The Monte Carlo Method: Versatility Unbounded In A Dynamic Computing World
Chattanooga, Tennessee, April 17–21, 2005, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2005)

VOXEL PHANTOM SETUP IN MCNPX

V. Taranenko, M. Zankl, H. Schlattl
Institute of Radiation Protection,
GSF—National Research Centre for Environment and Health
Ingolstädter Landstrße 1, D-85764 Neuherberg, Germany
taranenko@gsf.de

ABSTRACT

The implementation of voxel anthropomorphic model geometry in the MCNPX general purpose multiparticle code is addressed. Latest improvements of MCNPX resolve the problem of slow geometry initialization and scoring, which made it possible to describe a whole model by means of the standard MCNPX syntax. The error-prone code modifications allowing fast tracking in voxels is superfluous now, since the standard approach yields an acceptable run-time. The procedure of voxel data transformation into the code geometry is done via repeated structures representation. The MCNPX standard method of scoring the spatial dose distribution inside an organ is discussed. It is illustrated in two cases of a female and a male phantom (Klara and Godwin), which have been recently adjusted according to the ICRP Reference Man anthropomorphic data in the Medical Physics Group of GSF. The phantoms consist of several million non-void voxels assigned to approximately 90 organs using about 30 media. These well-established phantoms are used for the purposes of radiation protection and intended to become reference voxel phantoms in the field. Exemplary MCNPX calculations of average absorbed dose evaluation were carried out for internal homogeneous sources in the critical organs as well as for external geometries of irradiation with monoenergetic photons 0.01–3 MeV. The scoring efficiency and ways of its improvement are discussed. The MCNPX results of coupled photon-electron transport through the voxel geometry are compared with the outcome of similar calculations with the EGSnrc code.

Key Words: voxel phantom, Monte Carlo, MCNPX, EGSnrc

1 INTRODUCTION

MCNPX (MCNP eXtended) is a multiparticle, all-energy (eV—TeV) Monte Carlo general purpose transport code from Los Alamos National Laboratory in the USA (Waters 2002). It is based primarily on the MCNP code (Briesmeister 2000) as well as on the other Monte Carlo codes LAHET, Fluka, CEM, etc. While keeping the conventional MCNP option of tracking photons and electrons at low energies (keV and MeV range) along with the neutron transport, it adds, as an example, the possibility of proton transport simulation. The latter is of particular interest in the research connected with novel proton therapy.

Since the geometry description in MCNPX is independent of the physics involved for a particular model, it was the primary task of the present work to set up the geometry of a voxel phantom while keeping an acceptable scoring efficiency. Starting from well-studied photon-electron transport simulations, the ultimate aim of the Monte Carlo modelling with MCNPX was to provide the method of transport simulation for particles other than photons and electrons.

Recently, the Monte Carlo simulation of particle tracking in the voxel geometry using MCNPX required either a tricky code modification (Burn *et al.* 2002) or extensive resources of computational power. It is due to the latest improvements of the code that nowadays we are able to set up a voxel geometry using standard features. The scoring and source sampling can be

efficiently employed by using standard MCNPX syntax, namely input file records which activate the desired by the user functionality (see Hendricks 2004 and other documents on recent code extensions available on MCNPX web-site: mcnpx.lanl.gov). The error-prone code modifications that allow fast tracking in voxels are superfluous now, since the standard approach yields an acceptable run-time.

2 VOXEL MODEL SETUP IN MCNPX

2.1 Geometry and materials

The procedure of voxel data transformation into the code geometry is done via repeated structures representation. This eliminates the code limit for the number of cells in the model. The set of input data is a combination of: a three-dimensional array of integers of organ or tissue identification numbers (IDs); the dimensions of the phantom elementary voxel; and, finally, material specifications along with densities for every organ/tissue ID.

We describe the procedure as follows. First, in the second block of the input file, which assigns surfaces, we define the surfaces of a box (macro body) that holds the whole three-dimensional array of voxels (below a set of box facets is labeled as **1**):

```
1 rpp 0 53.1675 0 27.522 0 176       $ big box, surfaces
```

(Here and below, for reference on code commands—cards—see Waters 2002). The numerical parameters in the line above represent overall dimensions of a phantom. Note, that along each axis, the phantom extends in positive direction starting from the origin. The overall phantom positioning is arbitrary, but must be essentially compatible with the input cards on later stages of coding. As an example, here, the x-extent equals **53.1675** cm, the product of the voxel x-size (0.2085 cm) and the number of voxels along the *x*-axis 255.

Second, in the first block of the input file, where cells are specified, we fill the above mentioned box with a space lattice that was especially prepared:

```
100 0 -1 fill=999        $ big box filled with lattice #999, cell
```

Here, cell number **100** is defined inside the box surfaces denoted as **1**. The cell is filled with lattice number **999**, the construction of which is described below. Zero followed by the cell number stands for void material in those places of the cell where no lattice is defined. Since the lattice will have the same size as the box which envelops the phantom, there will be no such undefined regions; nevertheless this input parameter is required.

The definition of the rectangular space lattice of voxels is the central part of the voxel phantom setup. It is based on the highly useful MCNPX feature of repetitive structure definition that specifies a regular geometry filled with varying entities. These entities—in MCNPX terms "universes"—in turn, are defined as normal cells. Thus, in the first block of the input file, we define a lattice **999** (which is also a universe—hence we can fill cell 100 with it) as follows:

```
999 0 -2 lat=1 u=999 fill=0:254 0:131 0:219 & $ 3D lattice 255x132x220, defines universe #999
```

This line defines a cell **999** and rectangular (**lat=1**) lattice **999** as a close packing of voxels: 255 × 132 × 220 along the *x*-, *y*- and *z*-axis, respectively. The voxel itself is passed over via its

surfaces denoted as 2, with minus for "inside box". The voxel surfaces (macro body) are defined in the second block of the input file, similar to the definition of the large box, as follows:

```
2 rpp 0 0.2085  0 0.2085  0 0.8 $ (2.085x2.085x8)mm voxel, surfaces
```

The previous card for defining the lattice, after a continuation symbol **&**, is followed by the list of universes which is essentially an array of voxel data. Hence, the code will fill voxel by voxel the whole lattice according to the organ/tissue IDs separated by blanks in the input file. The filling progression is done, by definition, starting from the leftmost index, i.e. first along the *x*-axis, then the *y*-, and finally the *z*-axis. The array of IDs must be structured according to this convention.

