# Analytical and interpolated cross sections into the Monte Carlo code

Erminia Leonardi
12/11/1999

## Introduction

The Monte Carlo code (MC) developed by Prof. Carlo Rubbia makes use of a cross section table for each foreseen temperature. This implies either a coarse approximation of the evolution or a large memory buffer to contain all cases.

Therefore, it is important to search for the possibility of performing a local smearing starting from a low temperature during the determination of the cross sections at a higher temperature.

In this context, an algorithm has been implemented into the MC code, which calculates analytically the cross sections of the isotopes at a given temperature $T$, starting from those loaded from one of the available databases at a lower temperature $T_0$. Finally, it is illustrated the possibility of using an interpolation procedure which allows the calculation of the cross sections valid for a wide range of temperatures.

## Physics background

The content of this section is taken from the references [1,2,3] and is recalled here for convenience.

If we have a monoenergetic beam of neutrons impinging on an atom, the reaction rate will be defined as $R(V)$, where $V$ is the speed of the incoming neutron. By definition, the cross section is the ratio of the reaction rate and the flux $\sigma(V) = \frac{R(V)}{\phi(V)}$. When the flux is a single neutron per cubic meter, the flux reduces to $V$, and we can write $R(V) = V\sigma(V)$. In reality, even if the flux is monoenergetic, at any temperature above $T_0$=0 K. the atoms vibrate due to the thermal agitation, and the beam, as seen from the atoms, is not monoenergetic. Thus the reaction rate is also a function of the temperature. Now, if we define with $V_r$ the relative speed of the neutron with respect to the atom which moves at speed $V_T$, the reaction rate can be written as :

$$R(V) = \int_{All\ \vec{V}_T:V_r\geq 0} d\vec{V}_T R(V_r, T_0) P(\vec{V}_T) \tag{1}$$

If we assume isotropy we have the following expression of the probability distribution of the atoms speed:

$$P(\vec{V}_T) = \frac{1}{4\pi}P(V_T)dV_T d\mu d\phi = \frac{1}{2}P(V_T)dV_T d\mu, \tag{2}$$

and the integral becomes:

$$R(V,T) = \int_{-1}^{1} d\mu \int_{All \;\; V_T : V_r \geq 0} R(V_r, T_0)P(V_T)dV_T \tag{3}$$

The connection between the three variables $V, V_T, V_r$ is defined as:

$$V_r^2 = V^2 + V_T^2 - 2\mu V V_T \tag{4}$$

We can express $d\mu$ as :

$$d\mu = \mid \frac{d\mu}{dV_r} \mid dV_r = \frac{V_r}{V V_T} dV_r, \tag{5}$$

so that:

$$R(V,T) = \frac{1}{V}\int_0^{\infty} dV_r V_r R(V_r, T_0) \int_{|V-V_r|}^{V+V_r} \frac{1}{V_T} P(V_T) dV_T. \tag{6}$$

The explicit form of the probability distribution is obtained assuming the standard Maxwell distributions of speeds:

$$P(V_T, T) = \frac{4}{\sqrt{\pi}} \beta^{3/2} V_T^2 e^{-\beta V_T^2} \tag{7}$$

where $\beta = \frac{M}{2k(T-T_0)}$, and $M$ is the mass of the nucleus at rest. The integral can be performed analytically with the following result:

$$R(V,T) = \frac{1}{V}\sqrt{\frac{\beta}{\pi}} \int_0^{\infty} dV_r V_r R(V_r, T_0) \left[ e^{-\beta(V_r - V)^2} - e^{-\beta(V_r + V)^2} \right] \tag{8}$$

From this last expression we can easily obtain the form of the cross section:

$$\sigma(V,T) = \frac{1}{V^2}\sqrt{\frac{\beta}{\pi}} \int_0^{\infty} dV_r V_r^2 \sigma(V_r, T_0) \left[ e^{-\beta(V_r - V)^2} - e^{-\beta(V_r + V)^2} \right], \tag{9}$$

which is ultimately the function to integrate.

**The routine FASTCS**

Efficient strategies to calculate the eq.9 with a strong gain in computational effort are indicated by Rubbia (CERN/ET/Internal Note 98-06) and recalled here for convenience and sake of clearness.

