Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2015-0011

Vol. **21**, No. 5, pp. 1475-1488
May 2017

Computational Software

FXRS: Fast X-Ray Spectrum-Simulator Theory and Software Implementation

Ciprian C. Chirilă[1,*] and T. M. H. Ha[2]

[1] *Institute of High Performance Computing, Agency for Science, Technology and Research, 1 Fusionopolis Way, #16-16 Connexis, Singapore 138632.*
[2] *Singapore Institute of Manufacturing Technology, 2 Fusionopolis Way, Singapore 138634.*

Communicated by Michel A. Van Hove

Received 19 November 2015; Accepted (in revised version) 8 May 2016

Abstract. We propose a simple, computationally efficient scheme for an X-ray spectrum simulator. The theoretical models describing the physical processes involved are employed in our Monte Carlo software in a coherent way, paving the way for straightforward future improvements. Our results compare satisfactorily to experimental results from literature and to results from dedicated simulation software. The simplicity, excellent statistical errors, and short execution time of our code recommend it for intensive use in X-ray generation simulations.

AMS subject classifications: 68U20, 65C05

Key words: Electron-matter interaction, X-ray generation, Monte Carlo simulation, interaction forcing, detection forcing.

Program Summary

Program title: FXRS: Fast X-Ray Spectrum-Simulator

Nature of problem: Simulation of X-Ray energy and angle-resolved emission spectra. Optionally, the deposited energy as a function of depth can be computed.

Programming language(s): Fortran 90

Computer platform: Any

Operating system: Any

[*]Corresponding author. *Email addresses:* `chirilacc@ihpc.a-star.edu.sg` (C. C. Chirilă), `tmhha@simtech.a-star.edu.sg` (T. M. H. Ha)

Compilers: gfortran

RAM: 4.5 MB

External routines/libraries: Xraylib, ELSEPA

Running time: The execution time for 10^5 injected electrons with energy of 20 keV is about 1 minute on an Intel Xeon X5650 processor (2.67 GHz).

Restrictions: The low-energy spectrum is not describable with the physics included in the present software.

Additional Comments: The software is very modular, and can be easily changed for research purposes.

1 Introduction

X-rays are widely used in various applications such as non-destructive structural analysis, imaging, microscopy and radiography, *etc*. In any application, especially in metrology measurement and diagnostic, performance of the technique-in-use strongly depends on the quality of X-ray beam. Understanding of the X-ray beam quality therefore plays an important role in interpretation of data, correction of artifacts as well as optimization of X-ray source design. The quality of X-ray beam varies from one to another depending on various parameters such as anode material, filtration, high voltage and focusing system. Experimentally investigation of how X-ray generation is correlated to the specifications of a given X-ray source would be difficult and time consuming, as well as not cost effective. This difficulty can be overcome by performing computer simulation of X-ray spectra.

A few codes are available as open source for X-ray generation simulation [1–3]. However, many of these codes are designed for a wide scope of application, and consequently grow in size and complexity. Hence, the user who is interested to implement new physical models, for example, faces a difficulty. Moreover, the other state-of-the-art general-purpose codes with coupled electron-photon transport like Geant4, PENELOPE, MCNPX [4–6] are generally computationally expensive. For this reason, we pursue here the goal to create FXRS (Fast X-Ray Simulator): a modular software simulating the emission of X-rays under electron impact, with allowance for more sophistication to be added, as future requirements of the users dictate. This way, we have access to all the variables that describe the interactions and to the way the output is generated. By employing interaction and detection forcing method, our code can deliver results with very good statistics in fast execution time.

Atomic units are used throughout this work, unless otherwise specified.

2 Monte Carlo method: Variance reduction techniques

Monte Carlo methods are a class of methods that simulate real-life processes by means of random sampling. In the case of the simulation of X-ray spectra, Monte Carlo method finds direct application since the process of the emission of X-ray is inherently random. However, the cross sections of different processes involving in X-ray emission vary widely over a range of many orders of magnitude as represented in Fig. 1. There is almost three orders of magnitude difference between elastic scattering (black curve) and bremsstrahlung (blue curve) cross sections. It means that out of 10^3 simulated scattering events, only about 1 actually contributes to the emission of X-ray photons. Such a scenario is disadvantageous to the simulation, since one needs a large number of photons in order to have a small statistical error. To overcome this, we adopted two powerful methods: interaction forcing and detection forcing [8] to save significant computer time while little programming effort is needed.

