer input array.
/XM	Multiply value with x-values (Slice).

The "bis" program

The bis program works on the calculated energy spectra from **simind**. Its main purpose is to generate a two-column ASCII string of the spectra to be used for plotting or printing. The default format is FORMAT(2F12.3) but it can be changed if the source code is available. The program is invoked in the same manner as **simind**.

```
bis input.bis output.prn/switches
```

The default input file is **simind.bis** and output is **simind.bis** if no base name for the output file is given. Wildcards are allowed according to the rules described in Option 10 in the **change** description.

Switches

/BI	binary from ASCII PRN -> BIS
/CO	Convolve with Gaussian. FWHM = 'value'*0.1
/+*-	Algebra with files
/SM	Smoothing with filter = 'value'
/SP	Convert Spectrum Number
/LT	Lower Limit Integration (CHANNEL)
/UT	Higher Limit Integration (CHANNEL)
/AS	ASCII from BINARY BIS -> PRN (default)
/NP	Normalize Peak to 'value'
/YA	Addition 'value' to y-values (Slice)
/YM	Multiply 'value' to y-values (Slice)
/XA	Addition 'value' to x-values (Slice)
/XM	Multiply 'value' to x-values (Slice)
/H	Help file (this table)
/UT	Higher Limit Integration (CHANNEL)
/NR	Number of channels in the BIS spectra
/OY	Generate y-values only
/MX	Maximum number of data points
/HI	Generate a histogram in the prn file
/-O	NO output to terminal
/CH	New keV/channel
/NA	Normalize Area to 'value'

The "smc2castor" program

By nature, **simind** is a virtual SPECT camera. This means that it only creates projection images. No reconstruction software is included or supported. This has been a source of questions because one wants to reconstruct data and stand-alone software is not always available. Although it is possible on some clinical systems to import the Interfile file, it is often difficult and requires a deeper understanding of the Interfile file format and how to make **simind** to write such a file properly. In March 2017, the Castor reconstruction software was released for download in the public domain. Information about the program can be found by visiting http://castor-project.org/. However, the file format for this reconstruction software was not based on the Interfile, at least with regard to the input data. Therefore, I have written a conversion program called smc2castor, which makes it easy to reconstruct **simind** simulated projections. The general syntax for the program is

```
smc2castor smc_spect_projs.h00 result_file [scatter file]/switches
```

where smc_spect_projs.h00 is the header file for the **simind** SPECT simulation and result_file is the result file for the reconstructed data. The smc2castor program creates a castor file and can spawn a reconstruction by adding the switch /ru. It is assumed that the castor software has been installed properly according to the instruction. The CASTOR_CONFIG environmental variable is set to the castor config folder, and the executables are accessible from the path.

The following switches are available:

/AT	Include attenuation correction
/HE	Help
/IT	Number of iterations (default = 2)
/IT	Number of iterations
/SI	Save all iterations
/MB	Activity in the phantom (default = 1 MBq)
/PR	Projector
/RU	Run the castor reconstruction
/SF	Scaling factor scatter (default = 1.0)
/SU	Number of subsets (default = 8)
/TI	Time per projection (default = 1 s)

INFORMATION OF AVAILABLE VOXEL-BASED PHANTOMS

The following phantoms are available as binary images in smc_dir.

vox_man1.dat	128×128×243 integer, 8-bit	Original Zubal torso coded phantom with no arms and legs. *Note: This is used when then* ***change*** *Index-14,15* = -2
vox_man3.dat	192×96×498 integer, 8-bit	Modified Zubal phantom with added legs and arms, which is straightened out and aligned along the body to be useful for whole-body imaging. Note: This is used when **change** Index-14,15 = -4.
vox_brn.dat	256×256 integer, 16-bit	Original Zubal brain phantom file. Note: This is used when **change** Index-14,15 = -3
alders.dmi	64×64×60 integer, 16-bit	A CT scan of an Alderson phantom. The density is scaled by a factor of 1000.
rod_cyl.smi	512×512×1 integer, 16-bit	Cylindrical CT-based hot-source map of a Quality Control pie phantom.
rod_ell.smi	512×512×1 integer, 16-bit	Elliptical CT-based hot-source map of a Quality Control pie phantom.
rod_ellh.smi	256×256×1 integer, 16-bit	Elliptical CT-based hot-source map of a Quality Control pie phantom.
rod_ellc.smi	256×256×1 integer, 16-bit	Elliptical CT-based cold-source map of a Quality Control pie phantom.

The Zubal phantom
The phantom used in this site has been developed by George Zubal, Ph.D., Dept. of Diagnostic Radiology, Yale School of Medicine, 333 Cedar Street, New Haven, CT 06510. The phantoms

included are located in the smc_dir directory and have the names `vox_man.dat,vox_brn.dat` and `vox_man3.dat`. The main references are (11-13).

The NCAT/XCAT phantom
This phantom is provided by Dr Paul Segars from Duke University. The phantom is actually a software package that creates attenuation and activity maps from a user-specified input file, which are is used as input in simind. Because this family of phantoms have proven to be very useful, the simind program has routines that makes it easy to input such data. Additional information about the phantom can be found in references (14-16).

DESCRIPTION OF SOME FILES OF IMPORTANCE

The simind.ini file
`simind` initially reads a file, called **`simind.ini`**, that is located in the **smc_dir** folder. This file includes default values for the program, but it is also possible to some extent to control the physics used in **`simind`**. Each line has a number that can be used to change the default value to a new value at runtime using the new switch /IN.

The error-file
This file includes the error text displayed by **`simind`**.

The energy resolution file
This file includes paramaters by **`simind`** to model the energy resolution (in keV) as function of energy based on a curve with three parameters. Currently two functions can be used and which of these two is defined in the file. The review the file **simind.erf** template in the smc_dir folder. Thois function can be initiated by different ways. The file name can be defined in the change main menu. If index 22=0 then the fitting will come in use. An alternative is to use the flag /FE:filename. By adding this switch, the fitting is used based on the parameters in the filename (which override the filename in the change program).

The **simind.erf** template
```
# -------------------------------------------------
# ERMODEL: 0=1/SQRT(Energy)
#          1=ER=ER_A + ER_B * SQRT(Energy + ER_C)
#          2=ER=ER_A + ER_B * SQRT(Energy + ER_C*Energy**2)
#          3=ER=ER_A + ER_B * EXP(ER_C * Energy)
#          4=ER=ER_A + ER_B * Energy
#          5=ER=ER_A + ER_B * Energy + ER_C * Energy**2
#
#       In 1-5, ER is in unit of keV
#
# -------------------------------------------------# Template camera
ER_MODEL=2
ER_A=-129.592
ER_B=4.848
ER_C=747.4756
```

The Zub File for code-based phantoms
The example below shows the first line of the template file **vox_man.zub**, located in the smc_dir folder. The first line, that should remain the same all the time, has a number '1', which is an identifier for the table to be used. The file can contain several tables separated by a unique number. Please look at the file **vox_man.zub** file for a more understanding of this. The number defined in **change** index 45 should then be the number of the table that will be used in the simulation.

The table contain four columns. The first columns describe the organs (or volume of interest. The second is an integer and the code that the voxel shold have in the image. The third column is an integer and is the density in units of g/dm^3 and the values in the fourth column are a relative number of the activity concentration. In practice, this column defines then number of emitted photons from this voxel location.

```
== V4.3 Code Section 1 vox_man!======!
adrenals                  21  1025   0
bladder                   40  1040   0
blood pool                23  1060   358
bone marrow               26  1030   0
brain                      2  1040   0
cartilage                 30  1100   0
cerebellum                77  1040   0
cerebral aquaduct        122  1040   0
cerebral falx            113  1040   0
colon                     19  1030   0
dens of axis              70  1180   0
```

The zub file is invoked by the switch /FZ: and with the extension “.zub”. The zub file can be used for own cose-based phantoms by using index 14=-5 (byte data image file) or 14=-6 (short float image file). If you create your own zub file, please note that the positions of the columns are important.

The level of the counts in a simulated image is normalized to the activity in the whole source volume, as defined by index 25 in **change**. Thus, the fourth column above indicate the relative concentraction in each voxel of that activity. After a simulation, the absolute activity for each organ/VOI is caculated and shown in the res file.