The smeared cross section of the eq.9 can be written as an integral over the elementary cross section $\sigma(x, T=0)$:

$$\sigma(V,T) = \sigma^*(x,y) - \sigma^*(x,-y) \tag{10}$$

where,

$$\begin{aligned} x^2 &= \beta V_r^2, \quad y^2 \; = \; \beta V^2, \\ \sigma^*(x,y) &= \frac{1}{y^2}\frac{1}{\sqrt{\pi}}\int_0^\infty x^2\sigma(x,T_0)e^{-(x-y)^2}dx, \end{aligned} \tag{11}$$

Since our cross sections are a linearly interpolable table of $(\sigma_k, E_k)$ we can write:

$$\begin{aligned} \sigma^*(x,y) &= \frac{1}{y^2}\frac{1}{\sqrt{\pi}}[\int_0^{x_1} x^2\sigma(x,T_0)e^{-(x-y)^2}dx + \\ &\sum_{k=1}^{n-1}\int_{x_k}^{x_{k+1}} x^2(A_k + C_k x^2)e^{-(x-y)^2}dx + \int_{x_n}^\infty x^2\sigma(x,T_0)e^{-(x-y)^2}dx] \end{aligned} \tag{12}$$

If the above formula is taken literally, for each point $y$, a sum over all the other points has to be performed, which is a very time consuming procedure when the number of points is of several hundred thousands. But, because of the exponential terms in the integral, the contribution of the points which are far from $y$ is drastically reduced, so that an important approximation can be done: since $y$ is fixed by the input requirement, the integral on $x$ can be replaced with an integral on $z = x - y$ over the interval $|z| \leq 4$ (refer to Carminati's note CERN/ET/Internal Note 96-020 for a justification of this assumption).

The algorithm implemented in the routine FASTCS to calculate the eq.9 takes into account the fact that the table $(\sigma_k, E_k)$ is ordered in increasing energy. Thus, instead of evaluating $z$ through the chain $V_r \to x \to z$ and retaining only the $z$ which satisfy the condition $|z| \leq 4$, for each value $y$, the integral is reduced to a sum over only the points in the interval y$\pm$4.

The contribution of $\sigma^*(x,-y)$ is important only for the points corresponding to the beginning of the table, being $z = x + y$ generally $\gg$4 elsewhere; thus, this part of the eq.10 is computed only few times.

Therefore, the cross section $\sigma^*(x,y)$ is reduced to a sum over the surviving bins of the table $(\sigma_k, E_k)$ for which $|z| \leq 4$:

$$\sigma^*(x,y) = \frac{1}{y^2}\frac{1}{\sqrt{\pi}}\sum_{k=kmin}^{k=kmax} I_k \tag{13}$$

with

$$\frac{1}{y^2}I_k = \frac{\sqrt{\pi}}{2}\left[\frac{1}{y^2}C_k H^{(4)} + 4\frac{1}{y}C_k H^{(3)} + \left(\frac{1}{y^2}A_k + 6C_k\right)H^{(2)} + \left(2\frac{1}{y}A_k + 4C_k y^2\right)H^{(1)} + \left(A_k + C_k y^2\right)H^{(0)}\right]$$

where,

$$\begin{aligned}
A_k &= \frac{\sigma_k E_{k+1} - \sigma_{k+1} E_k}{E_{k+1} - E_k} & C_k &= \frac{m}{2\beta}\frac{\sigma_{k+1} - \sigma_k}{E_{k+1} - E_k} \\
H^{(n)}(z_{k+1}, z_k) &= F^{(n)}(z_{k+1}) - F^{(n)}(z_k) & & \\
F^{(n)}(z) &= \frac{2}{\sqrt{\pi}}\int_0^t t^n e^{-t^2} dt & & \\
F^{(0)}(z) &= erf(z) & F^{(1)}(z) &= \frac{1}{\sqrt{\pi}}\left(1 - e^{-z^2}\right) \\
F^{(2)}(z) &= \frac{1}{2}erf(z) - \frac{z}{\sqrt{\pi}}e^{-z^2} & F^{(3)}(z) &= \frac{1}{\sqrt{\pi}}\left[1 - \left(1 + z^2\right)e^{-z^2}\right] \\
F^{(4)}(z) &= \frac{3}{4}erf(z) - \frac{z}{\sqrt{\pi}}\left(\frac{3}{4} + z^2\right)e^{-z^2} & erf(z) &= \frac{2}{\sqrt{\pi}}\int_0^z e^{-t^2} dt
\end{aligned} \tag{14}$$

and $m$ is the mass of the neutron.