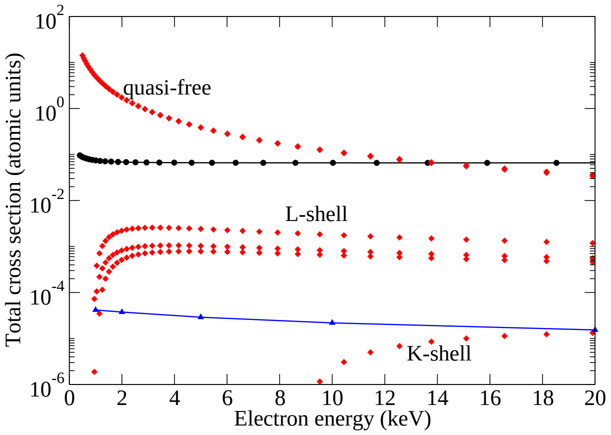

Figure 1: Comparison of total cross section for various processes in Cu. Black circles: elastic scattering, red diamonds: inelastic scattering (comprising valence-electron scattering or quasi-free, L-shell scattering showing the three sub-shells, and K-shell scattering), and blue triangles: bremsstrahlung cross section.

2.1 Interaction forcing

In interaction forcing algorithm, cross section of low probability events are artificially increased so that they become statistically relevant. At the same time, the studied quantity attached to that rare event acquires a statistical weight such that the overall the average value is not changed. This technique has been implemented in our code as following:

- Introduce forcing factor $F>1$ to total cross section of rare event, example K-shell ionization;
- Change direction of primary electron only if $\xi<1/F$ (with ξ a random number);
- Weight any emitted photon with w_{el}/F where w_{el} is the weight of the parent electron (initially, the injected electron has weight equal to unity).

Any interaction that is less probable can be artificially enhanced in this manner, sometimes by significantly large forcing factors. The statistical error of the quantities improves significantly compared to the case when the interaction is not forced. This technique must be followed by *detection forcing* for the successfully simulation of the bremsstrahlung emission spectrum.

2.2 Detection forcing (bremsstrahlung)

Detection forcing is related to the fact that X-ray photons are emitted in a 4π solid angle, while the detector only occupies a small solid angle. The principle of this method is to force each and every emitted X-ray photons to the detector. Each photon is weighted with a factor corresponding to its probability of hitting the detector and is calculated as:

$$\frac{p(\Delta S)}{\Delta S}=\frac{1}{\sigma}\frac{d\sigma}{d\Omega}\frac{1}{R^2}\exp(-\mu l), \tag{2.1}$$

where $\sigma^{-1}d\sigma/d\Omega$ is normalized differential cross section, R and ΔS represent detector position and detector surface respectively, l is the length the photon travels in the medium and μ is the absorption coefficient in the material at the corresponding photon energy. Note that in Eq. (2.1) the result is independent of the considered detector area ΔS, which may be taken arbitrarily small. Therefore, this approach gives the X-ray fluence at the detector exactly.

3 Physics of X-ray generation

The physical processes that control X-ray generation by electron impact are elastic scattering, inelastic scattering, and bremsstrahlung. In addition, Coster-Kronig transitions, Auger electrons, and emission of characteristic X-ray radiation need to be accounted for. In the following, we focus on describing the way each of these processes could be simulated optimally in a computer program in order to ensure a fast execution time with low memory usage.

3.1 Elastic scattering

In the elastic scattering the electron collides with an atom, changing its direction, while keeping its kinetic energy unchanged. By using a Random Number Generator (RNG) [7], the type of interaction (elastic, inelastic, or bremsstrahlung) is chosen at random, according to the relative ratios of the total cross sections (TCS) of each process.

An accurate description of elastic scattering is given by Salvat *et al.* in the `FORTRAN77` code ELSEPA [9]. This code can calculate accurately the differential cross section (DCS) of elastic scattering and the total cross section for a given impact energy of the electron. We have implemented a new interface to this code, to make its use more suitable to our purpose. As a result, the DCS and TCS are stored as tabulated tables with 15 energy points per decade, (*i.e.*, the tabulated energies vary as $E_{\mathrm{n}+1}=E_1\times 10^{\mathrm{n}/15}$). The tables are stored for later use in an `UNFORMATTED` form (useful `FORTRAN` feature), such that the computer memory used is negligible.