The Interfile-key file

This text file includes the tags necessary to write the header files in the Interfile format. It is not meaningful for a user to edit this file. Note that there are several additions to the specified Interfile syntax, but these are starting with the characters ‘;#’. Thus, when importing these into a workstation, these tags are ignored because of the ‘;’, i.e. a remark sign in the interfile format.

The collimator data file

This file includes data for the most common collimator files. It is based on a 6-character code that is associated with the dimensions of the collimator. The user can easily modify this file and add new collimators. If a user lacks some common collimators and wants to include them for other users, please send the data to Michael so that he can include them in this file for further releases.

The result-file

simind calculates some basic detector parameters by default. These appear on the screen and in the result file (*.res) together with the input parameters, defined in **change.** Collimator data are only written to the result file if a collimator has been included. SPECT input parameters are written only if the SPECT flag is set. Moreover, the scatter fraction and the percentage scatter, described below, are only written if a phantom has been simulated.

The calculated detector parameters are listed below.	
AcceptanceAngle	This is the calculated solid angle through which the photons are forced. The angle is normalized to 4π, that is, a value of unity equals an isotropic emission.

Compiler	This shows the compiler that has been used to compile the module for which the result file has been obtained. This information may be of value when comparing results from versions of **simind** for different operating systems.
Scatter/Primary	This is the ratio between the events from photons scattered in the phantom to events from photons that have penetrated the phantom without interaction.
Scatter/Total	This is the ratio between the events from photons scattered in the phantom to all events from photons.
ScatterOrder 1,2,3..	This is the percentage of the scatter events originated from photons that have been scattered once in the phantom.
DetectorHits	This is the number of photons that have reached the scintillation crystal.
DetectorHits/CPUsec	Just the number of detector hits per simulation time. Can be useful to compare the performance for different computers.
MaxValue Spectra	This shows the maximum value in the calculated energy spectra. Note that this value may be less than unity because the spectra reflect the accumulated weights of the interacted photons.
CountRate Spectrum	This shows the calculated count rate for the whole energy spectra. This value is based on the activity, given by Index-25 in **change**.
CountRate Window	This shows the calculated count rate for the defined energy window.
Efficiency E-Window	This is the fraction of the number of photons detected within the energy window to the number of photons impinging on the crystal.
Efficiency Spectrum	This is the fraction of the number of photons detected in the whole energy spectrum and the number of photons impinging on the crystal.
Sensitivity Cps/MBq	The count rate in the energy window divided by the activity defined for the simulation (Index-25 in **change**) [cps/MBq].
Sensitivity Cpm/uCi	The count rate in the energy window divided by the activity defined for the simulation (Index-25 in **change**) [cpm/uCi].
Comments	An optional comment line defined in **change** (Index-9 in the main menu).
Message	Sometimes, a routine can give a message to the result file concerning something that may be serious but not fatal for the simulation. The text is then displayed after the "Message" text.

Additional information dependent on selection in change

1. If image-based phantoms with values -2 to -6 are used (based on the *.zub files), then the volumes, activity, number of pixels, and activity concentrations are given in the result file for each organ/structure.
2. If the multiple-source routine is used in the simulation, then information about the activity in each of the spheres is given in the res file.

FILE EXTENSIONS AND FILE FORMATS

The various file types encountered in **simind** are listed in the following sections. On UNIX, Linux, or MacOS systems, all file names are in lower case!

Input files used by simind

*.cr3	Represents table listings of cross-sectional data for various materials. These files are written in ASCII and have an obscure file format. The first four floating values in the first record are the density for the material, the highest atomic number in the compound, and the corresponding K-alpha energy and fluorescence yield for that material. The next value is an integer and is the number of cross-section values in the four columns below. The four additional floating values are reserved for future development of pile-up simulations. The different cross sections for photo absorption, Compton, coherent, and pair production are then tabulated in column form. Finally, atomic form factors and the incoherent scatter functions are tabulated in a special compressed format.
*.dmi	Represents a binary density map file used to generate a non-homogeneous phantom. These files should be in a 16-bit integer format. The density of the phantom should be a factor of 1000, i.e. for water (density = 1.0 g/cm3), a pixel value of 1000 should be used.
*.smc	Represents a listing of the input data created by the program **change.** This file is an ASCII file that contains the information defined in the **change** program. It has a defined format and should not be changed.
*.win	File that includes the window settings used by the scattwin program. Each row is one window with lower and upper energy thresholds in keV. The file format is in ASCII.
*.inp	Represents a file that stores input data used by certain score, source, and isotop routines. The format may depend on the routine but it is suggested that the proper extension be used for clarity.
*.smi	Represents a binary source distribution. These files are similar in format to the files for the density maps (*.dmi) but represent a source distribution instead of an arbitrary density distribution. These files should be in an integer 16-bit format.

Output files created by simind

*.a00	Represents the results of simulated SPECT projections. This binary file consists of projection data in 32-bit float format.
*.bim	Represents a binary matrix image created by the output of **simind**. This file contains 32-bit float binary values.
*.bis	Represents a binary energy spectra image created by the output of **simind**. This file contains 32-bit float binary file.
*.res	Represents the results of the basic detector parameters calculated from the simulation. It also contains input parameters and statistics of the simulation.
.him, *.h00 *.hct	ASCII file that includes header information for the Interfile v3.3 file format. The extension differs depending on the image data. ‘him’ is a planar image with the extension ‘bim’, ‘h00’ is the header file for SPECT projections with the extension ‘a00’, and ‘hct’ is the header file for the aligned density/attenuation file useful for post processing stages, such as attenuation correction.
*.cor	A file that contains the used centre-to-collimator distance for each projection as calculated from the voxel-based phantom. This file will not be created when using analytical phantoms (Index-14 > 0).
*.csv	A comma-separated file useful for importing to spreadsheets such as Excel. Created by Index-84.
*.lmf	Listmode files that include details about each event. The format may **change** dependent on the choice. Please refer to the description of the listmode scoring routine.

TUTORIALS

Here are some examples to get started. These examples assume that you have a display program to show images represented by float values (32-bit float). ImageJ is one example of such a program that handles short float images. If, for some reason, you do not have the tutorials, then you can download the zip file from the link 'Files' to the left.

1. Point source simulation in a cylinder

This tutorial shows how to simulate a simple point source in a cylindrical phantom. The assumption for the camera is as follows. The corresponding index numbers in the **`change`** program are also given

Collimator: LEGP

Energy resolution: 10.0% at 140 keV	(Index-22)
Phantom radius: 11 cm	(Index-6,7)
Phantom length: 20 cm	(Index-5)
Height from phantom to camera: 5 cm	(Index-12)
Energy window: 20%	(Index-20,21)
Pixel size 0.54 cm (64x64)	(Index-27)
Crystal thickness: 0.952 (40x54 cm^2)	(Index-9,8,10)
Photon energy 140 keV	(Index-1)

denotes the prompter sign in the command window

Run 1: Perform a simple run by the command

```
simind point point1
```

The output from the program is now in the files point1.res, point1.bis, and point1.bim. Print out point1.res and identify the different results from the description in the on-line manual. To get a readable version of the spectra, perform the command

```
bis point1
```

which now produces a new file point1.prn. This is a two-column text file of the energy spectra which can be imported to Excel or plotted using, e.g. the freeware gnuplot program. The file point1.bim is a floating-point 64×64 matrix that can be briefly viewed by the command

```
bim point1
```

To convert to an integer image for display using, e.g. the ImageJ program, give the command

```
bim point1/oi/nm:1000
```

A new point1 file with the extension *.int is now created and the maximum pixel value is normalized to 1000. You can now display this image in a display program

Run 2: Change the point source location by changing the z-value (Index-18). This can be done conveniently by using switches, e.g.

```
simind point point1/sc:10/18:4
simind point point2/sc:10/18:2
simind point point3/sc:10/18:-2
simind point point4/sc:10/18:-4
```

Note that the shift is relative origin so the actual depth is Index-7 – Index-18, or for the example above, the depth is 7,9,13,15 cm. Plot the spectra for the different runs and also check from the result files how the sensitivity (cps/MBq) and the scatter-to-total fraction vary with the depth. Try also to turn off the phantom by adding the switch /fa:11 (Flag-11 is set to false). What happens to the spectrum, sensitivity, and scatter-to-total fraction?