The approximations discussed above have been implemented in the routine FASTCS. The coefficients $A_k$ and $\beta C_k$ are computed only once in FASTCS, when the cross sections are loaded, and stored in arrays. The $x$ variable is also calculated once during the loading of the tables ($\mathrm{E}_k, \sigma_k$). Finally, the values of $F$ are tabulated (5 look up tables) and extracted at each time for the relevant values of the cross sections table.

The FASTCS routine has been implemented in the MC code and it works in the following way: for each elements and reaction loaded and stored in the array XSTORE at the temperature $T_0$, the routine FASTCS is called to compute the cross sections corresponding to a given higher temperature $T$, using the cross sections of XSTORE, and the smeared cross sections are stored into a new array, XSTORE1. The tabulated functions $F$ are computed once, in the routine FTAB, called by the main program, and passed to FASTCS via common.

An array, called IMTABL1, which represents the table of the pointers to the cross section informations, is continuously updated as long as XSTORE1 is filled. XSTORE1 contains only energies/cross sections at the temperature $T$, while XSTORE contains energies/cross sections at the temperature $T_0$ and neutrons energy distribution. Thus, during the MC simulation, the neutrons energy distribution are taken from XSTORE, while the energies and cross sections are taken from XSTORE1.

**The routine FASTCSINT**

The possibility of evaluating smeared cross sections by means of interpolation has been explored because it could represent a very fast procedure. We have looked for a formula of the following form:

$$\sigma(V,T) \quad = \quad \sum_{i=1}^{N} c_i \sigma(V,T_i), \tag{15}$$

where $\sigma(V,T_i)$ is the cross section at the temperature $T_i$, $c_i$ is a function of the various $T_i$ and $T$, and $T_1 < T < T_N$.

To this purpose, we recall the Lagrange's classical formula, for which the interpolating polynomial of degree N-1 through the N points $y_1 = f(x_1), y_2 = f(x_2), ...y_N = f(x_N)$ of the generic function $f(x)$ is:

$$P(x) = \frac{(x-x_2)(x-x_3)....(x-x_N)}{(x_1-x_2)(x_1-x_3)....(x_1-x_N)} y_1 + \frac{(x-x_1)(x-x_3)....(x-x_N)}{(x_2-x_1)(x_2-x_3)....(x_2-x_N)} y_2$$

$$+..... + \frac{(x-x_1)(x-x_3)....(x-x_{N-1})}{(x_N-x_1)(x_N-x_3)....(x_N-x_{N-1})} y_N \tag{16}$$

Thus, the interpolation of the cross sections $\sigma(V,T)$ with polynomials of order N-1 in $T$ gives:

$$\sigma_{interp.}(V,T) \quad = \quad \sum_{i}^{N} c_i \sigma_{analyt.}(V,T_i), \tag{17}$$

with

$$c_i \quad = \quad \prod_{j \neq i}^{N} \frac{T-T_j}{T_i-T_j}, \tag{18}$$

The routine FASTCSINT is a slightly different version of FASTCS, and it was also implemented and tested in the MC code. In this case, we have to choose a set of N temperatures and calculate the N coefficients $c$ of the eq.18 (this is done in the main program). The cross sections corresponding to the N temperatures are computed analytically with FASTCSINT, whose algorithm is the same of FASTCS, except that, at the end, the cross sections stored in the array XSTORE1 are obtained using the formula of the eq.17.

Of course, in this case we were only interested to the comparison of the cross sections with those obtained with FASTCS, and not to the CPU time needed to evaluate the cross sections.

This comparison has been done for all the elements contained in the database, and the results are discussed below. The applicability and the advantages of this method must be seen in a different context; for example, if we have a system of materials at different temperatures, and these temperatures can change during a simulation.

The application of this technique needs some compromises. First of all we must define the range of temperatures inside of which we want to calculate the cross sections with the interpolation. Then, we should select a restricted number of isotopes and reactions, the most frequently used during a MC simulation, and keep in memory the cross sections needed for the interpolation of each of them. The evaluation of the cross sections outside the defined range of temperature, or for all the other isotopes and reactions could be done analytically with FASTCS.

**Discussion of the results**

The cross sections calculated with FASTCS have been compared with those loaded from the database "endf/jar" at $T$=1300 K. In more details, the comparison was done spanning all the 766 elements of the database. The cross sections at 1300 K have been computed with FASTCS starting from those both loaded at 973 K and 293 K.