For the X-ray simulation one has to sample the scattering angle. Interpolation over the scattering angle θ is used. To be more precise, it was found that it is more advantageous to use the variable $\mu=(1-\cos\theta)/2$ instead of θ as the DCS variation with μ is much smoother than with θ. Let us note $p(\mu)$ the DCS with respect to μ. Then, the total DCS up to μ is $P(\mu)=\int_0^\mu p(\mu)d\mu$ and using quadratic interpolation:

$$P(\mu)=P(\mu_i)+p(\mu_i)(\mu-\mu_i)+\frac{p(\mu_{i+1})-p(\mu_i)}{\mu_{i+1}-\mu_i}\frac{(\mu-\mu_i)^2}{2}, \tag{3.1}$$

with the indexed quantities tabulated. With ξ a random number, one can easily solve now $\xi=P(\mu)$ to sample the scattering angle. Nonetheless, if the scattering electron's energy is not tabulated, one must first sample in energy (using interpolation by weight technique), and afterward sample the angle. As described in [10] and references therein, let us assume we have an electron with the energy E not tabulated, such that $E_j<E<E_{j+1}$ (E_j's are tabulated). Instead of interpolating the angular distribution from the E_j and E_{j+1} angular grids, we use interpolation by weight $P(E)=\pi_j P(E_j)+\pi_{j+1}P(E_{j+1})$, where $\pi_j=\ln(E_{j+1}/E)/\ln(E_{j+1}/E_j)$ and $\pi_{j+1}=\ln(E/E_j)/\ln(E_{j+1}/E_j)$. Since these "interpolation weights" are positive and add to unity, they can be interpreted as point probabilities. One can thus sample the variable k that can take values j or $j+1$ with point probabilities π_j and π_{j+1}, respectively. Once k is sampled (choosing the value of the tabulated energy), the scattering angle is sampled by inverting the cumulative distribution function for the corresponding tabulated energy.

In conclusion, we have implemented a *very efficient and accurate* code for the simulation of the process of elastic scattering.

3.2 Inelastic scattering

In the process of inelastic scattering, an electron scatters on an atom, changing its kinetic energy and direction. The energy loss per unit path length of an electron with kinetic energy E along its trajectory is calculated by using relativistic Beth stopping power:

$$\begin{aligned}\frac{dE}{ds}=-\frac{2\pi ZN}{c^2\beta^2}\Bigg\{&\ln\left(\frac{\beta^2}{1-\beta^2}\frac{c^2E}{I^2}\right)-\left[2(1-\beta^2)^{1/2}+\beta^2\right]\ln 2\\ &+(1-\beta^2)+\frac{1}{8}\left[1-(1-\beta^2)^{1/2}\right]^2\Bigg\},\end{aligned} \tag{3.2}$$

where N is the number of atoms with atomic number Z per volume, β is the relativistic factor, and I is the mean excitation potential [11]). The Bethe stopping power can be used in order to perform a simplified simulation without the emission of X-rays, mostly to obtain the distribution of the ejected electrons (*e.g.*, see [11]). In the present study, its use is essential in order to simulate the inelastic scattering of the impact electron on the valence/quasi-free electrons in the material. In other words, after summing up the contributions to the stopping power coming from inelastic scattering on the K, L, ... shells (with the sub-shells included, except for the K-shell), the remainder after subtracting the sum from the Bethe stopping power represents the contribution of the scattering by the quasi-free electrons.

3.2.1 Electron impact ionization

Upon impact of an electron, an occupied shell (like K, L, $\cdots$) can be ionized. For the simulation of the X-ray generation, it is important to know the total ionization cross section for individual shell (and sub-shell), and also the differential cross section (giving information on the partition of total energy between the projectile and the ionized electron, as well as their scattering angles after the ionization event).

The calculation of the TCS for such an event is extremely complicated. A very popular and useful approach is to fit the available data for any given sub-shell, atomic charge number Z and electron energy E. The fitted values are either experimental or calculated. For the latter case, see [12–14]. For the former, such an easy to implement approach is given in [15], which is given below (correcting some typos in the original article, after contacting the authors [16]):

$$\begin{aligned}\sigma(\text{a.u.}) = \frac{4\pi N}{c^4(\beta_t^2 + k\beta_b^2)2b'} \Bigg\{ \frac{1}{2}\left[\ln\frac{\beta_t^2}{1-\beta_t^2} - \beta_t^2 - \ln 2b'\right] + (1 - \frac{1}{t^2}) + 1 - \frac{1}{t} \\ - \frac{\ln t}{t+1}\frac{1+2t'}{(1+t'/2)^2} + \frac{b'^2}{(1+t'/2)^2}\frac{t-1}{2}\Bigg\},\end{aligned} \tag{3.3}$$

where

$$\begin{aligned} t' = \frac{T}{c^2}; \quad \beta_t^2 = 1 - \frac{1}{1+t'^2}; \quad t = \frac{T}{B}; \\ b' = \frac{B}{c^2}; \quad \beta_b^2 = 1 - \frac{1}{1+b'^2}; \quad k = \frac{C_{nlj}}{B}, \end{aligned}$$

with N the number of electrons on the sub-shell, T the kinetic energy of the electron and B the ionization energy. Furthermore, $C_{nlj} = 0.3Z^2_{\text{eff,nlj}}/(2n^2) + 0.7Z^2_{\text{eff,n'l'j'}}/(2n'^2)$ $(n'l'j'$ is the next energetic level) where Z_{eff} is the effective nuclear charge that represents the electronic shielding and electronic correlation.