Run 3: See how the shape of an energy spectrum changes by performing the following simulation:

```
simind point point5/sc:10/fa:11/fa:12
simind point point6/sc:10/tr:11/fa:12
simind point point7/sc:10/tr:11/fa:12/tr:10
simind point point8/sc:10/tr:11/fa:12/tr:10/tr:7
simind point point9/sc:10/tr:11/tr:12/tr:10/tr:7
```

The simulation will be as follow

- No phantom, no cover, no backscatter, and no energy resolution (cover and backscatter switches not needed because these are set as false in the point.smc inputfile)
- Phantom added
- Phantom and cover added
- Phantom, cover, and backscatter added
- Phantom, cover, and backscatter added with energy resolution

Perform the simulations for a couple of photon energies up to 700 keV and determine the energy range in which the cover and the backscatter compartments become important. The energy switch is /01: so add, e.g. /01:300 to the commands above for a simulation of 300 keV photons.

2. Jaszczak SPECT simulation if six spheres in a cylinder

The aim of this tutorial is to show how a SPECT simulation can be made using a source routine jaszak.fso that simulates spheres of different sizes and locations inside a phantom. The routine is invoked by setting Index-15 = 7. The source routine reads data from jaszak.inp, in which each row defines a sphere. Run the command

```
simind jaszak jaszak1
```

The output files are, in addition to the res file, a file called jaszak1.a00, which contains 64 float 64×64 projections. Moreover, the file jaszak1.h00 is stored, which is an Interfile header file. This file may be of interest to use. If integer maps are needed for display purposes, one can use the following command to obtain such a file.

```
bim jaszak1.a00/oi/nm:100/loop:64
```

The shape, location, and activity of the spheres are controlled by the file jaszak.inp. Each row represents one sphere. The first three define the dimension and shape, the next three are the position relative to the origin, and the last value is a relative activity concentration in the sphere. If the input file has a different name, **simind** looks for this name.inp file and if this is not found, it looks for the default jaszak.inp file. Try and add or **change** the sphere definitions so that you become familiar with how they need to be set up. Then, try the following simulation.

```
simind jaszak jaszak2/co
```

to make a cold-spot simulation where the activity in the spheres is disregarded and assumed to be non-active while the remainder of the phantom is assumed to be uniformly filled with activity. The third case is

```
simind jaszak jaszak3/bg:0.5
```

where the activity in the spheres defined by the values in the inp file are used as well as background activity equal to the value defined by the /bg switch.

Exercises to investigate the SPECT reconstruction:

1. Simulate different numbers of projection angles (Index-29) and different rotation modes (Index-30).
2. Try also changing the start angle for a certain simulation (Index-41).
3. Change the matrix size for the projection angles (Index-76,77). You need to remember to **change** the pixel size to half the value (Index-28).

The scatter fraction in the result file and all other detector parameters are obtained by the sum of all data for all projections simulated. Index-15 in **`change`** needs to be 7 in this version in order to invoke the jaszak.fsc routine.

3. Scattered radiation

Study the contribution from photons, scattered in the phantom, and that have generated events in a fixed energy window (15% is defined in the input files) as a function of different photon energies (Index-1 in **change**). In the res file, you can find values for 'scatter-to-total', ‘scatter-to-primary’ together with the relative contribution of photons with different scatter orders. At which energy is the scatter-contribution greatest? Think about why this is the case?

```
simind legp out1/sc:8/15:4/tr:11/01:75
```

Study the scatter contribution within the energy window as a function of the position of the point source. Note that the minor axis of the elliptical phantom is 11 cm. The simind command for this will be

```
simind legp out1 /sc:8/15:5/18:5/tr:11/84:1/01:140
```

Index 18, which define the shift in the z-direction relative to origin, should therefore be varied within the range of [±10.5 cm] and where a negative value makes the point-source move towards the camera. The switch /TR:11 assure that photon interactions in the phantom are simulated. /SC:8 defines the maximum order of photon scatterings that is allowed. The switch /01: defines the photon energy 140 keV and /15:5 define a point source of radius [0,0,0]. Study the variation in the scatter-tot-total fraction, as can be obtained from the result file, and plot these values as a function of the source depth.

Plot the energy spectrum for some different depths from the binary file out1.bis. Because we run the scattwin routine, energy spectrum is stored in a different way as compared to the default simulation mode. Spectra is here stored in such a way that the first 512 floating point values represent the total energy spectrum, that is all events. The next 512 events represent the total events from scattered photons, and the consecutive next 512 values are all events from photons scattered 1 order, 2 order, 3 orders, etc. You can use the bis program to extract the data into text files *.prn that is suitable to plot, for example, with gnuplot.

```
bis out1 total/sp:1/nc:512 # total spectrum
bis out1 total/sp:2/nc:512 # total primary spectrum
bis out1 total/sp:3/nc:512 # total scatter spectrum
bis out1 total/sp:4/nc:512 # spectrum of 1st order scatter events
bis out1 total/sp:5/nc:512 # spectrum of 2nd order scatter events
bis out1 total/sp:6/nc:512 # spectrum of 3rd order scatter events
..
..
```

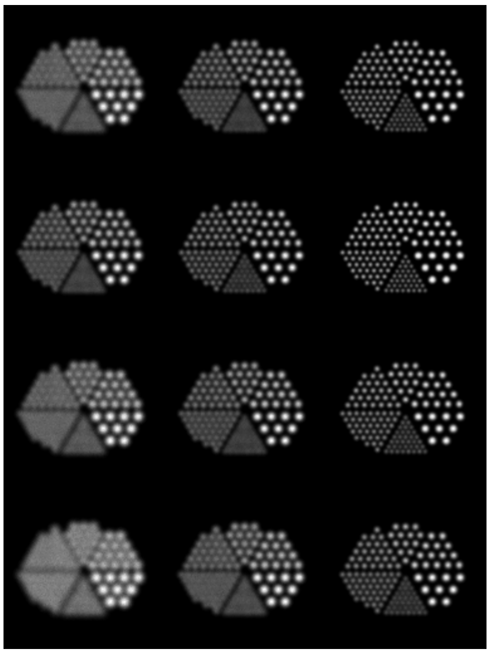
Study the energy spectrums appearance for different photon energies (75, 140, 364 and 511 keV for a point in the middle of the phantom with the command

```
simind legp out/sc:8/15:5/tr:11/84:1/01:75
```

Plot both primary spectra and spread in the same chart. Why do the primary events occur in Compton distribution at high energies? Are not primary photons just that - primary?

4. Simulation of spatial resolution

Simulate the spatial resolution for a bar phantom and 140 keV photons as a function of distance and collimators. Display the bim files using some display program and evaluate the pie slice, which can be resolved for different distances and collimators. Run the simulations with the following command.



```
    simind lehr test1/px:0.15/th:0.1/nn:1000/dir:2/12:30
```

where /12: indicates the distance from the collimator to the source. /PX is a switch that defines the pixel size of the image of the bar phantom (rod_ellh.smi), which is used for the simulation. Note that this pixel size it not the same as the pixel size of the image that **simind** calculates. /TH is the thickness of the rods, /DIR denotes the direction of the source, and /NN is a multiplier (refer to the page on "Image-Based Source" for additional details about how to simulate 3D source distribution using images. Repeat the simulation for LEHR, MEGP, and HEGP. You should get something like this. The bim images are stored as float 128×128 in the file test1.bim

5. Bone-scan simulation using an MDP-like radiopharmaceutical (SPECT)

This tutorial aims to be familiar with the setup of the voxel-based Zubal phantom and how the input file to **simind** for that phantom is defined. The Zubal phantom consists of a set of 8-bit-coded images where each voxel has a unique value that can be related to the organ or structure that the voxel belongs to. These coded images can then be used to define either density maps or activity maps provided that the user has appropriate density and activity values.

simind is written such that it is easy to use this phantom. The program reads in the 8-bit-coded images and from a special user-written table, creates activity maps. A template of such a file is located in the `simind/smc_dir` directory and is called vox_man.zub. In this tutorial, a special file called mdp_ect.zub is used. The file is of ASCII type and consists of four columns

1. The name of the organ,
2. The unique code of that organ,
3. The density value, expressed as g/cm^3*1000, and
4. A value representing the activity concentration (MBq/cc).