A complete analysis over all the elements and reactions of the database was done, evaluating the relative errors at $T$=1300 K, defined as

$$\epsilon_i = \frac{\|\sigma_{from\ the\ database}(E_i, T) - \sigma_{analy.}(E_i, T)\|}{\sigma_{from\ the\ database}(E_i, T)} \times 100.$$

The results are shown in Tab.1, where we have indicated the percentage of $\epsilon_i$ inside the different ranges $\epsilon_i < 1\%, 1\% < \epsilon_i < 2\%, 2\% < \epsilon_i < 3\%, ...., \epsilon_i > 10\%$.

In the case of the cross sections computed starting from 973 K, the percentage of $\epsilon_i > 10\%$ is 0.05%, and the percentage of $\epsilon_i < 1\%$ is 92.46%.

The results obtained calculating the cross sections from the starting temperature of 293 K are slightly worse than those obtained starting from 973 K; in fact, in this case, the percentage of $\epsilon_i > 10\%$ is 1%, and the percentage of the $\epsilon_i < 1\%$ is 87.5%.

A correct interpretation of the distributions of the errors $\epsilon_i$ must take into account the fact that the cross sections of the database are generally tabulated on different grids of energies at the different temperatures. Now, the $\epsilon_i$ have been evaluated after a linear interpolation of the cross section computed with FASTCS on the grid of energies of those loaded from the database at 1300 K. This means that an approximation is introduced when we compare loaded and calculated cross sections, with the resulting $\epsilon_i$ magnified.

The Fig.1, which represents the inelastic cross section of H-1 at 1300 K, shows a typical example of this frequent situation. The cross sections loaded at 1300 K (line A), those calculated with FASTCS starting from both 973 K and 293 K (points B and C, respectively), and, finally, those obtained from their linear interpolation on the grid of energy of the loaded cross sections (points D and E, respectively) are simultaneously plotted. The cross section loaded from the database is 27.44 barns at 0.016 eV, while those calculated from 293 K and 973 K and then

Table 1: Distribution of the errors $\epsilon_i$. All the cross sections of the database "endf/jar" have been both loaded at $T_0$=973 K and $T_0$=293 K and calculated with FASTCS at 1300 K. N is the percentage of $\epsilon_i$ inside the different ranges indicated in the first column.

| range | N,% ($T_0$ = 973 K) | N,% ($T_0$ = 293 K) |
|---|---|---|
| $\epsilon_i < 1\%$ | 92.46 | 87.5 |
| $1 < \epsilon_i < 2\%$ | 6.97 | 7.69 |
| $2 < \epsilon_i < 3\%$ | 0.3 | 1.74 |
| $3 < \epsilon_i < 4\%$ | 0.09 | 0.79 |
| $4 < \epsilon_i < 5\%$ | 0.05 | 0.44 |
| $5 < \epsilon_i < 6\%$ | 0.02 | 0.27 |
| $6 < \epsilon_i < 7\%$ | 0.01 | 0.18 |
| $7 < \epsilon_i < 8\%$ | 0.008 | 0.13 |
| $8 < \epsilon_i < 9\%$ | 0.006 | 0.10 |
| $9 < \epsilon_i < 10\%$ | 0.004 | 0.10 |
| $\epsilon_i > 10\%$ | 0.05 | 1.00 |

interpolated at 0.016 eV are 28.26 eV ($\epsilon_i = 3\%$) and 27.891 eV ($\epsilon_i = 1.64\%$), respectively. But, the cross section calculated from 973 K has been interpolated between its values at 0.127 eV and 0.202 eV, while the cross section calculated from 293 K has been interpolated between 0.101 eV and 0.202 eV, which is a larger interval of energy.

A careful check of the cases with large $\epsilon_i$ has shown that strong deviations ($\epsilon_i > 10\%$) are always around regions where the cross section undergoes sudden jumps. At T=1300$^o$, the whole table of the cross sections loaded from the database presents 3260 points with duplicated energies (which corresponds to the 0.075% of the total number of points): in the regions around these points, which are, on the other hand, without physical meaning, the cross sections are strongly discontinuous. Even if the duplicated energies are removed using the algorithm implemented by Carminati in the routine XNDUPL (it checks if there are two points with the same energy and a different cross section, and, if this is the case, moves the second point by a relative amount of $10^{-5}$ eV), the comparison of the cross sections still gives large $\epsilon_i$, as shown in Fig.2 for the total cross section of Kr-82 at $T$=1300 K, where both the loaded cross sections and those calculated with FASTCS starting from 293 K and 973 K are plotted. In this case, $\epsilon_i$ comes up to $\sim$ 40% at 400 eV.

