
Article:
Tallents, Gregory John orcid.org/0000-0002-1409-105X, Aslanyan, Valentin and Aslanyan, A. G. (2017) Efficient calculation of degenerate atomic rates by numerical quadrature on GPUs. Computer Physics Communications. ISSN 0010-4655

https://doi.org/10.1016/j.cpc.2017.06.003

Efficient calculation of degenerate atomic rates by numerical quadrature on GPUs

V. Aslanyan[a,*], A. G. Aslanyan, G. J. Tallents[a]

[a] *York Plasma Institute, University of York, York, Heslington, YO10 5DD, United Kingdom*

Abstract

The rates of atomic processes in cold, dense plasmas are governed strongly by effects of quantum degeneracy. The electrons follow Fermi-Dirac statistics and their high density limits the number of quantum states available for occupation after a collision. These factors preclude a direct solution to the usual rate coefficient integrals. We summarise the formulation of this problem and present a simple, but efficient method of evaluating collisional rate coefficients via direct numerical integration. Numerical quadrature has an intrinsically high level of parallelism, ideally suited for graphics processor units. GPUs are particularly suited to this problem because of the large number of integrals which must be carried out simultaneously for a given atomic model. A CUDA code to calculate the rates of significant atomic processes as part of a collisional-radiative model is presented and discussed. This approach may be readily extended to other applications where rapid and repeated evaluation of many integrals is required.

Keywords: Plasma, Fermi-Dirac, Quadrature, GPU
PACS: 52.25.Jm, 05.30.Fk, 52.50.Jm

1. Introduction

Fermi-Dirac statistics have long been used to describe the distribution of electrons in metallically bonded solids, but plasmas of comparable electron densities have only recently become experimentally accessible. Re-

*Principal corresponding author
Email address: va567@york.ac.uk (V. Aslanyan)

cent advances in short wavelength lasers have allowed the creation of warm dense plasmas at solid or greater densities, due to their correspondingly high critical densities. These advances have spurred investigations into the effects of degeneracy on the atomic rates and hence on macroscopic plasma properties[1, 2, 3].

Fermi-Dirac statistics complicate the calculation of quantities in dense plasmas because many of the integrals in the calculations of basic plasma properties and atomic rates, which are discussed below, do not have closed-form solutions. Collisional-radiative models typcially require > 100 rates to be assembled into the rate matrix and may in turn be evaluated at a large number of spatial or temporal points in line with a hydrodynamic solver. We therefore require a large degree of parallelism to make this problem tractable. The Compute Unified Device Architecture is a programming tool to enable large scale parallel computation on Nvidia GPUs, which are emerging as a computational asset for physicists.

In Section 2 we review the Fermi-Dirac distribution and its use in the integrals for atomic rates in degenerate plasmas. In Section 3 we discuss quadrature on GPUs, which may be of general interest. In Section 4 we present the results of a simple collisional-radiative model of a degenerate plasma, which is greatly sped up by carrying out integrals on a GPU. Example code for generic 1- and 2-dimensional quadrature on GPUs, as well as the collisional-radiative model are available for download[4].

2. Definition of atomic rate coefficients

2.1. The Fermi-Dirac distribution and chemical potential

For a given electron kinetic energy ϵ, density n_e and temperature T_e, the Fermi probability of occupation is given by[5]

$$F(\epsilon, T_e) = \frac{1}{\exp\left(\frac{\epsilon-\mu}{T_e}\right) + 1}, \tag{1}$$

where $\mu(T_e, n_e)$ is the chemical potential. It is convenient to use the familiar units of electronvolts (1 eV $= q_e$ J, where q_e is the elementary charge) for energies; temperatures are implicitly multiplied by the Boltzmann constant, $k_B = 8.617$ eV K^{-1}. The energy distribution function is given by

$$f_{FD}(\epsilon, T_e) = \frac{G}{n_e}\sqrt{\epsilon}F(\epsilon, T_e) \tag{2}$$

with $G = 4\pi \left(2m_e/h^2\right)^{3/2}$ the degeneracy of a free electron. The chemical potential is defined so as to normalise this distribution, but it is computationally convenient to use a direct formula; for example, see the Padé approximation given here[6]. A collision in a degenerate plasma may only occur if there are sufficient unoccupied quantum states for all the resultant fermions and we take account of this by using Pauli blocking factors,

$$\tilde{F}(\epsilon, T_e) = 1 - F(\epsilon, T_e). \tag{3}$$

Finally, the Fermi-Dirac distribution has a corresponding heat capacity defined through

$$C_V(T_e, n_e) = \frac{G}{n_e} \int_0^\infty \frac{\epsilon^{3/2}}{1 + \exp(\frac{\epsilon - \mu}{T})} d\epsilon. \tag{4}$$

The total electron kinetic energy density $\varepsilon = n_e C_V$ allows the electron temperature to be determined. We do this by inverting the heat capacity through a Padé approximation.