The last value is a relative value but is regarded as 'activity per voxel'. It is important that the file is filled with data so that the positions of the values remain the same.

simind assumes a file with the same name as the input file but with the extension *.zub. If this file cannot be found, then a standard file 'smap.zub' is used. If there is no such file, the program is terminated. Index-14 and/or 15 should be set to -2 for the torso phantom simulation and -3 for the whole-body phantom. The size of the phantom in the y- and z-directions is determined by the phantom pixel size (Index-31) and the size in the x-direction is determined by the source (and phantom) length, i.e., Index-2 and -5.

The following example performs a simulation of an MDP-99m-Tc SPECT with and without interactions in the phantom.

```
    simind mdp_ect mdp_ect1/fz: mdp_ect/tr:11
    simind mdp_ect mdp_ect2/fz: mdp_ect/fa:11
```

Here, the file mdp_ect.zub is used as input by the switch /FZ to define the activity distribution.

Tasks: (here, the SPECT option can be turned off by the switch /FA:5)

1. Change the phantom pixel size (Index-31) to different values and investigate the output in the image file.
2. Change Index-2 and -5 and view the output of the images file. Note that if you have selected the phantom to be true, then the sizes in Index-2 and -5 should be equal. If the activity distribution is outside the phantom (Index-2 > Index-5), then an error occurs.
3. Change the image pixel size (Index-28) to different values and investigate the output in the file mdp_ect1.bim
4. What is the difference between 1 and 2? Hint: investigate the scatter to total ratio values.

The step size (Index-34) determines how the path length of the photon in the voxel phantom should be incremented. The size relates to the voxel size in that a value of the step size that is too large can result in missing some voxels. A value that is too small may result in incremental changes to get the photon leaving the voxel. Try to change the step size and see how this changes (a) the speed of the calculation, (b) the accuracy of the calculated results (e.g. scatter-to-total ratio, and (c) the quality of the image.

6. Bone-scan simulation using an MDP-like radiopharmaceutical (Whole body)

This tutorial shows the possibility of simulating whole-body studies with the extended Zubal phantom (called vox_man3.dat), which has been developed by Dr Katarina Sjögreen, Lund University. The major improvement here is that the arms are in a realistic position for WB investigations.

In **simind**, there is a special switch called /WB that makes simulation of WB studies easier. It makes the simulation a SPECT study with two projections, and the posterior projection is mirrored. The matrix size is by default 192×512, but if a value /WB:2 is given, then the pixel size will be half and the matrix size will be 384×1024. The interfile will also have a slightly different layout and the ability to read this has been tested on an ADAC Pegasys workstation

Simulation

```
simind mdp_wb mdp_wb1/fz:mdp_wb/wb/fa:11 # without phantom interactions
simind mdp_wb mdp_wb2/fz:mdp_wb/wb/tr:11 # with phantom interactions
simind mdp_wb mdp_wb3/fz:mdp_wb/wb/tr:11/if # add lesions
```

Note that for the third simulation, **simind** is looking for a file named mdp_wb.inp. If found, it reads from the file and replaces those pixel values corresponding to the locations and sizes given in the input file with new values. The format is similar to the input file format for the "Multiple Sphere Routine" with the difference that the size and location are given as pixel numbers instead of spatial coordinates (x,y,z) and that the activity concentration needs to be an integer.

Tasks: **change** the WB value from 1 to 2 and verify that you can read the new image.

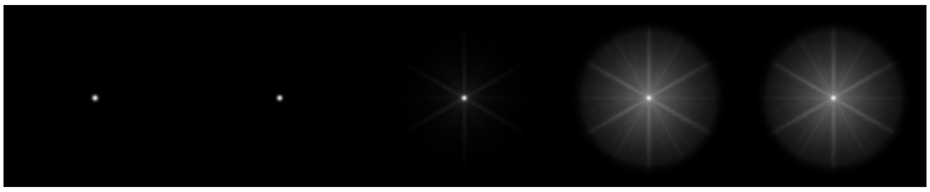
7. Simulation of photon penetration in an LEGP collimator

Study the effect of septum penetration as a function of photon energy for the energies 75, 140, 245, 364, and 511 keV for a low-energy general-purpose collimator. In the result file, the fractions of "geometric collimation", "penetration", and "scatter in the collimator" are given.

1) Use the following command

```
    simind legp result1/lf/19:-45/15:4/01:245/53:1
```

where switch /01 is the photon energy in keV and / 53:1 represents a collimator that includes the interaction used. The switch /19:-45 implies that the acceptance angle for photons emitted towards the camera is 45 degrees and not determined by the collimator hole dimensions. /LF means linear sampling of the polar angle and makes the simulation somewhat shorter. /15:4 mean a cylindrical horizontal source. You should get something like this for an LEGP if you look at the bim files.



The bim images are stored as float 128×128 in the file test1.bim.

2) Plot the values 'geometrical', 'penetration', and 'scatter' from the result file as a function of photon energy.

3) Repeat the simulations for LEHR and HEGP collimators and study the differences.

```
    simind lehr result2/lf/19:-45/15:4/01:245/53:1
    simind hegp result3/lf/19:-45/15:4/01:245/53:1
```

8. Some examples of castor reconstructions

Perform a simulation according to tutorial 5 to get a SPECT projection file mdp_ect.h00 and mdp_ect.a00. These files are then the Interfile header (*.h00) and the data file (*.a00). Give the command

```
    smc2castor mdp_ect.h00 ect/run
```

which will, if all things are set up properly, reconstruct the data to an Interfile ect_it2.hdr and the data file. ect_it2.img. The data format for the images will be 32-bit floating as will the data format from **simind.** The number of iterations and subsets can be changed by the switches /IT and /SU.

If you want to incorporate attenuation correction of simind generated data, you need to set Flag-15 in **change** to true. By doing this, simind will produce a map of density values in a format that is registered to the pixel size and matrix size of the SPECT projections. These maps *.hct and *.ict can therefore be used as an attenuation map in the castor reconstruction. However, castor assumes that the data format is in units of attenuation coefficients (cm-1). To get this from **simind**, the default value of density in the simind.ini file needs to be changed. This can be done by adding the switch /in:x22,5x to the command for the simulation of the projection data. The command for attenuation is

```
    smc2castor mdp_ect.h00 ect/run/atten
```

Note that the information about the name of the attenuation is embedded in the mdp_ect.h00 header information. Thus, no information about the attenuation file is needed in the command line.

If scatter correction also is to be included in the reconstruction, then a file with scatter projections is needed, which has the same format and is properly scaled. Castor uses this file in the forward projector as an additive term. The command for scatter and attenuation correction is

```
    smc2castor mdp_ect.h00 scatter_file.h00 ect/run/atten
```

In order to make it easier for a simind user to obtain this file with the requirements, the scattwin scoring routine has been modified to allow for a direct scaling matching the DW (dual-energy) and TEW (triple-energy) window scatter correction methods. Please refer to the description of the parameters 'dw' and 'tew' in the scattwin routine section.

Smc2Castor program produces some files that may be useful.

1. The projection files needed for the castor reconstruction have the extension *.cdh (header file) and *.cdf (data file). These are only needed at the reconstruction time and will be safely removed.
2. In the CASTOR_CONFIG/scanner folder, a file that is needed for the reconstruction is created, which starts with 'smc_'. These text files are generated by the smc2castor program based on the contents in the *.h00 simind projection file. These files can also be removed occasionally.
3. The result of the castor reconstruction is in Interfile format and has the extensions *.hdr and *.dat. Note that castor adds information about the number of iterations in the file name.
4. The smc2castor program creates a script file starting with the characters 'castor_' for the reconstruction (*.bat for Windows and *.sh for Linux/MacOS). These files are executed when the /run switch is given. However, these can also be started by a keyboard command. These are safe to delete occasionally.

FAQ

Installation

Why do I get the error message "`simind` - Unknown compiler"?