Another frequent circumstance guilty of large $\epsilon_i$ is when the cross section loaded from the database is zero in some regions. Although these cases have been skipped in the calculation of the $\epsilon_i$, in the close points the loaded cross section abruptly fall down to zero, while the cross section evaluated analytically never becomes exactly zero and decreases and increases smother in the same region. A typical example of this behavior is shown in Fig.3, for the capture cross section of Fe-58 at $T$=1300 K, where the cross section calculated with FASTCS differ of 800%

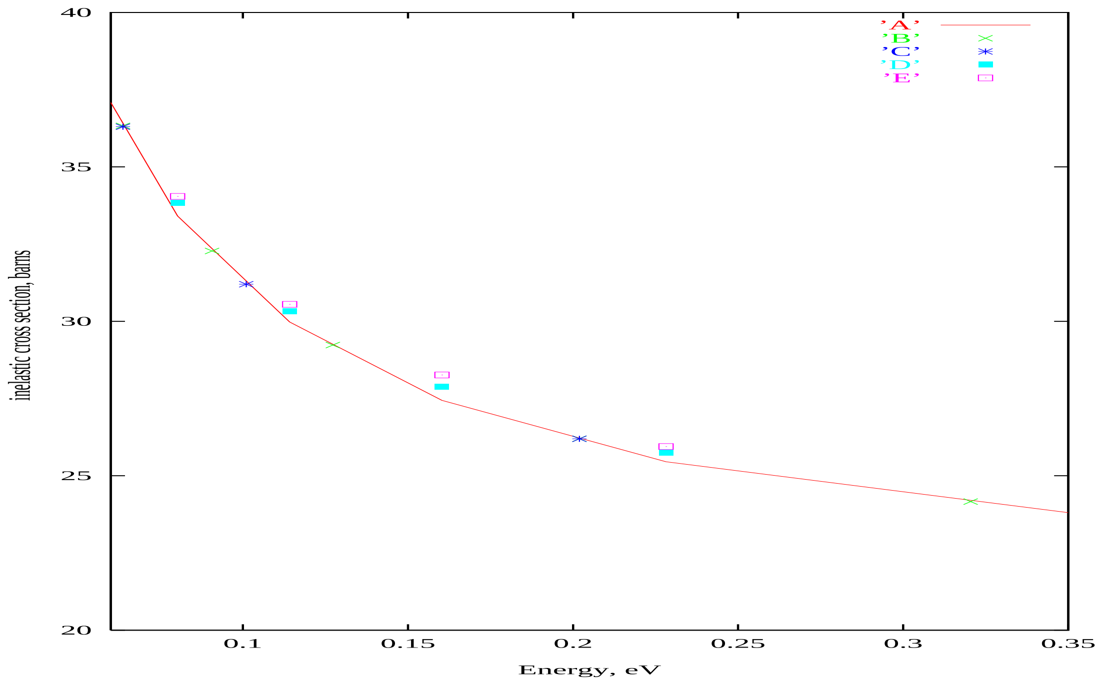


Figure 1: Inelastic cross section of H-1 at $T$=1300 K.
Line A is the cross sections loaded from the database.
Point B is the cross section computed with FASTCS starting from 973 K.
Point C is the cross section computed with FASTCS starting from 293 K.
Point D is the cross section computed with FASTCS starting from 973 K and interpolated on the grid of energy of the cross section loaded from the database.
Point E is the cross section computed with FASTCS starting from 293 K and interpolated on the grid of energy of the cross section loaded from the database.