To address the DCS for inelastic scattering, we make use of the Gryzinski cross section [17]:

$$\sigma \propto (1-\epsilon)^{\epsilon_i/(\epsilon_i+\epsilon)} \left[\frac{\epsilon}{\epsilon_i}(1-\epsilon_i) + \frac{4}{3}\ln\left(E + \sqrt{\frac{1-\epsilon}{\epsilon_i}}\right)\right] / \epsilon^3, \tag{3.4}$$

with $\epsilon_i = B/T$ and $\epsilon = \Delta T/T$ (ΔT being the energy loss). The proportionality constant (prefactor) is calculated such that TCS calculated using the Gryzinski formula is identical to that given by Eq. (3.3). The scattering angles are given by the classical formulas: the projectile electron's angle $\cos\theta = E_f(E_i+2c^2)/(E_i(E_f+2c^2))$, and the angle of the ionized electron $\cos\theta = \Delta E(E_i+2c^2)/(E_i(\Delta E+2c^2))$. Here, E_i is the initial energy of the electron, ΔE is the energy lost, and $E_f = E_i - \Delta E$ is the final energy of the projectile electron. The azimuthal angles ϕ are random numbers chosen from 0 to 2π.

The effect of the valence shells are taken into account in a way suggested in [18]. Since the valence electrons are weakly bound, we can describe the scattering as a Møller cross section for free electrons:

$$\frac{d\sigma_{\text{free}}}{d\epsilon} \propto \frac{1}{\epsilon^2} + \frac{1}{1-\epsilon^2} + \left(\frac{\gamma-1}{\gamma}\right)^2 + \frac{2-4\gamma}{2\gamma^2}\frac{1}{\epsilon(1-\epsilon)}, \tag{3.5}$$

where $\epsilon = \Delta E/E$ (E being the incident electron kinetic energy and ΔE the energy loss) and γ is the relativistic factor. We did not specify the prefactor in Eq. (3.5) since we use the Bethe formula to determine and tabulate its values. We ask that the stopping power for the inelastic scattering on the K, L, and M shells plus the stopping power of the free-free scattering add to the value given by the Bethe formula (3.2). For any process (inelastic for this example), the stopping power reads $dE/ds = \rho\int_{\epsilon_{\min}}^{\epsilon_{\max}} E\epsilon(d\sigma_{\text{inel}}/d\epsilon)$ (ρ is the number of atoms per unit volume). The upper integration limit is $\epsilon_{\max} = 0.5$ (identical electrons in free–free collisions) and the lower limit has to be non-zero (since the cross section becomes divergent). We take $\epsilon_{\min} = 10^{-2}$. For the Møller scattering, we calculated the Bethe stopping power analytically, while for the shell ionization we use numerical quadrature (this is the reason that we choose to tabulate the values, since numerical quadrature is time-consuming).

Last, we have to sample the energy loss from the free-free scattering. With $g_2 = (E/c^2)^2$ and $v_{\text{M}} = 1+5/4g_2$, the sampling is done as follows: let $x = \epsilon_0/(1-\xi(1-2\epsilon_0))$ and $v = 1+(x/(1-x))^2+g_2x^2+(g_2-1)x/(1-x)$. If $v < \xi v_{\text{M}}$, a new random number ξ is generated until the inequality is not satisfied. The sampled value of the energy loss is then xE. This way of sampling is 10× faster than the rejection sampling method applied directly to free-free scattering.

3.2.2 Coster-Kronig transitions and characteristic X-Ray emission

The atom after inelastic scattering is left in an excited state. As a result, one electron from higher shells will fill the hole, results the emission of a characteristic X-ray photon with a specific energy; or emission of an Auger electron that is set free in the continuum. Before the de-excitation, another process can take place: the Coster-Kronig transition where the remaining electrons on the sub-shells re-arrange themselves according to specific transition probabilities. This process must be included in the simulation *before* taking into account the emission of the characteristic X-ray radiation. The procedure is described in detail in [19].