The environmental variable SMC_OS is not set to a proper value. Please read the installation guide for further help on this issue. This is not an issue from version 4.9.

Phantoms

How do I simulate a map from a patient obtained by CT?

The simulation is based on images, which means that you need to define Index-14 and 15 to -1. You also need to specify the name of the files in the main menu by the numbers 12,13. **`simind`** does not know about the format except that it has to be an 16-bit integer file for both the phantom and the source images, so the matrix size is defined by Index-78,79 and 81,82.

The phantom file needs to be scaled to density values * 1000 (voxels with water will have values equal to 1000). The density file (Option 13 in main menu) needs to have the extension *.dmi and the source file (option 14) needs the extension *.smi. The number of phantom images is defined by Index-34 and the start image in the file by Index-33 (usually equal to 1 but you do not need to start from the beginning of the data set).

To scale the image to real physical dimensions, a pixel size for the phantom image needs to be defined by Index-31. This scale the phantom in the y- and z-directions. The length (x-direction) of the phantom in cm (slice thickness) is defined by the half-length of the phantom (Index-5) divided by the number of images (Index-34). The corresponding pixel size for the activity maps is in the current version of the program only given by the external switch /PX:

Could you explain how `simind` decides the #particles/prom? I tried to `change` the parameter #26, which is the number of photon history, but it seems to have no effect; the output file always states a number of 767,555.

When you are using images as a source, Index-26 is not used to maintain a proper sampling of the desired number of decays described by the images. Instead, the value of Index-26 is replaced by the sum of pixels in your images (which is equal to 767555). You can increase this number by multiplying with a factor given by the switch /NN. For example, /NN = 10 gives a simulation by 7 675 550 histories.

Could you explain the geometry in more detail?

If you picture a real gamma camera patient measurement, the patient axis from the feet to the head is aligned to the x-axis and the direction from the middle of the patient to the feet corresponds to negative x-values (**simind** phantoms are always centred in the origin of the coordinate system). Please review the image in section 'Change'. If you view the patient from the feet, a positive **simind** y-value goes from the midline to the right and a negative y-value to the left. A positive z-value goes towards the camera (assuming a camera angle of 0 that is the anterior position) and a negative z-value goes away from the camera. When using standard phantoms (Index-14,15 is 1,2,3,4,5..), the source defined by Index-2,3,4 is centred around the origin. You can move the source around by Index-16, -17, and -18, and this is useful when simulating, e.g. a point source at different depths by changing Index-18.

When using voxel-based phantoms from images, these are only relative number distributions, e.g. 64×64×64 or 128×128×128. Therefore, you have to first scale to a physical dimension. The scaling is done by two values, namely the pixel size (y- and z-directions) if transaxial images are assumed, and the slice thickness (x-direction). For the phantom, Index-31 defines the pixel size for the phantom images assuming square pixels and Index-5 defines the total axial length of the phantom (= number of CT-slices (Index-34) times the slice thickness). Thus, **simind** calculates internally the slice thickness. Index-6,7 is not in use for image-based phantoms if orientation Index-32 = 0. For known phantoms, such as the Zubal phantoms (Index-14 = -2,-3,-4) and the NCAT phantoms, the source maps correspond to the phantom maps; the pixel size for the source maps will be the same, as will the slice thickness. In these cases, to be safe, you can also set Index-2 equal to the value of Index-5.

Output

What about pixel values in bim file. Are they fractions of one (unity) or what is this value? How can one obtain counts in each pixel?

The pixel values are normalized in such a way that is the probability for counts in that pixel per defined activity (Index-25) in the source and for an acquisition time of 1 s. The activity can be defined in a cylinder (Index-14 = 4) or spread out according to a distribution provided by a 3D image set. To get realistic counts, one needs to scale the image after a simulation by the appropriate activity (if not done already by Index-25) and the appropriate time per projection. If this is done for a long simulation, one can finally add Poisson noise to the data to get a realistic image. Not that the magnitude of the pixels remains the same even if additional photons are simulated by increasing Index-26 or using the nn: switch. The statistics will of course be better if more photons are simulated.

Runtime

Is there a possibility to continue a simulation after a power failure?

The only option one has to continue a simulation that has been terminated is to use the switch /pr:n to start a simulation at the particular projection number 'n'. However, the data in the RES and the energy spectrum will be based on this point and forward, so if you want to simulate an energy spectrum that corresponds to the average of all projections, you will use this switch for correction. You must give the command exactly as the initial command and only add the /PR switch

Simulations

How do I create and simulate a new isotope?

1. Create a file in the smc_dir folder, which, e.g. will be called p32.isd with your spectra. The format is a simple text with energy in keV and the abundance. Look at, e.g. i131.isd in the smc_dir to understand the format. It is important that you follow this format, i.e. the number appears in the right place and columns.

2. If you specify a negative energy Index-01 in **`change`**, then this indicates the middle energy of the energy window as before but the negative sign tells **`simind`** to use an energy file instead of the actual number in Index-01.
3. At the command line, when running **`simind`**, add the switch /fi:p32 to tell which isotope file you shall use. Alternatively, copy p32.isd to the default name spectra.isd in the current folder where you perform your simulations.

INSTALLATION PROCEDURE

Linux/MacOS

1 Create a home directory for simind. In the following, it is set to `$HOME/simind`

1. Download the distribution you want to install into the `$HOME` directory and unpack it using the following commands :

   ```
   gunzip smc_intel_64_v70.tar.gz
   tar -xvf smc_intel_64_v70.tar
   ```

   where xx is the version number.

2. Add the following lines to the .cshrc file. This is only needed the first time.

   ```
   setenv SMC_DIR $HOME/simind/smc_dir  (tcsh,csh)
   set SMC_DIR=$HOME/simind/smc_dir     (bash,sh)
   ```

3. Add `$HOME/simind` to the `$PATH` search.
4. Execute the command `source .cshrc` to make the changes effective.

In addition to the **`simind`** code, you might need Michael's help to run the following programs.

```
bis
bim
```

Both `bim` and `bis` are described in the manual but discrepancies can occur. Included is a data file 'bench.smc'. Read in this file by the command

```
change bench
```

Simulation

```
simind bench
```

Windows

1. Create a directory for simind. Here, the example is `c:\simind`
2. Unpack smc_win_intel_64_v70.zip in c:`\simind`. Keep folder information while unzipping and be sure that a directory `c:\simind\smc_dir` with some files of extension *.cr3 have been created.
3. In the System/Advanced/Environmental Variables/User Variables/ window add (by clicking "new") the following environmental system variables by filling in proper fields:

   Variable `SMC_DIR` set to value `C:\simind\smc_dir\`

4. Add the path `C:\simind` to the environmental variable `PATH` in the "System Variables".
5. To use the new text-based change program, add the following key to the Windows registry by open a cmd terminal and give the following command

   REG ADD HKEY_CURRENT_USER\Console /v VirtualTerminalLevel /t REG_DWORD /d 1

You only need to make this once.

6. Restart the command line window.

The "Castor" programs

On the simind webpage, there is a link to a zip file containing a distribution for the castor software version 1.0 with pre-build executables for Linux, MacOS and Windows (32 and 64-bit). This zip-file can be seen as a starting point to use the castor reconstruction. It is, however, strongly recommended that users of this program register themseft to the community list at the www.castor-project.org in order to receive information about future updates and to get other important information. On the page, there is also information about to configure and compile the builds to particular preferences since it is not the intention that this website should be regarded as a mirror-site.

To get started and use the smc2castor program for reconstruction of simind-genereated SPECT projections then following installation/preparation is needed.

1. Download the zip file castor.zip from the webpage.
2. Extract the whole file structure to a convenient place on your harddrive.
3. Define the environmental variable CASTOR_CONFIG to the absolute path of the config folder within the castor directory structure.
4. Add the relevant bin folder to you PATH environmental variable (that is, either bin_osx, bin_linux, bin_win32 or bin_win64).

VERSION HISTORY

V 7.1

- The definition of a new projection in the list mode routine has been changes. Please refer to section for more information.
- The definition of the zub file is now only made by the switch /FZ: in order to avoid confusion of what file that is used in the simulation.
- The definition of the input file in the multiple source routine and the images-based source routine is now only made by the switch /IF: in order to avoid confusion of what file that is used in the simulation.