The array size may be greatly reduced exploiting another very useful feature of the code: the repetition record **r**. Thus, such a "compressor" yields a compact text file of IDs which is approximately 2–3 MB for 7–12 million voxels phantom. The file may be linked with the input file by the **read** card. It is also important to skip the external voxels by assigning a zero universe.

Finally, we need to define the universes used in the lattice specification. In the first block of the input file we write:

```
1  10  -0.95   7   u=1    $ defines universe #1
5  6   -1.05   7   u=5    $ ... universe #5
```

Here in the first line, the cell numbered as 1 is assigned to the material numbered as 10 with mass density 0.95 g cm^{-3}. The cell defines universe number 1. Similarly, the second line defines cell 5 filled with material 6 and mass density 1.05 g cm^{-3}; the cell becomes universe 5 according to the last term of the line. It is important to bear in mind, that mass densities must be entered as negatives in accordance with the code notation. Here, both cells/universes, from the geometrical point of view, lie in the positive half-space of surface 7. The latter is assigned in the second block of the input file by means of any surface or macro body, in such a way that its positive sense would hold the entire phantom geometry. To speed up the tracking, it is sensible to define these filling universes via a single plane, e.g. card **7 px -100e2** will serve well if the phantom is defined above plane x = -100 m.

The above mentioned cards along with a set of material definitions completely specify the voxel phantom in MCNPX, i.e. three surface cards, one cell card defining a lattice, another cell card that fills the circumscribing box with the lattice, and, finally, a set of cell cards defining elementary universes used in the lattice description.

Several peripheral MCNPX cards may be useful:

- Linking of voxel data (list of universes) with the input file via the **read** card with the additional option **noecho** will prevent the output of a huge text body into the output file.

- Suppressing the print of some tables in production runs, like **print -128 -140**, will help to reduce the output file considerably, and, essentially, save operational memory during execution.

Last but not least, a word about the plotting capabilities of MCNPX. The recent code versions are able to plot the arrangement of materials as well as universes. These two-dimensional direct plots can greatly facilitate the debugging process. Using it, it is also very easy to check the orientation of the phantom which is crucial for the external source sampling discussed below.

2.2 Source sampling

The external source can be described by the standard MCNPX source card **sdef** along with cards for sampling distributions, if necessary. For example, the typical idealized external parallel broad beam of monoenergetic photons is made by using five cards:

```
sdef par=p erg=3 sur=2.1 x=53.1675 y=d2 z=d3 vec=-1 0 0 dir=1  $ 3-MeV ph, RLAT
si2     0  27.522   $ samples Y from 0 to 27.522 cm
sp2     0  1        $ samples Y uniformly
si3     0  176      $ samples Z from 0 to 176 cm
sp3     0  1        $ samples Z uniformly
```

Above, a so-called *right lateral* (RLAT) source is defined. The source particle is photon **par=p**, monoenergetic E_0 = 3 MeV (**erg=3**). The starting x-coordinate is fixed to 53.1675 cm, it is located on the surface number **2.1** of the bounding box that holds the whole phantom. The other two spatial variables, y and z, are sampled according to distributions number **2** and **3**, respectively. These uniform distributions are defined in the last four lines of the code above.

The other general types of external source are specified in a similar way using standard yet flexible source card capabilities. The unusual or optimized source sampling can be done by writing a specific Fortran code that is to be built into the code.

For internal source, the recent versions of MCNPX provide a very useful short hand notation for uniform sampling in all voxels of a particular organ or tissue—a typical task for internal exposure calculations. This is done by:

```
sdef  par=p  erg=0.1  cel=d5  x=d1  y=d2  z=d3  eff=1e-4  $ 100-keV ph, all voxels of id=11 (d5)
si5  L  (11<999<100)  $ all voxels of universe #11, within universe/lattice #999, within cell #100
sp5 1                 $ uniform sampling
#  si1       sp1       si2       sp2       si3       sp3       $ three distr-s for coord-s sampling
   0         0         0         0         0         0         $ coord-s start at low voxel border
   0.2085    1         0.2085    1         0.8       1         $ coord-s end at high voxel border
```

Here, the voxels are sampled uniformly throughout all voxels of a phantom; the voxel is rejected if the source path defined in the second line is not satisfied. The source path is defined via cell distribution **5** (note the **L**, which stands for List). It is important to bear in mind, that the default sampling efficiency in MCNPX is 0.001. Therefore, if the source organ consists of small number of voxels, the poor sampling will force the code to stop. That is the reason why, an additional option, namely **eff=1e–4**, was employed in order to bypass this generally very useful self-protection of the MCNPX for cost of source sampling. The numeric parameter of this option is chosen slightly lower than the quotient of the number of source organ voxels to the total number of voxels (including the voids).

Once the sampled voxel satisfies the source path, the actual coordinates, relative to the voxel position, are sampled according to uniform distributions **1**, **2** and **3** (x, y, z). Their description is written in the example above in the so-called vertical format, where the cards and corresponding parameters are aligned vertically.