3.3 Bremsstrahlung

The DCS of the process is complicated, since it contains a distribution over many variables (photon energy and emission direction, the new direction of the electron after the emission). We have adopted the analytical modified Bethe-Heitler DCS with Kirkpatrick-Wiedmann-Statham (KWS) angular distribution [20]

$$p_{\mathrm{KWS}} = N_\theta \frac{\sigma_x(1-\cos^2\theta)+\sigma_y(1+\cos^2\theta)}{(1-\beta\cos\theta)^2} \tag{3.6}$$

with the prefactor

$$N_\theta = \frac{\beta^3(1-\beta^2)}{2\{2\beta\sigma_y - 2\beta(1-\beta^2)\sigma_x + [\ln(1-\beta)-\ln(1+\beta)](1-\beta^2)(\sigma_y-\sigma_x)\}} \tag{3.7}$$

such that the normalization with respect to $\cos\theta$ is ensured. The sampling technique for the emission angle of the photon is also explained in Appendix A of [20]. To sample the energy of the X-ray photon, we use the tabulated values of Seltzer and Berger [21] and the same techniques are employed as for the elastic scattering sampling (interpolation). Once an X-ray photon is emitted, it can interact with the material in various ways [22]. To simplify, we take into account the overall effect in the form of an exponential attenuation in the material [23].

4 Software implementation

In this section we give brief details about the workflow in our code. Some subroutines are detailed as well. We designed the code such that its parallelization is simple to implement if necessary. The execution time for 10^5 injected electrons with energy of 20keV is about 1 minute on an Intel Xeon X5650 processor (2.67 GHz). Fig. 2 shows the gross details of our X-Ray simulation software. The incident electron is tracked until its energy drops below an absorption threshold. When a secondary electron is created, we include it in the propagation scheme. We use to this end a linked-list structure implemented in FORTRAN90 with the help of pointers. The new electron is added immediately after the head of the list that represents the parent electron. When it is absorbed, its `ACTIVE` flag (a logical variable, part of the 'electron' data-structure) is set to `FALSE` and at the next parsing of the list for scattering-event sampling, it is skipped. When all the electrons (parent and secondary) are absorbed, the list is deallocated and the memory is set free. If the latter step is not taken, a large amount of memory is unnecessarily blocked. After the simulation is done for the injected electron, its X-ray bin is added to the global X-ray bin, which after simulation of all injected electrons will be subjected to statistical analysis in order to obtain the final X-ray spectrum and the statistical errors.

After the spectrum is obtained, further analysis of the data must be done such as correcting for the detector efficiency. This operation is straightforward to implement in

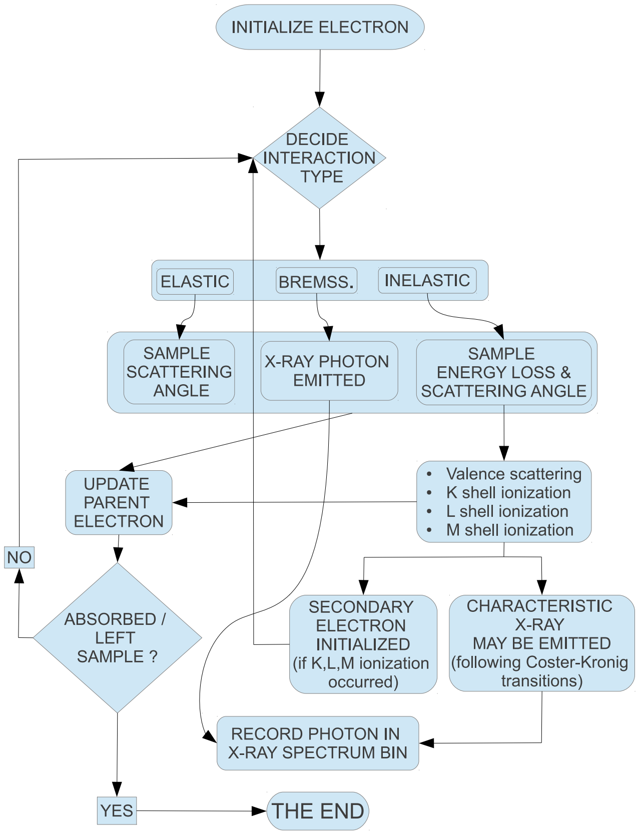

Figure 2: Work flow of the X-Ray spectrum simulator.

the form of a convolution of the simulation results with the detection efficiency curve that accompanies any X-ray detector.

The software allows studying the variation of X-ray intensity as a function of the detector angle. This way, expensive materials need not be used in the laboratory in time-consuming experiments to identify the optimum position of the detector for maximum radiation efficiency. In order to have a fast and reliable simulation result, beside the adequate physical description of all the processes involved in the X-ray emission, programming aspects play an important role. From the programming point of view, this code most sophisticated demand was to implement a recursive tree in FORTRAN90 in order to store the secondary electrons generated from the interaction of the primary electron (and the 'children' of the secondary electrons, as may be the case). To this end, we have used pointer, readily available in F90. Other techniques were list search and interpolation or finding a reliable random number generator (described in detail in this article). After organizing the code in modules, the task of designing the code and writing it is straightforward to achieve.