V 7.0

- A new file format for the *.smc files have been introduced. The old smc files can still be used but the **change** program is only writing to the new format. Hopefully this format will be easier to expand the files in the future.
- As a concequence of above, the main menu of the **`change`** program has been rearranged. Please note that the order of the numbering also has been changed. The possibility to clear all data has been removed.
- The transmission page has been removed from change and SIMIND and are not supported anymore since these types of simulations are most likely not intereing any more due to the introduction of SPECT/CT. The code for this also to some un-necessary space
- It is now easier to use fitted functions to model camera specific energy resolution. In the previous version, it was only possible to define the parameters within the simind.ini file but now the parameters can also be stored in an external text file, and the name of this can be pre-defined in the main menu of the **`change`** program.
- The way dynamic planar and SPECT are simulated has been completely re-written. Please read section “Dynamic Imaging” for information about this and the change in the format of the TAC files.
- Some minor bug fixes and cosmetic changes to the output in the res file.
- From Windows 10, there are now support for ANSI escape sequences when using the default **`cmd`** command-line console program. This allow the use of the new text-based **`change`** program to be used. However, this requires an initial change inte Windows Registry system. In the distribution zip file, the **`change`**.exe is now only the text-based version. See the installation information on specific details about the registry-key needed. From this version and on, the GUI-based change will not be supported but it will remain in the version 6 zip files.
- The directions of the image-based source maps and multiple-sphere routine have been changed to range from 0 to 2 to be consistent with the directions for the phantom (index 32).

V 6.2

- An aligned roi file can now be saved. This ROIs are created from the image-based phantom and resampled to the image resolution for the simulated projection. The purpose is to have available the geometric ROIs to be used, for example, to evaluate partial volume effects. Since this is not predicted to be used often the creation is initated from a parameter in the simind.ini file. However, since each value in this file can be changed by the switch /in the invocation of this option can be made by the switch /in:x54,1x. A 2-byte integer file with the source name base name and the extension *.iro will be created. If the switch is defined as /in:x54,999x then the simulation is terminated after the *.iro creation. If code-based images are used, then it is the code that is stored. If it is a *.smi file (index15=-1) then it will be the values in that file that are used. This means that if a user för this particular file type wants to separate sources, they need to have different values (that is for the kidneys specify a source value unique to each kidney. This may of course be a limitation but is the only way to get segmentation to work on this type of source maps.
- Several small bugs have been fixed and slight changes in the res file layout have been made.

V 6.1

- Bug fixes and a new print-out format to the screen (6.1.1)
- A value of the keV/chanel is not more written to the last cell in the energy spectrum (6.1.1).
- The **smc2castor** program has been written to make it easy to reconstruct **`simind`** projection data using the castor reconstruction software http://castor-project.org/.
- The scattwin routine has been updated to make is easier to simulate data for the dual-energy window scatter correction methods or the triple-energy window method. This is mainly done to get data that can be directly used in the **castor** reconstruction program.
- A new switch is defined /IN that allows the user to change the default value in the simind.ini file. These default values are now numbered and thus possible to change at runtime.
- The spectrum BIS format for the scattwin routine has been changed. The 1st spectrum in the bis file is now the total spectrum, the 2nd is the spectrum for all primary photons unscattered in the phantom, the 3rd is the spectrum for all photons scattered in the phantom and the 4rd, 5th, 6th, etc. spectra are for scatter order 1,2,3,…max scatter order.
- If a bremsstrahlung spectrum is used, the values are not printed anymore in the res file owing to the length of the spectrum.
- An issue with index 59 för parallel hole collimators (collimator movement) has been fixed.

V 6.0

- CZT detectors are now possible to simulate, and there is a new menu for this in the **`change`** program. The methods have been validated and described in reference (5).
- Dynamic planar imaging is possible to simulate using user-specific TAC files.
- Pinhole imaging has been improved and validated. Results from this validation are published in reference (17).
- The simulation of the scatter has been corrected because the first-order coherent scatter was not simulated correctly. The impact of this correction is mainly seen in the lower energy region of the pulse-height distribution where the coherent scatter becomes important.
- The **`change`** program now displays the context of the collimator-specific parameters in the menu. The menu is **`change`** when a different collimator type is **`change.`** Thus, the menu for pinhole collimators is different from that for the parallel-hole ones.
- **`simind`** is now supporting MPI parallel computing through the use of software from https://www.open-mpi.org/. A binary version of **`simind`** called **`simind_mpi`** is included. Currently, only Linux and MacOS are supported. Note that this required that you install openmpi accordning to instructions. No static builds for simind are yet available.

V 5.0c

- In the listmode, the indicator for a new projection is now defined as a whole listmode event record where all parameters except for the scatter order are set to the value -99. The previous version to indicate a new projection angle used only a negative value for the deposit energy, but in some cases, it was not possible to set the indicator for the new projection because no energy was deposited.

V 5.0

- The code is now built with 64-bit float variables as default in order to increase the precision. However, the output of the images *.bim and *.a00 and the energy spectrum *.bis remain stored as 32-bit float data.
- **`Simind`** now uses the RANLUX random number generator. Here, it is possible to configure the accuracy of the random numbers by a value in the **`simind.ini`** file ranging from level 0-4, where 4 is the most accurate but also the slowest. The default value is set to 2. Additional information on RANLUX can be found on Google.
- A new set of isotope files is available. The data has been taken from the following web site http://www.nucleide.org/DDEP_WG/DDEPdata.htm. Owing to the limitation in the cross-section ranges, the simulated photon energies are restricted to be within the range of 20 keV to 2500 keV.

- The code has been clean regarding the definition of the COR. This means that for no **`change`**, the values in the **`simind.ini`** file and Index-36,37 and 39 should be zero.
- I have encountered some problems in using the 'RR' switch when running on clusters. Identical projections appeared even if different values for the RR switch were given and I have not found the reason for this. However, I have developed a work-around for this problem in that the random number generator is called RR times before starting each history (if the switch is given). This means that there is no advantage of giving high values for the RR switch as long as they are not the same.
- The activity values and concentrations for the voxel-based phantom, as defined from the *.zub file, can now be written to a CSV file.
- There is now a possibility to simulate the shape of the tumours based on the Butterworth filter equation. The cut-off and order are given in the tumour input file.
- Now, four collimator-dependent variables are defined, which are specific to a particular collimator.
- The source distribution generated by the jaszak routine can now be changed in their different directions using the switch /DIR:
- A cone source shape has been added to the "Multiple Sphere" simulation. If the cone number is negative then the cone direction is reversed.
- A new scoring routine "penetrate" is included, which separates the image into components.
- The "Save Aligned Density Map" (Flag15) now also supports geometrical phantoms (Index-14 = 2-4).

V 4.9

- Four new shapes for the "Multiple Sphere" simulation. A horizontal hexagonal rod shape, a rectangular rod, and horizontal and vertical cylinder shapes are defined by an additional value in the input file. If these values are omitted, then the default will be spheres.
- Index-30 now is the rotation angle with the sign defining the direction. However, the old values 0, 1, 2, and 3 that define the rotation mode remain.
- A new scoring routine has been added. It produces a binary list mode file. Additional information and the format for this is found under "Scoring Routines".
- If there is a file named **`simind.loc`** in smc_dir, then this will be used instead for the **`simind.ini`** file. This is to avoid local settings being overwritten each time a new version of **`simind.ini`** installed. However, note that an error will occur if the content of the **`simind.ini`** file have increase due to an update.
- An option for an output to a CSV file has been added by the **`change`** Index-85. This makes it possible to easily import results to, e.g. Excel.
- The default values for the centre have been changed to better reflect the common definitions in reconstructions. This means that for a 64×64 matrix, the midpoint point x = 0, y = 0 will be in between pixel 32 and 33 but (0,0) will be scored in pixel 33. The mid-point for a DMI and SMI file with 64 images will similarly be between 32 and 33. These default values can be changed in the smc_dir\**`simind.ini`** file.
- Some bugs and error-handling problems have been fixed.

V 4.8

- The "Multiple Sphere" routine can now have any number of spheres, and combinations of spheres and cylinders can be made in a single simulation.
- Several important bug fixed.