In case of source sampling inside very small organs, it is more efficient to specify explicitly all voxels that belong to the particular ID. A small program—a tool—is then needed in order to extract the indices of all voxels of the source organ. The program can construct the exact input for MCNPX: a list of voxels, where each one has the following format:

```
(86<999[132  55  31]<100)
```

Here, one voxel is specified. This voxel belongs to universe **86** in universe/lattice **999**, it is the element with indices **[132, 55, 31]**, inside of cell **100**. The indices, *x*, *y* and *z* respectively, are written in accordance with the lattice definition. Once the list of voxels is ready, it must substitute the definition of the cell distribution, as follows:

```
read  file=id_86.src  noecho      $ include external file "id_86.src", don't echo into output
sp5   1  547r                     $ uniform sampling of 1+547 voxels
```

Here, the cell distribution number 5 is arranged by the tool in the external file according to the discussed format. The second line forces uniform sampling by assigning equal probabilities to all voxels (548 in this case).

2.3 Scoring

The way of scoring of an arbitrary quantity within a voxel phantom depends on how the phantom was specified. Generally, the mean organ doses are of interest. It is done by:

```
*f18:p,e  (1<999<100)  $ id=1    Adipose tissue, head
```

Here, the energy balance tally **8** is employed that scores deposited energy (asterisk in first position forces the output in units of MeV). This tally, number 18, scores in all voxels that satisfy path **1<999<100**, i.e. in the universe **1**, within the universe/lattice **999**, within the cell **100**.

Similarly, the other available types of tallies can be used. There is a limit for a number of tallies in the problem, but normally about a hundred organs at a time are followed without problem. A larger number of tallies may require code modification.

It is important to note that MCNPX is able to score the spatial distribution of the quantity of interest in each voxel of a phantom. Recently, this capability was extended. Now, starting from version 2.5.f, it applies to the energy balance tally as well, which is essential for simulations where there is no charged particle equilibrium. By employing this feature, with one card, it is possible to score in all voxels separately, namely:

```
*f8:p,e  (999<999[0:254  0:131  0:219]<100)      $ scores in each voxel
```

Here, the path in parenthesis is similar to those described above. All index bounds must be explicitly presented and synchronized with the definition of the corresponding lattice.

It is advisable, in the case of spatial distribution scoring, to utilise the input card **talnp** (tally no print) which will prevent printing very lengthy results in the output file. The output can be extracted via an auxiliary so-called MCTAL file using the card **prdmp 2j 1**.

3 MCNPX VOXEL MODELS: KLARA AND GODWIN

Two voxel phantoms developed in the Medical Physics Group of GSF Research Center, namely the female phantom Laura and the male phantom Golem, have been recently adjusted according to the Reference Man anthropomorphic data of the International Commission on Radiological Protection (ICRP). In order to distinguish the original well-benchmarked phantoms and their adjusted modifications, the new names Klara and Godwin were introduced (Zankl 2004). The female phantom is named Klara (from German "kleine Laura", which stands for small Laura;

because the adjusted modification is somewhat lower). Godwin (which resembles Golem) is the male voxel phantom.

Godwin is defined on mesh: 255 × 132 × 220. His voxel dimensions are (2.085 × 2.085 × 8) mm. He has 7.4 million voxels in total, of which only 2 million are non-void. The voxels are assigned to 85 organs or tissues (Fig. 1).

Klara is defined on a finer mesh: 244 × 138 × 346. Her voxel is (1.765 × 1.765 × 4.84) mm. She has 11.7 million voxels, of which 3.9 million are non-void. The voxels assigned to 89 organs or tissues (Fig. 1).