2.2. Definition of atomic rate coefficients

The rate coefficients of atomic or nuclear processes are typically given by an integral over the cross-section weighted by the distribution function, typically denoted $\langle \sigma v \rangle$. The usual approach to define such integrals is usually *ad hoc*; a more rigorous approach[7] is generalised for degenerate rates in Appendix A.

The total collisional excitation rate follows directly from the treatment in Appendix A, given by

$$J^{\uparrow}(E_j, T_e, \mu) = N_i G \sqrt{\frac{2}{m_e}} \int_{E_j}^\infty \Omega\left(\frac{\epsilon_0}{E_j}\right) F(\epsilon_0, T_e) \tilde{F}(\epsilon_0 - E_j, T_e) d\epsilon_0, \tag{5}$$

where E_j is the excitation energy and the collision strength is related to the total cross-section by $\Omega(\epsilon_0) = \sigma(\epsilon_0)/\epsilon_0$. We consider a collision strength typical of optically allowed transitions,

$$\Omega\left(\frac{\epsilon_0}{E_j}\right) = B_0 \ln\left(\frac{\epsilon_0}{E_j}\right) + \sum_{k=1}^{5} B_k \left(\frac{\epsilon_0}{E_j}\right)^{-(k-1)}. \tag{6}$$

The methods presented here can be extended to optically forbidden transitions straightforwardly.

The collisional ionization rate is given by

$$K^{\uparrow} = N_i G \sqrt{\frac{2}{m_e}} \int_{E_i}^{\infty} \int_0^{\epsilon_0 - E_i} \epsilon_0 \frac{d\sigma^{\uparrow}}{d\epsilon_1} F(\epsilon_0, T_e) \tilde{F}(\epsilon_1, T_e) \tilde{F}(\epsilon_0 - \epsilon_1 - E_i, T_e) d\epsilon_1 d\epsilon_0, \tag{7}$$

where E_i is the ionization potential and $\frac{d\sigma^{\uparrow}}{d\epsilon_1}$ is the differential cross-section. This differential cross-section gives the distribution of outgoing electrons and is necessary to account for both blocking factors in this case. The experimental uncertainty in differential cross-sections is typically much higher than that for the total collisional ionization cross-sections. We have proposed[3] a differential cross-section similar to that by Mott, modified to be consistent with the well-known BELI[8] formula,

$$\frac{d\sigma^{\uparrow}}{d\epsilon_1} = \frac{1}{2E_i\epsilon_0} \left[\frac{C_0 E_i}{(\epsilon_1 + a)(\epsilon_1 + b)} + \frac{C_0 E_i}{(\epsilon_0 - \epsilon_1 - E_i + a)(\epsilon_0 - \epsilon_1 - E_i + b)} + \sum_{k=1} k C_k \frac{\epsilon_1^{k-1} + (\epsilon_0 - \epsilon_1 - E_i)^{k-1}}{\epsilon_0^k} \right], \tag{8}$$

with the quantities

$$a = \frac{1}{2} \left(\sqrt{\epsilon_0^2 + 4E_i^2} - \epsilon_0 \right)$$
$$b = a + E_i.$$

In the calculations below, we consider for simplicity that the outer electrons are ionized preferentially, as the effect of Pauli blocking is severe for inner-shell electrons.

These atomic rates may be related to their inverse through simple algebraic formulas, which can be derived using the approach in Appendix A together with appropriate micro-reversibility relations[7], noting in particular that $\tilde{F}(\epsilon, T_e) = \exp\left((\epsilon - \mu)/T_e\right) F(\epsilon, T_e)$. In particular, we have for the rate of collisional deexcitation

$$J^{\downarrow} = \frac{g_j}{g_{j'}} \exp(E_j/T_e) J^{\uparrow}, \tag{9}$$

where the g factors correspond to the degeneracies of the upper and lower level respectively. For three body recombination, we have

$$K^{\downarrow} = \frac{g_i}{g_{i+1}} \exp(\mu/T_e) \exp\left(\frac{E_i}{T_e}\right) K^{\uparrow}. \tag{10}$$

The total rate of photoionization, for radiation with a photon of energy ϵ_γ and spectral intensity $I(\epsilon_\gamma)$, is given by

$$L^{\uparrow} = N_i \int_{E_i}^{\infty} \sigma_\gamma(\epsilon_\gamma) I(\epsilon_\gamma) \tilde{F}(\epsilon_\gamma - E_i, T_e) d\epsilon_\gamma. \tag{11}$$