V 4.7

- A MacOS version is now available
- The **`change`** program has been modified to include help for each index. When adding a ´?´ character after or before an index number, you get help on this topic. The help text is the same as in this on-line manual.
- The SMC_OS environmental variable is now obsolete and not used anymore.
- Bug fixes when calling the Multiple Source routine.

- Old Index-39 and -40 have been removed because the need for calculating a pure sinogram for a specific slice has been regarded as obsolete owing to the availability of large disk space in modern C. Index-39 has been replaced by a possibility to shift the phantom relative to the x-direction.
- The file name for the aligned transmission map (Flag-15) is now defined by the base name of the density map. Previous versions set the name from the base name of the input file.
- The default cross-section table for the collimator is now pb_sb.cr3 (set in smc_dir\**`simind.ini`**). This is because discussions with several collimator vendors have indicated that antimony is generally added to the lead in order to make the collimator harder and more resistant to damage. The amount of antimony may vary among vendors but the percentage amount seems to be of the order of 2%. The Pb_sb cross-sectional table therefore has 2% antimony. A clean lead cross-section table is pb.cr3.

V 4.6

- The Switch /PR has now been modified such that a positive number indicates the start projection number (as before) and a negative number indicates the stop projection number. The order does not matter!
- If you have changed the **`simind.ini`** file and want to store this on a special folder to avoid problems with installing a new version, you can assign the full path and file name to the environmental variable SMC_INI.
- The program is also compiled with the Intel Fortran Compiler for Windows version 10. The executables from this compiler are faster than Lahey but I will keep compiling versions for Lahey.
- The **scattwin** routine is now included in the main program by setting Index-84 = 1. The dummy scoring routine used earlier is invoked when Index-84 = 0. The possibility for linking user-written scoring routines remains the same.
- The two different collimators (the analytical collimator developed by Frey et al. and the collimator simulating explicit septal penetration) have now also been included in the main program. Thus, there is now no need for the **`simindc`** and **scattwinc** programs. The default collimator is the earlier version based on analytical weighting and is used by Index-53 = 0. The collimator simulating penetration and scatter is used when setting Index-53 = 1. In a later version of **`simind`**, a pinhole collimator will also be included.
- To reduce the number of programs, the former **jaszak** routine simulating multiple elliptical sources is now included in the main program and is selected by Index-15 = 6. Additional information about how to define the sources can be found on the page regarding "Multiple Spheres".
- A simple myocardial shaped distribution can be defined by selecting Index-15 = 7. Information about this distribution and how to set it up can be found on the page regarding "Myocardial source".
- Index-26 in **`change`** is now in units of 1 million histories. This is because the figures in **`change`** will otherwise become quite large. It is rare that one wants to run only 1000 histories and this can be represented by the value 0.001 in Index-26.
- Feature: A new switch /RR for initiating the random number generator when running on clusters has been made.
- Minor bug fixes and changes in the contents of the *.res file.

V 4.5

- Feature: Additional information about the registered events penetration events as function of acceptance angle (Index-19) is provided in the *.res file when running the **`simind`**c and scattwinc programs. This information can make it easier to verify that a proper value of Index-19 is defined.
- Feature: Simulations of scintillation cameras with COR as a function of the projection angle can be simulated. For further information, refer to Index-42 in the **`change`** program description. I think this feature needs some more validations with different phantoms, so please let me know about success or failure when you are using this.
- Feature: If Index-19 is set to 3, then the acceptance angle is calculated based on the detector size and phantom size and not only as a function of the collimator hole dimensions (as is the

case when Index-19 is set to 2). This ensures that the acceptance angle is not too small. The used of this value may however not lead to an optimal value regarding the calculation time. This is only important when running the **simind**c and scattwinc programs.

- Bug fix: There was some problem with the tutorial files in earlier versions. In order to avoid conflicts between versions, the tutorial files and phantoms are now included in the main distribution file.
- Feature: A density value for each tumour in the *.inp can now be defined. Further information will be found in the "source routine" page.
- Feature: For this version, a Windows Installer script is used to install the program on Windows computers. The script asks for a location under which the folders **simind** and smc_dir are installed. Most likely c:\ is OK. The script adds the environmental variables and adds the **simind** directory to the path. At present, you might need to uninstall a previous version of a "Windows Installer" installed program. This is still under development, so for those who are happy with the zip/tar files, these files will remain available.
- Feature: A possibility to use a cut-off energy to terminate photon histories with extremely low energy compared to the energy window has been added and when phantom interactions are simulated. The cut-off energy value is defined by Index-83 in the **change** program. Please refer to the main page for the **change** program for further information about this feature.
- Feature: A description about the activity and the volumes of each of the organs has been added to the results file when simulating a Zubal-like phantom (value -2 to -6 for Index-14 and -15 in the **change** program). This information is useful when setting up a simulation when the total activity in the organ rather than activity concentration is known.
- Feature: Sometimes one would like to have a fixed date and time in the interfile header instead of the actual computation time. Using the Switch /TS (Time shift) supported with a value, one can define a time shift in hours relative to a fixed initial time (that can be regarded as the injection time). The value of the switch is in units of hours. The fixed time and date are defined in the **simind.ini file**.
- Some more new switches have been added. Please refer to the page on "Running **simind**" for a complete description.
- The error messages have been revised.
- Some revisions of this web page.

V 4.4

- Feature: A new collimator routine has been developed, which is included in the pre-compiled versions. Those programs ending with 'c' have this new collimator. The method is described in the paper *Ljungberg, M., Larsson, A., and Johansson, L. A New Collimator Simulation in* simind *based on the Delta-Scattering Technique IEEE Trans Nucl Sci 52:5; 1370-1375, 2005.* What is important is that the acceptance angle is sufficient to allow for penetration effects. Index-19 defines this acceptance angle in the **change** program. By giving a negative number, the acceptance angle is then defined as the magnitude defining the degrees of the angle. There is no need to provide a value greater than 90 degrees (that is -90). By default, X-ray generation in the lead is turned off but you can add this by changing the **simind.ini** file in the smc_dir directory or add the switch /XR:1 to the command line.
- Feature: You can now specify a Zubal file as the third file name in the command line, which has have a different name from the input file name and the standard name smap.zub. You can also specify an ini file as the third and fourth names (the order does not matter). You can also specify the Zubal file name and the isotope file name using the switch /FI and /FZ, respectively. For example, **simind** input/01:- 140/FI:in111 indicate that the **in111.isd** file should be used. Note that you may still need to define /01 to a negative number because the actual value of the number defines the centre energy for the energy window.
- Feature: The density and source map file names can now be set by the switches /FS (source) and /FD (density). This makes it easier to set up scripts and avoid potential problems because different phantoms can be simulated using the same SMC input file.

- Feature: You can **`change`** the cross-section tables by the switch /X1-/X6 where the X1-X5 refers to the Index-9,10,11,12, and 15 in the main **`change`** menu and X6 is the material for the collimator when using **`simind`**c, scattwinc. Do not specify the extension.
- Feature: You can specify a different collimator by the switch /CC. Do **NOT** add the character '*' that needs to be present according to how to select a collimator in the **`change`** program. For example, >**`simind`** input output/cc:ge-legp gives the collimator ge-legp as listed in the collim.col file in the smc_dir directory.

V 4.3

- Feature: The tumour definitions have been refined by including two types of tumours based on the information in the *.inp file. Tumour type 1 has the centre in the pixel defined by value 4,5,6 in the *.inp file. If valued 1,2,3 are equal to zero, then the tumour will be a single pixel. Tumour two has the centre at the surface of the pixel that is between i,j,x and i-1,j-1,k-1. The minimum size for this tumour is 8 pixels.
- Bug fixes: The program for Linux did not read integer maps (30 Oct/2005). The inserted voxel tumours with the *.inp file did not appear symmetrical (26 Oct 2005).
- Feature: The definitions of the densities for the different organs in the Zubal type of phantoms have been moved from the **`simind.ini`** file to the *.zub. This makes it easier for the user to modify the densities and it does not require a modification of **`simind.ini.`**

V 4.2

- Feature: The format for the Zubal file *.zub has been changed in that additional phantoms (RSD) have their own section after the voxman and voxbrn sections. Before, some of the code was also used for the RSD phantoms.
- Feature: The parameter "photons/decay" (Index-24) is now automatically updated based on the probabilities in the isotope files *.isd if this are used.