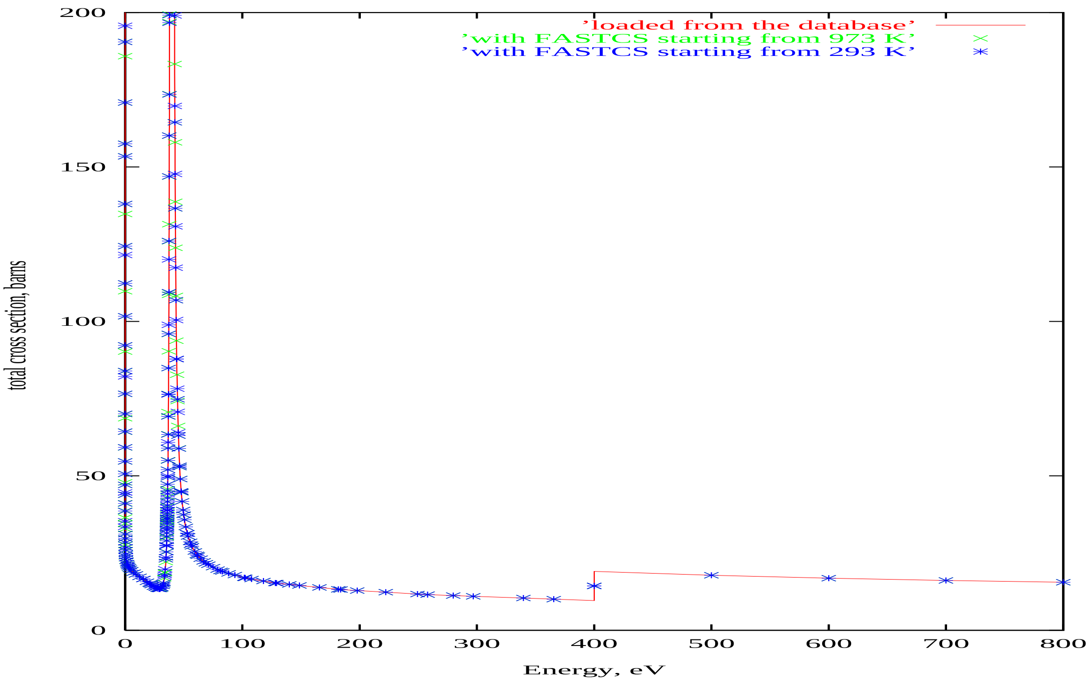


Figure 2: Total cross section of Kr-82 at $T$=1300 K.

at 8.4 eV with respect to the loaded one!

Some large $\epsilon_i$ ($\epsilon_i > 10\%$) is also found when the cross section is very small ($< 10^{-2}$), but in these cases the difference between the cross sections are, of course, also very small.

Finally, we have tested the validity of the interpolation method. Again, the check was done over all the elements of the database. In this context, we have chosen the following four temperatures to interpolate the cross sections at 1300 K: $T$= 1150, 1250, 1350, and 1450 K (set $A$). The cross sections have been loaded both at 293 K and 973 K, computed analytically at these four temperatures, and interpolated at 1300 K, using the eq.17. The results are in excellent agreement with those obtained with the analytical calculation, being the distribution of the $\epsilon_i$ perfectly reproduced, and the relative error, $100 \times \frac{\|\sigma_{analy.}(E_i,T)-\sigma_{interp.}(E_i,T)\|}{\sigma_{analy.}(E_i,T)}$, always well below 1%.

The interpolation was also tested with another larger set of temperatures: $T$= 700, 900, 1200, 1400, and 1600 K (set $B$). The cross sections have been loaded from the database at 239 K, calculated analytically at the five temperatures indicated above, and interpolated at 1300 K. Again, the distribution of the errors, $\epsilon_i$, does not change with respect to that obtained with the analytical calculation starting from 293 K.

Finally, it was calculated the CPU time needed to load all the elements of the database and to evaluate analytically or by interpolation the cross sections, both starting from 293 K and 973 K. The CONVEX V-CLASS parallel machine was used, and the results are shown in Tab.2.