5 Running the software

In this section, we describe how to prepare the files that contain the physical data of the material used in the generation of X-rays. We also describe the input file where the injection energy of the electrons is given, together with the position of the detector and a few other settings relevant to the calculation. We have tailored the simulation code such that the minimum effort is required from the user. Finally, we describe the content of the output files and how to read the data they contain.

5.1 Preparing the files describing the material

In order for the code to run, we need to provide the files that contain the physical parameters that describe the material used for the generation of X-ray. The data is found in the folder ATOMIC DATA.

5.1.1 ElementData.dat

This file is self-explanatory. The user should input the atomic number of the element, the atomic weight, the density of the material, the Bethe parameter, and the binding energies of the K and L shells.

5.1.2 CharRadData.dat

This file contains fluorescence yields, Coster-Kronig transition probabilities, and radiative de-excitation probabilities for K and L shells. This file is the output of our adaptation of the code Xraylib of Schoonjans *et al.* [24]. The user has to change only the atomic number in the driver program of Xraylib library.

5.1.3 DCS_EL.dat and TOTAL_CS.dat

These files contain tabulated data with the elastic cross sections (the DCS and TCS). They are obtained by using the code of F. Salvat *et al.* [9]. We adapted the original code by writing an interface that is able to generate in a straightforward manner the two data tables mentioned above. Again, the procedure is almost entirely automatic (the user needs to change only the atomic number of the element and the range of the injection energy of the electron).

5.1.4 BSS.dat and BSS_interp.dat

This file contains the bremsstrahlung energy spectra that can be taken directly from [25]. These data are unfortunately not covering too many points in energy of the emitted photon as it would be desirable. We actually provide a driver program that is interpolates the data of Selter and Berger in order to obtain a denser-data table (given as output in UNFORMATTED FORTRAN 90 form to occupy less storage memory).

5.1.5 XrayAbs.dat

This file contains mass absorption coefficients for the X-ray radiation in the given medium from NIST database. These files are stored in folders for each atomic element, similar to a data library. They are used as demanded by experimental needs. They need to be calculated only once; for the rest, the user just changes the electron injection energy and afterwards the X-ray emission spectrum, depth dose absorption is then calculated.

5.2 Input and output files

The file in the main directory called simulation.dat contains the variables that are important for the virtual experiment. For example, the user can input here the injection kinetic energy of the electrons, the thickness of the sample, the parameters of the histograms, the number of electrons that are injected into the sample, or the position of the detector. The folder ATOMIC DATA contains the tabulated values for various processes, and the material parameters as described above. The code is now ready to run. The output files contain the X-ray spectrum from the chosen position of the detector and the depth-dose distribution inside the material. The results are to be found in the folder RESULTS, while the names of the files are self-explanatory which contain the name of targeted material, the energy of incident electron and the angle of the detector.

6 Results

The X-ray emission of different selected materials (Al, W, and Cu) has been compared with experimental data from the literature as well as with the simulation from the state-of-the-art PENELOPE general purpose package. Fig. 3 shows our first comparison: copper bombarded by 30 keV electrons. The experimental data is digitized from Fig. 6 of [26]. The results agree very well. The X-ray energy spreads from 0.5 keV (*i.e.*, the value of the electron absorption threshold used in our simulation), to the maximum energy equal to the kinetic energy of the incident electron.

The left panel of Fig. 4 shows the emitted X-ray spectra for lower incident–electron energies in the case of tungsten. We chose tungsten for the large number of characteristic peaks that match reasonably to our simulated data (the experimental data is digitized from Fig. 5c of [27]). The right panel in Fig. 4 shows the spectrum generated by bombarding gold (experimental data is digitized from Fig. 7 of [26]).

Fig. 5 shows the emitted photon spectrum in copper and displays the error bars (3σ intervals) of the simulated data. The simulated spectrum agrees well with the experimental with remarkable small error bars. As described in the previous section, when bombarding a metal target with an electron beam of given incident energy, X-ray can be emitted in any angle depending on target material and electron energy.

Fig. 6 shows the angular distribution of X-ray emitted when bombarding Cu target and Al with 20 keV electron beam. The solid curve and dotted curve represents the result

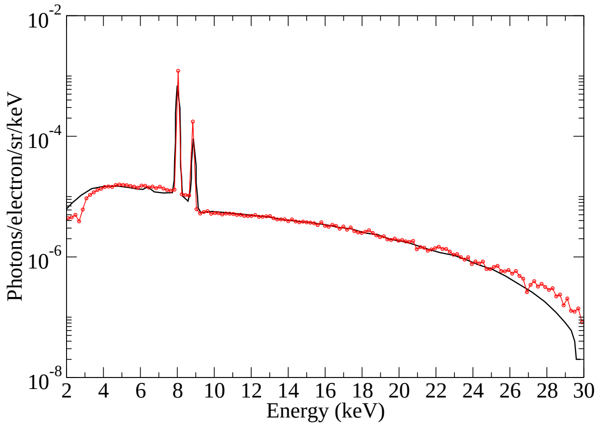

Figure 3: Simulated spectrum of Cu sample bombarded by 30 keV electrons at normal incidence. The black curve represents the experimental results (from [26], in arbitrary units scaled down by $1/50$), and the circles (red curve) represent our results. The take-off angle of the detector is 30°.