V 4.1

- Feature: The generation of X-ray photons has been improved and made more accurate. Now, sampling is made from Kalpha1, Kalpha2, Kbeta1, Kbeta2, and Kbeta3 as has been defined in the cross-section tables. The cross sections are better described in this manual.
- Feature: Cross-section tables have been modified, so previous *.cr3 files do not work.
- Feature: The **`simind.ini`** file has been modified and now densities can be specified for each of the structures also in the Zubal brain phantom (Index-15 = -3). Before, this was only possible for the Voxelman phantom. Further, some additional flags have been added to the ini file that turn off certain effects.

V 4.0

- Feature: A new sampling technique, called delta-scattering sampling, has been implemented for the simulation of non-homogeneous voxel phantoms. This method roughly improves the speed by a factor of 2-3.
- Feature: The sampling of the photon in the phantom has been changed so that it now uses true photon splitting instead of sampling a particular scatter order. This means that in each interaction point and from the decay point, a virtual photon is sent to the detector. The number of photons actually simulated will therefore be the number of photon histories times the scatter order minus some photons that escape the phantom. It may seem that the simulation time has increased but more photons are simulated and the efficiency therefore is increased.
- Feature: The form factors and scattering functions for bounded electrons. These data are now taken from the EPDL97 documentation. A new cross-section file format has been constructed with the new extension *.cr3. The energy grid for the cross sections is now 0.5 keV instead of 1.0 keV. The data ranges from 1 keV to 1 MeV.
- Several new materials have been added. A user cannot create his own table because the generation is based on an IDL program. However, a request to me for a new compound or materials can be made by e-mail.

V 3.9 (including some of the version 3.8 changes that were not documented)

- The number of detector hits per second has been added to the result file as a rough indicator of the efficiency. Note that it is not the true CPU time but rather the live time for the simulation.
- A bug in the scattwin routine resulted in wrong scatter order values and percentage values in the result file. This has now been fixed. Further, the results by adding data from two separate energy windows were different when compared with the situation where the two windows are added within the simulation. This should of course be the same .
- The presentation of the time for a simulation has been improved.
- The jaszak source routine has been updated so it now allows cylinders as the source shape in addition to spheres. Note that the direction of the cylinders is along the x-axis.
- Updated the manual and especially a) the description of available switches for the **`simind`** program b) the features for the scattwin routine
- The lesion code in the Zubal phantom vox_man.dat (located in the smc_dir) has been removed and replace by the liver code. This is because lesions are handled separately in **`simind.`**
- The order in the main menu was not consistent between the Windows version and **`change`** and the UNIX version.

V 3.7

- A Windows version of **`change`** was written because on the Windows 2000/ME system, the ansi.sys drivers have been removed.
- The location of the smc_dir directory is now handled by environmental variables both in Windows and in UNIX.
- The NCAT file binary format has been included as a phantom number. This is useful because the image direction is opposite to the MCAT phantom.

V 3.6

- The scattwin scoring routine now calculates the 'air' images at the same time as the images with scatter and attenuation. This means that only one simulation is necessary.
- Implementation on the MCAT3 file format and naming convention.

V 3.5

- **`simind`** has been re-written so it now allows for parallel computing using the Message Parsing Instruction (MPI) standard. The calculations have been validated on a IBM SP2 computer machine at the University of Michigan Computing Centre

V 3.2

- Densities can now be individually assigned to each organ in the Zubal phantom by editing the **`simind.ini`**. Moreover, a flag control of the earlier average densities for bone, soft tissue, and lungs shall be used.

V 3.1

- The weight is now multiplied with the Activity*PhotonsPerDecay and divided by the number of simulated photons. The total count in the projections and the spectra are therefore corresponding to a measurement of 1 s for the defined activity for the current radionuclide.
- Minor changes in the output file. One of these is that you get the command line, which can be useful, if you have to overwrite the values in the SMC file using a switch.

V 3.0

- A new format for the cross-section files that replace the interpolation step in the previous version.
- The implementation of direct access to MCAT phantom and source files.
- Implementation of an input file in smap.fso that generates tumours in a template source map file.

- External scattwin.fsc scoring routine that makes it easy to simulate multiple energy windows.
- Added shift in the y-direction for the camera relative to the phantom

V2.x

- The major **`change`** is the movement to Fortran 90, which allows dynamically allocated arrays. This greatly increases the flexibility of the program because the matrix size can now be selected from a menu in **`change`.**
- An INI file has been implemented, which stores some of the default values. You can have local INI files stored in different directories and have **`simind`** read these instead of the default file, stored in the smc_dir directory.
- A /WB switch has been added, which easily makes an A/P simulation with, e.g. the Zubal phantom. The output Interfile takes this into account.
- Minor improvements.

REFERENCES

1. Ljungberg M, Strand SE. A Monte Carlo program for the simulation of scintillation camera characteristics. Comput Methods Programs Biomed. 1989;29(4):257-72.
2. Ljungberg M. The SIMIND Monte Carlo program. In: Ljungberg M, Strand SE, King MA, editors. Monte Carlo Calculation in Nuclear Medicine: Applications in Diagnostic Imaging - Second edition. Florida: Francis & Taylor; 1998. p. 111-28.
3. Ljungberg M. The SIMIND Monte Carlo Code. In: Ljungberg M, Strand SE, King MA, editors. Monte Carlo Calculation in Nuclear Medicine: Applications in Diagnostic Imaging - second edition. Florida: Francis {&} Taylor; 2012. p. 315-21.
4. Ljungberg M, Larsson A, Johansson L. A new collimator simulation in SIMIND based on the delta-scattering technique. Ieee T Nucl Sci. 2005;52(5):1370-5.
5. Pretorius PH, Liu C, Fan P, Peterson M, Ljungberg M. Monte Carlo Simulations of the GE Discovery Alcyone CZT SPECT Systems. Ieee T Nucl Sci. 2015;62(3):832-9.
6. Guerra P, Santos A, Darambara DG. Development of a simplified simulation model for performance characterization of a pixellated CdZnTe multimodality imaging system. Phys Med Biol. 2008;53:1099-113.
7. Ramo S. Currents induced by electron motion. Proc IRE. 1939;27(9):584-5.
8. Shockley W. Currents to conductors induced by a moving point charge. J Appl Phys. 1938;9(10):635–6.
9. Chen L, Wei YX. Monte Carlo simulations of the SNM spectra for CZT and NaI spectrometers. Appl Radiat Isot. 2008;66:1146-50.
10. Gros d'Aillon E, Tabary J, Glière A, Verger L. Charge sharing on monolithic CdZnTe gamma-ray detectors: A simulation study. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. 2006;563(1):124-7.
11. Zubal IG, Harrell CR. Voxel Based Monte Carlo Calculations of Nuclear Medicine Images and Applied Variance Reduction Techniques. Image and Vision Computing. 1993;10:342-8.
12. Zubal IG, Harrell CR. Computerized three-dimensional segmented human anatomy. Med Phys. 1994;21:299-302.

13. Zubal IG, Harrell CR, Esser PD. Monte Carlo determination of emerging energy spectra for diagnostically realistic radiopharmaceutical distribution. NuclInstMethinPhysRes. 1990;A299:544-7.
14. Segars WP, Lalush DS, Tsui BMW. A Realistic Spline-Based Dynamic Heart Phantom. IEEE Trans Nucl Sci. 1999;46:503-6.
15. Segars WP, Lalush DS, Tsui BMW. Modeling respiratory mechanics in the MCAT and spline-based MCAT phantoms. IEEE Trans Nucl Sci. 2001;48:1192-5.
16. Segars WP. Development of a new dynamic NURBS-based cardiac-torso (NCAT) phantom. Universiy of North Carolina; 2001.
17. Peterson M, Strand S-E, Ljungberg M. Using the Alloy Rose's Metal as Pinhole Collimator Material in Preclinical Small Animal Imaging: A Monte Carlo Evaluation. Medical Physics. 2015;42:1698-709.