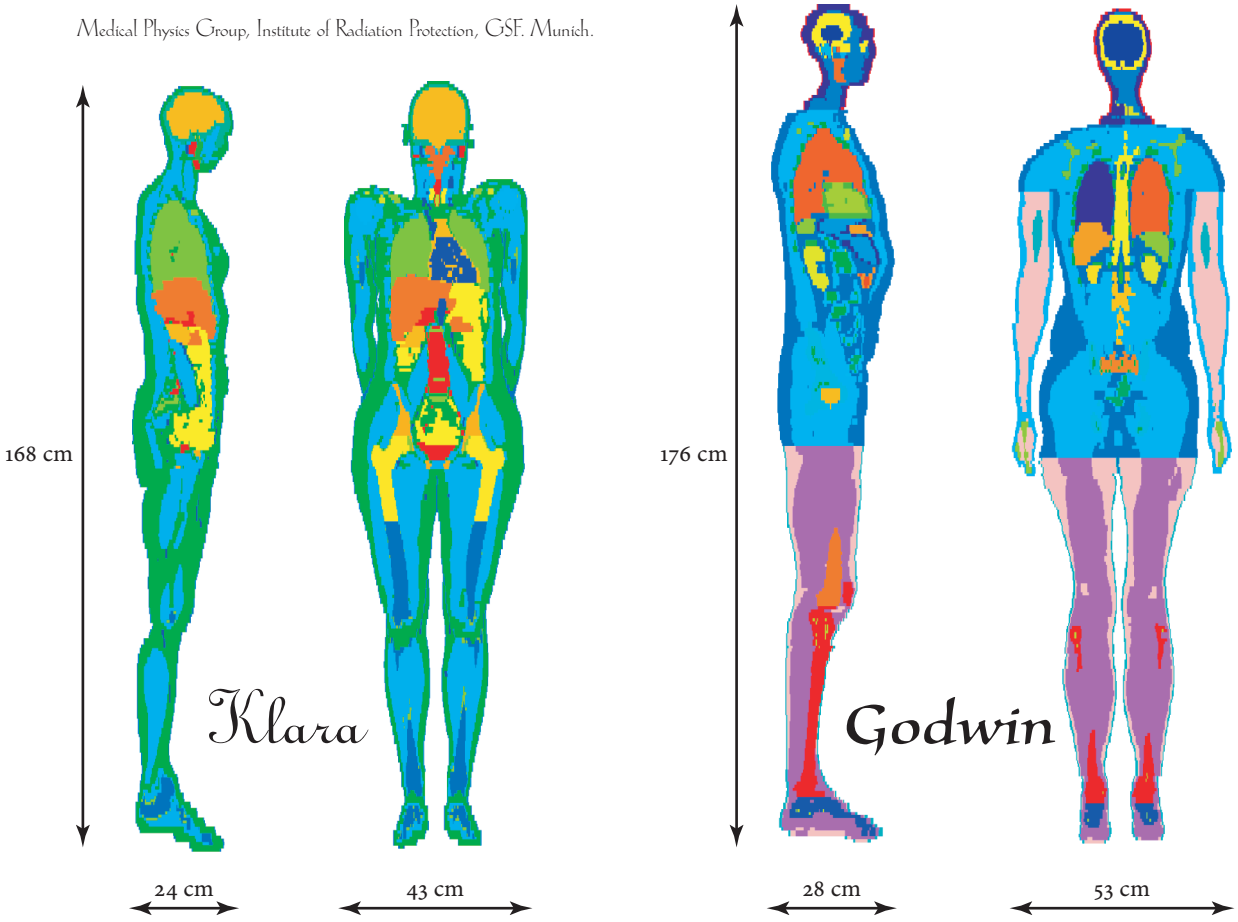

Figure 1. A direct geometry plot from MCNPX ensures the right conversion of the voxel data into the code. In the left half, the mapping of materials over the geometry is shown for the female phantom Klara. In the right half, organ assignment is shown for the male phantom Godwin.

Both phantoms were clipped to get rid of void voxels especially in front of the phantom. This greatly reduces the memory requirements and increases the speed of particle tracking. The voxels of both phantoms were assigned to approximately 30 materials; half of that belongs to the skeleton. The skeleton materials were evaluated separately for Klara and Godwin, whereas the rest materials for the soft-tissue organs are the same for both models.

The phantoms have been converted into MCNPX pre-release version 2.5.f according to the description above. The simulations for both phantoms run successfully on a personal computer (Pentium IV, 3 GHz, 2 GB RAM). Several idealized external photon irradiations were considered: antero-posterior AP, posterior-anterior PA, left lateral LLAT and right lateral RLAT

for seven energies in the range 0.01–3 MeV. Simulations of internal irradiation were carried out as well.

All calculations employed coupled photon-electron transport mode with the default detailed photon transport regime. The standard MCPLIB03 library of MCNP version 4C was used. The photon and electron energy cut-offs were 1 and 60 keV, respectively. The CSDA range of 60 keV electron in lung (mass density 0.4 g cm^{-3}) is 0.14 mm which is more than ten times shorter than the smallest dimension of a voxel, thus this energy cut-off is safe. The energy balance estimator (***f8**) was employed in all MCNPX calculations.

For the comparison, similar calculations were made using EGSnrc code system (Kawrakow and Rogers 2003) version 4.1.1.0 with the updated photon cross-section libraries (Seuntjens *et al.* 2002). Essentially the same voxel data and material sets were utilised. The standard transport parameters were used in the coupled photon-electron mode. The photon and electron energy cut-offs were 1 keV. The number of histories for both MCNPX and EGSnrc were set to ten million.

4 RESULTS OF MCNPX SIMULATIONS

The MCNPX running time in case of external irradiation geometry (AP, PA, LLAT, RLAT) was approximately 1–2 hours per 10 million source photons (per process), which is acceptable. The MCNPX simulations for internal exposure for thyroid, liver and bladder content showed similar timings. The runs for the male phantom Godwin were slightly faster compared to corresponding runs for Klara which has finer resolution.

For scoring of mean organ doses, for external source, the peak amount of operational memory did not exceed 1 GB, whereas during the transport phase of the run program occupies about 500 MB (per process). Preliminary simulations for a spatial dose distribution (dose in each voxel) showed higher memory requirements of the order of 1–2 GB. The runs for internal sources showed similar memory requirements. However, depending on the tables that are to be print, the size of a program in operational memory can be larger by a factor two or more.

The results of comparison between MCNPX and EGSnrc are presented below in Figs. 2–6. In these figures, the percent difference is calculated relative to the EGSnrc values, and two sigma uncertainties are shown.

Figs. 2 and 3 illustrate the differences in results for external irradiations: AP for Klara and LLAT for Godwin. These two figures are typical. The other combinations of source geometries and phantoms show a similar tendency, i.e. a good agreement in the range 0.1–3 MeV and systematic, statistically valid discrepancies at the lower energies of 10, 40 and 60 keV (results for 10 and 100 keV are not shown). The latter discrepancy is attributed to the defect (in this range of energies) of photon cross sections in the MCPLIB03 library that was employed. This library underestimates the photoeffect by approximately 10% for elements of low atomic number such as tissues of a human phantom. The problem of low energy cross sections was well investigated in the following studies: DeMarco *et al.* 2002, Bohm *et al.* 2003, Reniers *et al.* 2004.

Three figures for internal exposure (Fig. 4–6) show a similar tendency, despite higher uncertainties.