The photoionization cross-section typically falls off above threshold as some negative power of the photon energy; for example, the cross-section for hydrogen-like ions scales as $\sigma(\epsilon_\gamma) \propto \epsilon_\gamma^{-3} Z^{-2}$. We parameterize the photoionization cross-section above threshold by

$$\sigma_\gamma = D_0 \left(\frac{\epsilon_\gamma}{E_i}\right)^{-2} + D_1 \left(\frac{\epsilon_\gamma}{E_i}\right)^{-3}. \tag{12}$$

Aside from including the Fermi-Dirac energy distribution, the effects of degeneracy do not alter the rate of radiative recombination from its classical form, because the outgoing photons are bosons and not subject to Pauli blocking. Furthermore, radiative recombination is likely to be insignificant compared to photoionization for the large radiation fluxes in high energy density physics experiments.

Radiative processes have an effect not only on the ionization of the plasma, but also the temperature. Photoionization also heats the plasma, because the remainder of the photon energy is carried away mostly by the ionized electron. The corresponding rate of change in the electron kinetic energy is

$$\left(\frac{d\varepsilon}{dt}\right)_L = N_i \int_{E_i}^{\infty} (\epsilon_\gamma - E_i) \sigma_\gamma(\epsilon_\gamma) I(\epsilon_\gamma) \tilde{F}(\epsilon_\gamma - E_i, T_e) d\epsilon_\gamma. \tag{13}$$

Photons are captured by free electrons in the presence of ions by inverse bremsstrahlung, with a corresponding rate of change of kinetic energy

$$\left(\frac{d\varepsilon}{dt}\right)_{\mathrm{IB}} = \int_{\epsilon_c}^{\infty} \int_0^{\infty} \kappa_{IB}(\epsilon_\gamma) I(\epsilon_\gamma) F(\epsilon_0, T_e) \tilde{F}(\epsilon_\gamma + \epsilon_0, T_e) d\epsilon_0 d\epsilon_\gamma, \tag{14}$$

where the inverse bremsstrahlung absorption coefficient is given by

$$\kappa_{IB} = \frac{1}{6\sqrt{6}\pi^{5/2}} \frac{e^6 (hc)^2}{\varepsilon_0^3 (m_e c^2)^{3/2}} \epsilon_\gamma^{-3} \left[1 - \exp\left(-\frac{\epsilon_\gamma}{T_e}\right)\right] \sum_{i=1}^{Z} i^2 N_{i,j}. \tag{15}$$

Radiation with frequencies below the resonant electron frequency is reflected and we therefore take the corresponding critical photon energy $\epsilon_c = \hbar c \times \sqrt{n_e e^2/m_e c^2 \varepsilon_0}$ as the lower limit for the inverse bremsstrahlung integral. The integral over the electron energy ϵ_0 can be carried out analytically, leaving the integral over the photon energy to be done numerically.

If the distibution of incoming photons f_γ is close to a delta function, as in a laser, the total rate of photoionization differs in a degenerate plasma only by a Pauli blocking factor, so that it is $\Gamma^\uparrow = N_i I \sigma_\gamma(E_{\rm Laser}) \tilde{F}(E_{\rm Laser} - E_i, T_e)/E_{\rm Laser}$. However, of particular interest is the case of incoming black body radiation where the photon distribution function is given by a Bose-Einstein distribution at temperature T_r with vanishing chemical potential,

$$I_{\rm BB}(\epsilon_\gamma, T_r) = \frac{8\pi\epsilon_\gamma^3}{(hc)^3} \frac{1}{\exp\left(\frac{\epsilon_\gamma}{T_r}\right) - 1}. \tag{16}$$

In summary, we require efficient solutions for Equation (5) for every transition between two atomic levels, for Equations (7), (11) and (13) for each atomic level, and once for Equation (14). In particular, collisional ionization requires a double integral as given in Equation (7). Inverse collisional rates are calculated by the algebraic detailed balance relations of Equations (9) and (10); we do not calculate the inverse radiative rates here.

2.3. Upper limits of integration

In order to numerically integrate the functions in the definition of atomic rates, the integrals which normally run to infinity must be truncated at some finite upper limit A. This limit must be selected sufficiently high to capture the main contribution to the integral, but low enough to exclude the region where the integrand has effectively vanished. The Fermi-Dirac distribution tends to zero faster than the cross-sections of atomic processes and therefore the upper limit is largely defined by it.