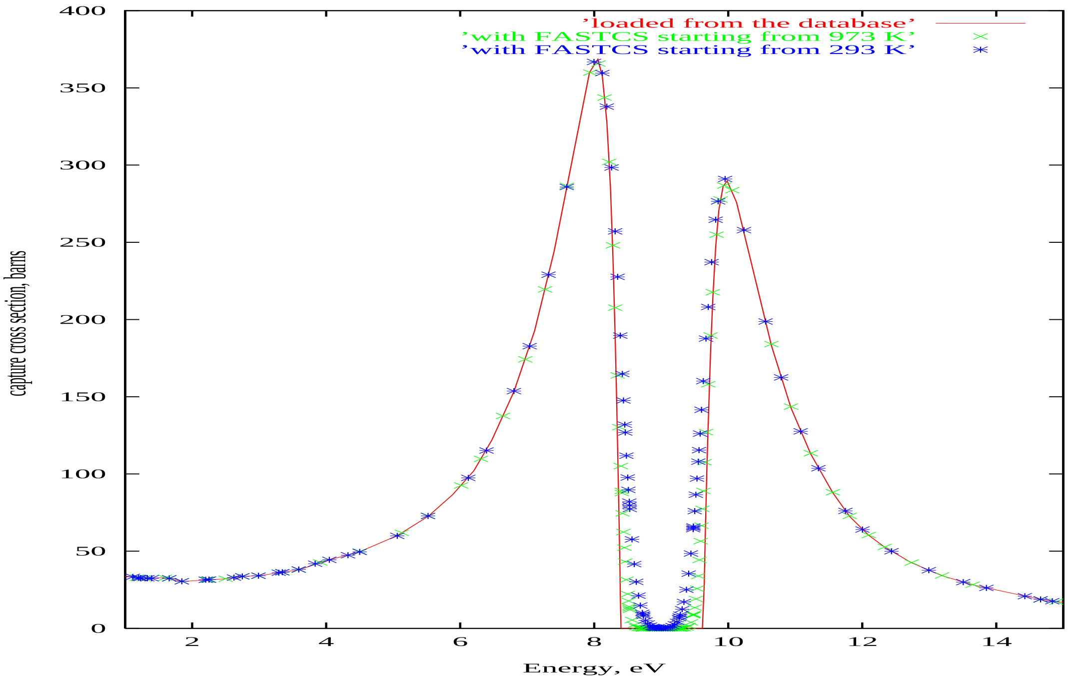


Figure 3: Capture cross section of Fe-58 at $T$=1300 K.

Table 2: CPU time needed for the evaluation of all the cross sections of the database "endf/jar" at $T$=1300 K on the CONVEX V-CLASS parallel machine.

| | CPU time, s |
|---|---|
| with FASTCS, starting from 973 K | 368 |
| with FASTCS, starting from 293 K | 421 |
| with FASTCSINT and the set $A$, starting from 973 K | 512 |
| with FASTCSINT and the set $A$, starting from 293 K | 708 |
| with FASTCSINT and the set $B$, starting from 293 K | 756 |

## Conclusions

In this work we have presented the results obtained, spanning all the elements of the database "endf/jar", and comparing, at 1300 K, the loaded cross sections with those calculated analytically with the routine FASTCS. Cross sections interpolated at 1300 K over different grids of temperatures have also been compared with the loaded ones. A perfect overlap was found between the cross sections evaluated analytically with FASTCS and those interpolated using FASTCSINT, even if the temperatures considered for the interpolation covers a large interval ($\Delta T = 900$ K for the case indicated above as set $B$).

Differences between loaded and calculated cross sections with $1\% < \epsilon_i < 10\%$ are "introduced" when the calculated cross sections are interpolated on the grid of energy of those loaded from the database, and thus, they cannot be considered relevant (see Fig.1). Instead, large differences ($\epsilon_i > 10\%$) are mostly concentrated around regions where the loaded cross sections do not behave correctly, because of sudden jumps (see Fig.2), or where the loaded cross sections are exactly zero (see Fig.3).

Out of these cases, the agreement between loaded and calculated cross sections is very good ($\epsilon_i < 1\%$), and an example is shown in Fig.4, for the capture cross section of U-238 at $T$=1300 K. In this case, the interpolation was performed with the temperatures of the set $B$. A zoom over a smaller range of energies evidences the overlap of the cross sections computed with different methods and those loaded from the database.

Finally, the CPU time needed to load all the cross sections of the database "endf/jar" and to perform their calculation with FASTCS or FASTCSINT at a higher temperature on a CONVEX V-CLASS parallel machine was reasonable, as shown in Tab.2.

## References


1. D.E. Cullen and C.R. Weisbin, *Exact Doppler Broadening of Tabulated Cross Sections*, Nuclear Science and Engineering, 60, 199-299, (1976).
2. D.E. Cullen, *Program SIGMA1 (version 79-1): Doppler broaden evaluated cross sections in the evaluated nuclear data file/version B (ENDF/B) format*, UCRL 50400, Volume 17.
3. F. Carminati, *XNBROA routine*, CERN/ET/Internal Note 96-020, (1996).