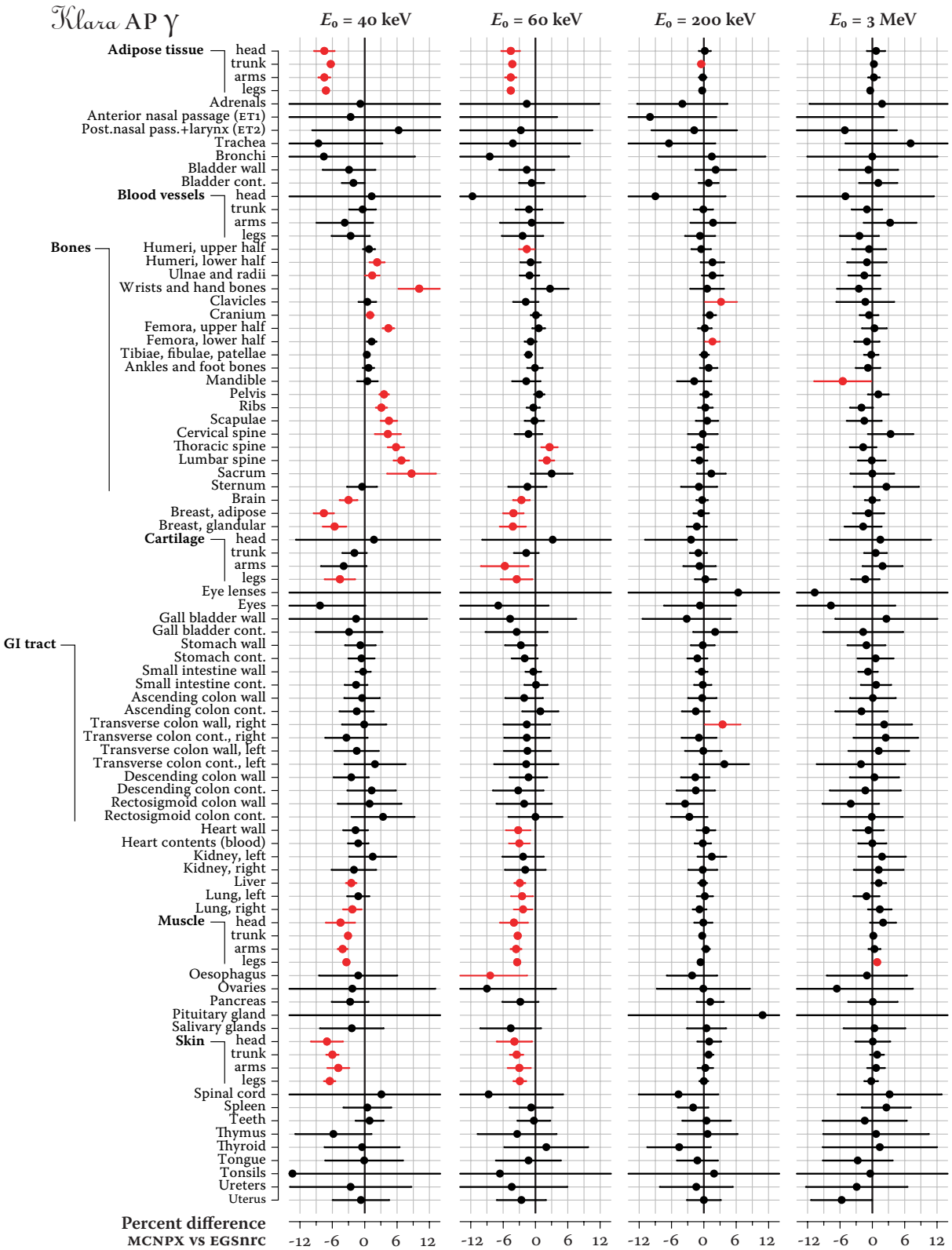

Figure 2. Comparison of organ doses calculated with MCNPX and EGSnrc for *antero-posterior* external photon irradiation of the female phantom Klara. The EGSnrc results were taken as reference. Two sigma uncertainties are shown as lines. Cases of statistically significant discrepancy are shown in red.