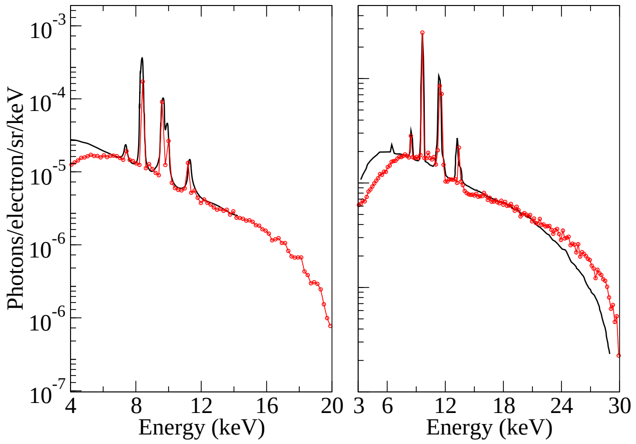

Figure 4: Left panel: simulated spectrum of W sample bombarded by 20 keV electrons at normal incidence. The black curve represents the experimental results (arbitrary units, scaled up by 10^3), and the circles (red curve) represent our results. The take-off angle of the detector is 40°. Right panel: simulated spectrum of Au sample bombarded by 30 keV electrons at normal incidence. The black curve: experimental results (scaled, see text). The circles (red curve): our results. The take-off angle of the detector is 30°.

obtained by our software and PENELOPE software respectively. As can be seen that the two distributions match quite well and most of the X-rays are emitted backward at the angle of about 110 degrees (measured from the incident direction of the bombarding electron).

In conclusion, we have presented in a coherent manner all the pieces that are nec-

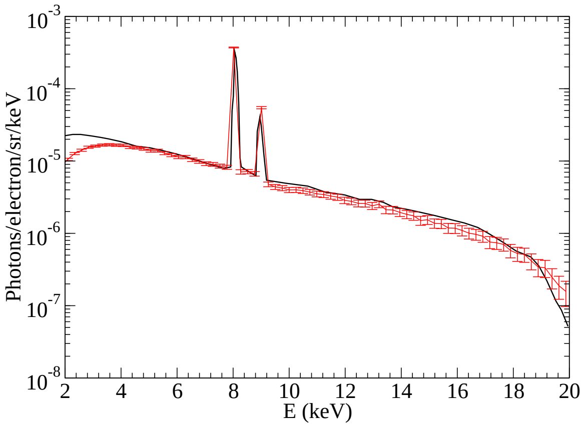

Figure 5: Simulated spectrum of Cu sample bombarded by 20 keV electrons at normal incidence (red curve with error bars). The black curve shows the experimental results from Fig. 4 in [28] (digitized from the original article, convolved with the response function of a SiLi detector). The detector take-off angle is centered at 140°.

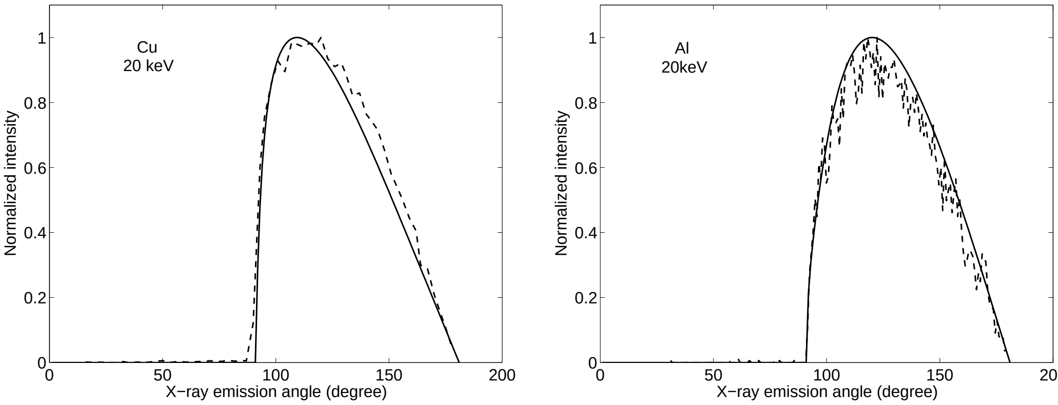

Figure 6: Angular distribution of X-ray emitted when bombarding Cu target (left) and Al target (right) with 20 keV normal electron beam. The solid line and dotted line show the results obtained from our software and PENELOPE software respectively.