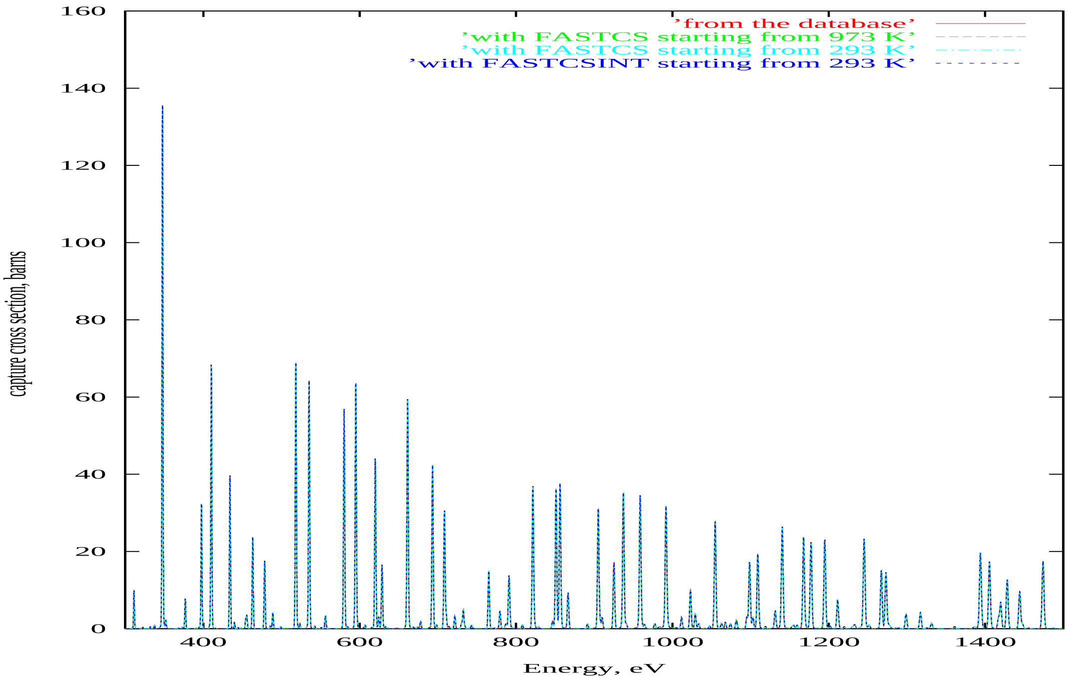


Figure 4: Capture cross section of U-238 at $T$=1300 K.

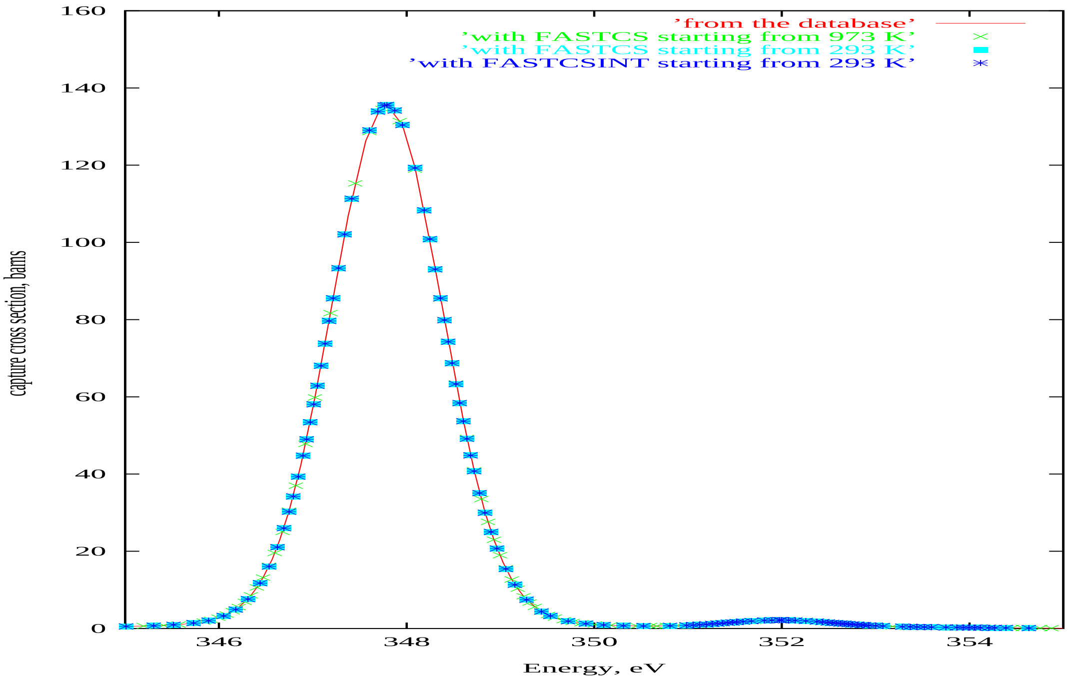


Figure 5: Capture cross section of U-238 at $T$=1300 K. A zoom over the region between 345 and 355 eV.