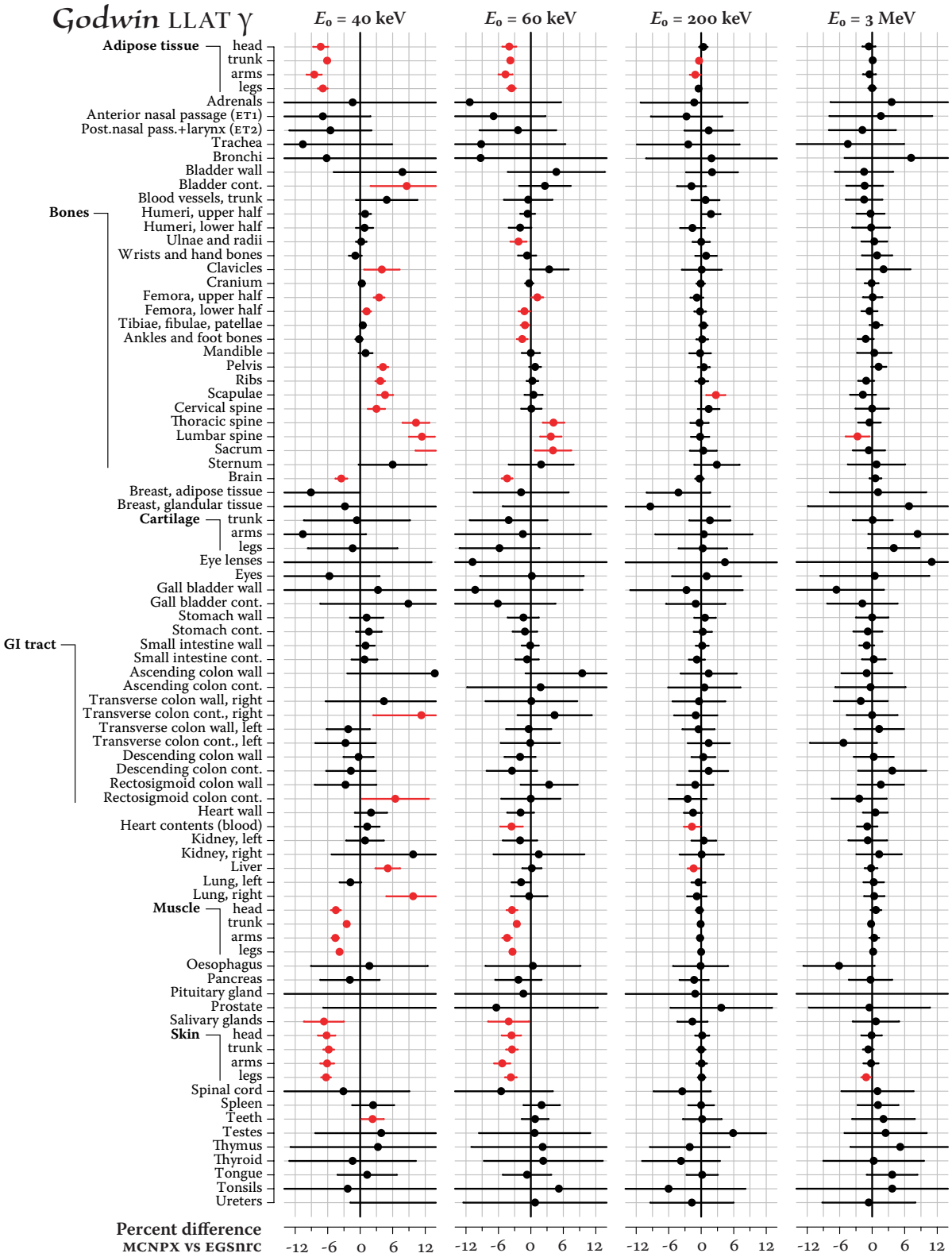

Figure 3. Comparison of organ doses calculated with MCNPX and EGSnrc for *left lateral* external photon irradiation of the male phantom Godwin. The EGSnrc results were taken as reference. Two sigma uncertainties are shown as lines. Cases of statistically significant discrepancy are shown in red.

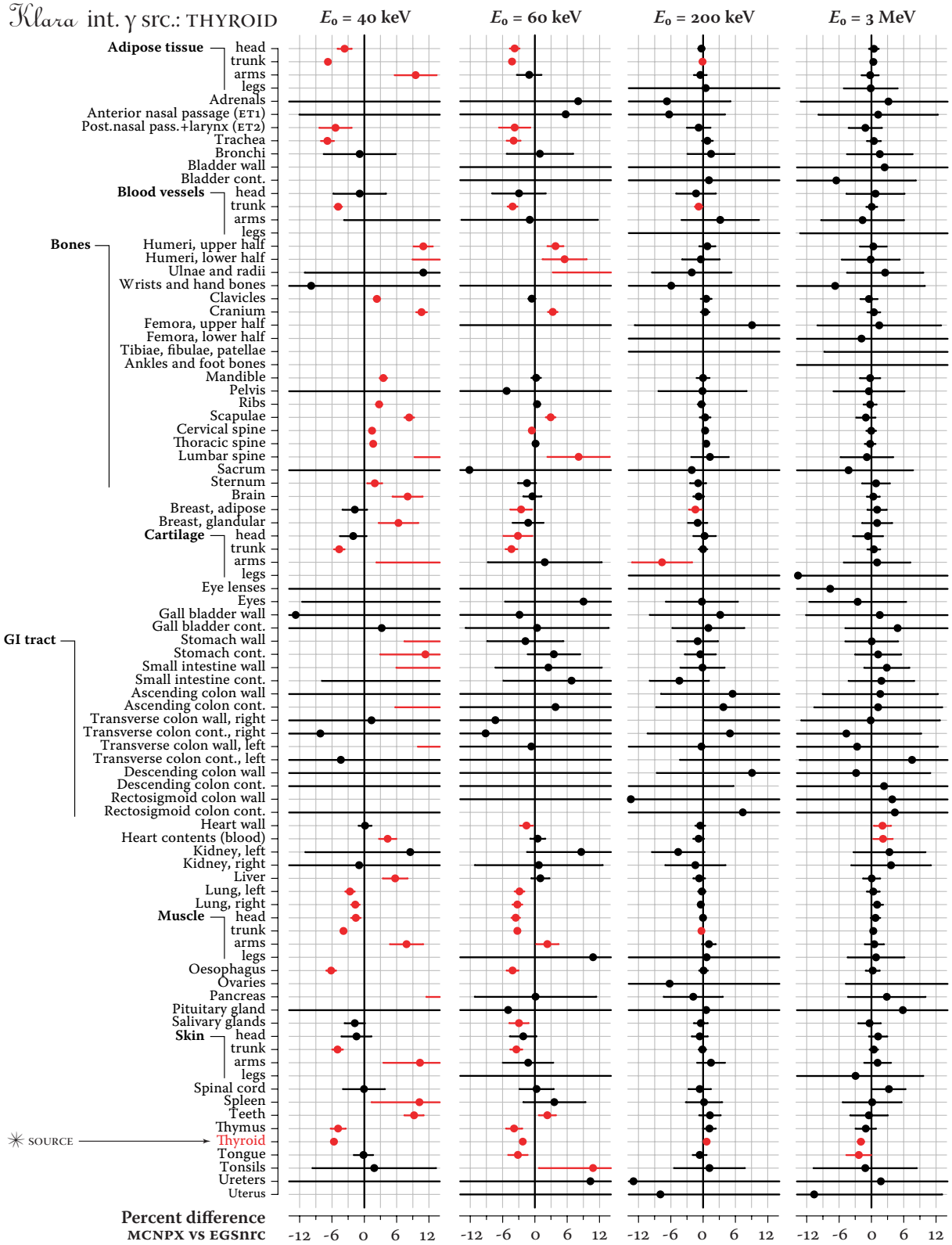

Figure 4. Comparison of organ doses calculated with MCNPX and EGSnrc for *thyroid* as internal photon source in the female phantom Klara. The EGSnrc results were taken as reference. Two sigma uncertainties are shown as lines. Cases of statistically significant discrepancy are shown in red.