essary to build a reliable X-ray spectrum simulator. We have developed a simple, fast computer code based on the Monte Carlo method for the simulation of X-ray spectra, in good agreement to the experimental data and other available software. For 10^5 injected electrons at 20keV bombarding Cu target, the execution time takes about 1 minute on an Intel Xeon X5650 processor (2.67 GHz) with a very good statistical errors. The software is designed in a modular fashion so that it can be easily integrated and extent with more sophisticated cross section database and more complicated X-ray source configuration.

References

[1] R. Gauvin, E. Lifshin, H. Demers, P. Horny and H. Campbell, *Microsc. Microanal.* **12**, 49 (2006).

[2] G. Poludniowski, *et al.*, *Phys. Med. Biol.* **54**, N433 (2009) R. Gauvin, E. Lifshin, H. Demers, P. Horny and H. Campbell, *Microsc. Microanal.* **12**, 49 (2006).

[3] N. W. M. Ritchie, *Microscopy and Microanalysis* **15**, 454 (2009); *ibid.* **15**, 512 (2009); *ibid.* **16**, 248 (2010).

[4] http://geant4.cern.ch/

[5] https://www.oecd-nea.org/tools/abstract/detail/nea-1525 ;
http://www.oecd-nea.org/dbprog/courses/nsc-doc2011-5.pdf

[6] https://mcnpx.lanl.gov/

[7] S. Kirckpatrick and E. P. Stoll, *J. Comput. Phys.* **40**, 517 (1981).

[8] F. Salvat *et al.*, *Radiation and Environmental Biophysics* **38**, 15 (1999).

[9] F. Salvat, A. Jablonski and C. J. Powell, *Computer Physics Communications* **165**, 157 (2005).

[10] E. Benedito, J. M. Fernández-Varea and F. Salvat, *Nuclear Instruments and Methods in Physics Research B* **174**, 91 (2001).

[11] F. Salvat, J. D. Martinez, R. Mayol and J. Parellada, *Computer Physics Communications* **42**, 93 (1986).

[12] A. Jablonski, F. Salvat, and C. J. Powell, *Journal of Applied Physics* **106**, 053706 (2009).

[13] D. Bote, F. Salvat, A. Jablonski, and C. J. Powell, *Atomic Data and Nuclear Data Tables* **95**, 871 (2009).

[14] D. Bote, F. Salvat, A. Jablonski, and C. J. Powell, ERRATUM *Atomic Data and Nuclear Data Tables* **97**, 186 (2011).

[15] M. Guerra, F. Parente, P. Indelicato and J. P. Santos, *International Journal of Mass Spectroscopy* **313**, 1 (2012);

[16] Private communication with Mauro Guerra.

[17] M. Gryzinski, Phys. Rev. **138**, A305 (1965); Phys. Rev. **138**, A322 (1965); Phys. Rev. **138**, A336 (1965).

[18] G. Weber, R. Märtin, A. Surzhykov, M. Yasuda, V. A. Yerokhin, Th. Stöhlker, *Nuclear Instruments and Methods in Physics Research B* **279**, 155 (2012).

[19] T. Schoonjans, *et al.*, *Spectrochimica Acta Part B* **70**, 10 (2012).

[20] E. Acosta, X. Llovet, E. Coleoni, J. A. Riveros and F. Salvat, *Journal of Applied Physics*, **83**, 6038 (1998).

[21] S. M. Seltzer and M. J. Berger, *Atomic Data and Nuclear Data Tables* **35**, 345 (1986).

[22] D. E. Cullen, *Nuclear Instruments and Methods in Physics Research B* **101**, 499 (1995).

[23] http://physics.nist.gov/PhysRefData/XrayMassCoef/tab3.html

[24] Tom Schoonjans, *et al.*, *Spectrochimica Acta Part B* **66**, 776 (2011

[25] S. M. Seltzer and M. J. Berger, *Atomic Data and Nuclear Tables* **35**, 345 (1986).

[26] D. Roet, P. Van Espen, *Nucl. Instr. and Methods in Physics Research B* **268**, 2794 (2010).

[27] X. Llovet, L. Sorbier, C.S. Campos, E. Acosta, and F. Salvat, *Journal of Applied Physics* **93**, 3844 (2003).

[28] F. Salvat, *et al.*, *Microchim. Acta* **155**, 67 (2006).