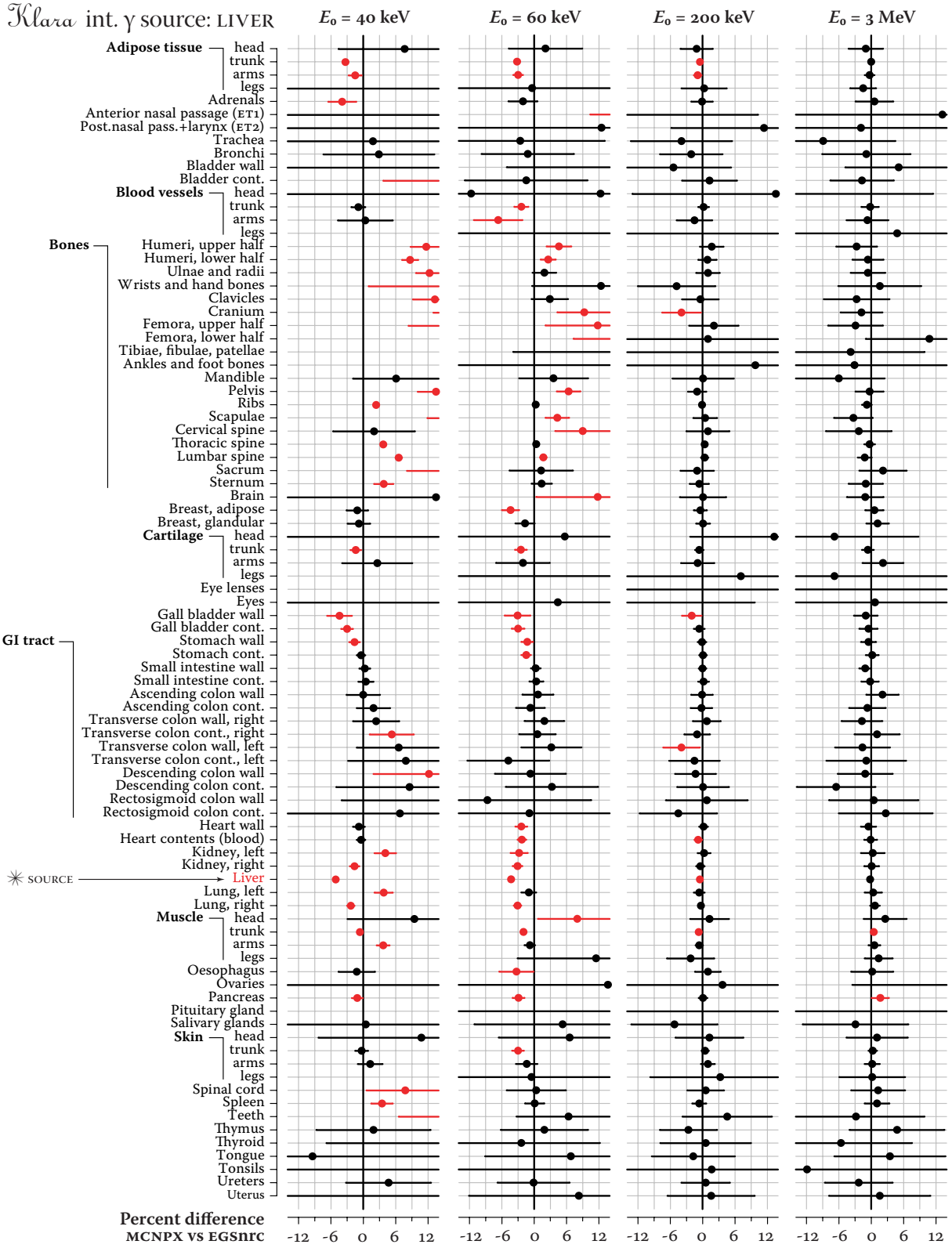

Figure 5. Comparison of organ doses calculated with MCNPX and EGSnrc for *liver* as internal photon source in the female phantom Klara. The EGSnrc results were taken as reference. Two sigma uncertainties are shown as lines. Cases of statistically significant discrepancy are shown in red.

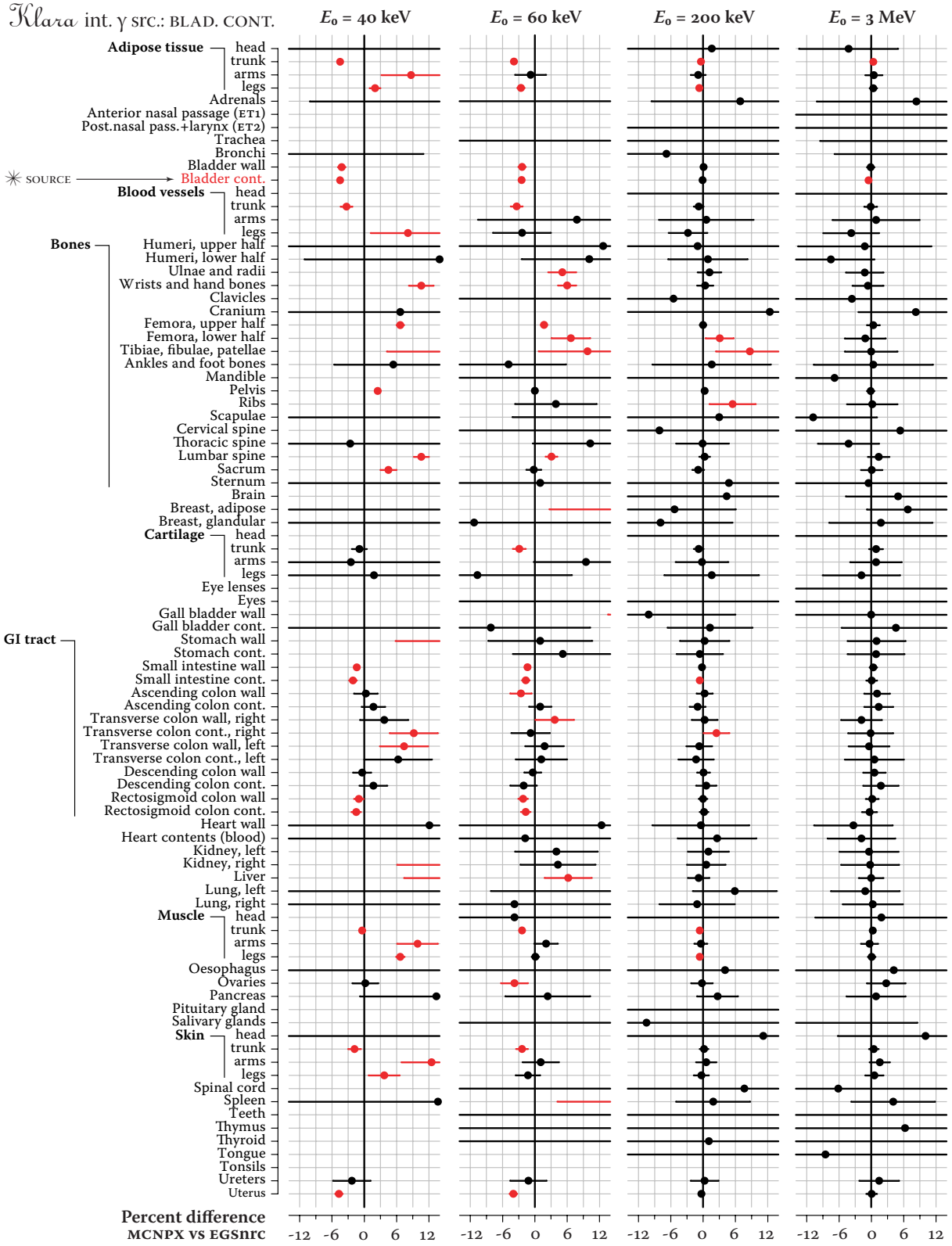

Figure 6. Comparison of organ doses calculated with MCNPX and EGSnrc for *bladder content* as internal photon source in the female phantom Klara. The EGSnrc results were taken as reference. Two sigma uncertainties are shown as lines. Cases of statistically significant discrepancy are shown in red.

5 CONCLUSIONS

The setup procedure of voxel phantom in MCNPX is described and illustrated. Exemplary results of MCNPX in terms of difference with analogous EGSnrc calculations are presented for external as well as internal irradiation with photons for the female phantom Klara and the male phantom Godwin. Good agreement was found when the primary photon energy is in the range 0.1–3 MeV. However, in the low-energy region, systematic discrepancies between doses calculated using MCNPX and EGSnrc exist for many organs. About one third of all organs have got statistically different doses (on the level of 95% confidence) calculated by MCNPX versus EGSnrc for 40 and 60 keV external sources AP, PA, LLAT and RLAT. This discrepancy is thought to be attributed to the MCPLIB03 photon cross-section library that was used and which is considerably underestimates the photoeffect below about 80 keV. Its update is vital.

The preliminary comparisons with the published results on dose conversion coefficients for the original predecessors of Klara and Godwin (phantoms "Laura" and "Golem") showed good agreement, again except for the low-energy region, and will be reported soon.

6 ACKNOWLEDGMENTS

The authors would like to thank Dr. G. McKinney—a primary developer of MCNPX at LANL—for the fruitful discussion of the latest code improvements related to the voxel model construction and scoring.

This work has been financially supported by the German Federal Office of Radiation Protection under contract no. StSch 4417.

7 REFERENCES

Bohm T D, DeLuca Jr P M and DeWerd L A, "Brachytherapy dosimetry of ^{125}I and ^{103}Pd sources using an updated cross section library for the MCNP Monte Carlo transport code", Med Phys, **30**(4), pp. 701–711, (2003).

Briesmeister J F (ed), "MCNP—a general purpose Monte Carlo n-particle transport code. Version 4C", LA-13709, Los Alamos, NM, USA, (2000).

Burn K W , Daffara C, Gualdrini G, Pierantoni M, "Radiation transport in voxel geometries: an integrated approach based on an extended version of MCNPX", RT/2002/50/FIS, ENEA, Bologna, Italy, (2002).

DeMarco J J, Wallace R E and Boedeker K, "An analysis of MCNP cross-sections and tally methods for low-energy photon emitters", Phys Med Biol, **47**, pp. 1321–1332, (2002).

Hendricks J S *et al.* "MCNPX extensions, version 2.5.0", LA-UR-04-0570, Los Alamos, NM, USA, http://mcnpx.lanl.gov, (2004).

Kawrakow I and Rogers D W O, "The EGSnrc code system: Monte Carlo simulation of electron and photon transport", *National Research Council of Canada*, PIRS-701, http://www.sao.nrc.ca/inms/irs/EGSnrc/pirs701.pdf, (2003).

Reniers B, Verhaegen F and Vynckier S, "The radial dose function of low-energy brachytherapy seeds in different solid phantoms: comparison between calculations with the EGSnrc and

MCNP4C Monte Carlo codes and measurements", Phys Med Biol, **49**, pp. 1569–1582, (2004).

Seuntjens J P *et al.*, "Calculated and measured air-kerma response of ionization chambers in low and medium energy photon beams", *Proceedings of an International workshop "Recent developments in accurate radiation dosimetry"*, Medical Physics Publishing, Madison, WI, USA, pp. 69–84 (2002).

Waters L S (ed), "MCNPX User's Manual, version 2.3.0", LA-UR-02-2607, Los Alamos, NM, USA, (2002).

Zankl M, Becker J, Fill U, Petoussi-Henss N and Eckerman K F, "GSF male and female adult voxel models representing ICRP reference man", *Topical Meeting "Monte Carlo 2005",* American Nuclear Society, Chattanooga, TN, USA, April 17–21 2005, (to be appeared in this proceedings).