For each integrand H, we define the relative accuracy of a truncated integral by the parameter p,

$$\left| 1 - \frac{\int_E^A H(\epsilon) d\epsilon}{\int_E^\infty H(\epsilon) d\epsilon} \right| < 10^{-p}. \tag{17}$$

Simple formulae for the upper limits for a given accuracy parameter p for the integrals considered in §2.2 are given in Table 1, valid for accuracies

$5 \leq p \leq 15$ and for the range of parameters $0.1 < T_e < 30$ eV (above this the Maxwell-Boltzmann distribution can typically be used) and $n_e < 10^{24}$. Several quantities appear in the table, which depend on the desired accuracy and need only be calculated once.

Integral	**Upper limit**
Collisional excitation	$2E_j + 2\|\mu\| + T_e[7.5 + p\ln(10)]$
Collisional ionization	$E_i + \|\mu\| + 50 + T_e[13.74 + p\ln(10)]$
Photoionization	$\mu \leq 0$: $E_i + T_r x$ $\mu > 0$: $E_i + \mu + T_r x$
Inverse bremsstrahlung	$0.1 < T_e \leq 3$: $T_r/T_e[y_1 + z_1\mu/T_e]$ $3 < T_e \leq 30$: $T_r/T_e[y_2 + z_2\mu/T_e]$

Table 1: Upper limits of integration for the degenerate atomic rates considered in this article. For values of x, y, z, see text. The radiative rates assume a black body radiation spectrum at a temperature T_r.

The quantity x used for the photoionization integrals is the solution to the transcendental equation

$$4(1+x)\exp(-x) = 10^{-p}. \tag{18}$$

In particular, we have that $x(p=5) = 15.7$ and $x(p=6) = 18.2$. For the inverse bremsstrahlung integral, we have derived an upper limit given by

$$A = \frac{T_r}{T_e}\ln\left(\frac{\exp(u)}{\exp(u)-1}\right), \tag{19}$$

where the quantity

$$u = \frac{10^{-p}}{2}\ln\left(\frac{\left(1+\sqrt{1+\exp(\mu/T_e)}\right)^{T_e/T_r}}{\left(1+\sqrt{1+\exp(\mu/T_e)}\right)^{T_e/T_r}-1}\right). \tag{20}$$

Direct numerical evaluation of this formula requires several computationally expensive function calls, which significantly slows down the calculation of this rate. Instead, we choose for A the upper bound to Equation (19), valid over some region in parameter space, such that

$$A = \frac{T_r}{T_e}\left(y + \frac{\mu}{T_e}z\right) > \frac{T_r}{T_e}\ln\left(\frac{\exp(u)}{\exp(u)-1}\right). \tag{21}$$

For $p = 5$, we have for $0.1 < T_e \leq 3$ $y_1 = 11.97$, $z_1 = 0.0053961$ and for $3 < T_e \leq 30$ $y_2 = 11.66$, $z_2 = 0.066357$.

We present an example of the derivations of these upper limits in Appendix B in addition to strategies to widen the range of valid parameters. Calculating an upper limit is a bottleneck in the evaluation of the corresponding integral; therefore they should ideally involve only arithmetic operations. We find that an accuracy of 10^{-5} is sufficient to solve time-dependent rate equations or to invert the rate matrix to find the steady state ion populations, which are typical uses of atomic rate coefficients.

3. Quadrature

3.1. Nodes and weights in Gaussian quadrature

The general process of quadrature involves approximation of a definite integral by a discrete sum of the integrand evaluated at certain nodes x_j, weighted at each node by some coefficient w_j,

$$\int_a^b f(x)dx \approx \sum_{j=1}^{N} w_j f(x_j). \tag{22}$$

In order to approximate the definite integral to a given accuracy, there is a general tradeoff between the number of function evaluations and the complexity of the nodes and weights. The Newton-Cotes methods, for instance, take many simplified nodes to calculate an integral. It is therefore clear for this application that even a relatively large initial investment to compute the nodes and weights is prudent, as they will allow repeated subsequent integral evaluations. Gaussian quadrature formulae are for this reason most suited to the current problem.

Using the Golub-Welsch algorithm[9], the nodes and weights may be obtained respectively from the eigenvalues and eigenvectors of a symmetric tridiagonal Jacobi matrix. If an eigenvalue λ_j has a corresponding eigenvector $\mathbf{q}_j$, then the nodes and weights are given by[10]

$$x_j = \lambda_j \tag{23}$$
$$w_j = (\mathbf{q}[1]_j)^2, \tag{24}$$

where $\mathbf{q}[1]$ is the first element of the vector.

3.2. Choice of nodes and weights

We have found that the simplest Gauss-Legendre method is sufficient for the current problem. In this case, the diagonal α and off-diagonal β elements of the symmetric tridiagonal Jacobi matrix are

$$\alpha_i = 0 \tag{25}$$

$$\beta_i = \frac{1}{\sqrt{1-(2i)^2}}. \tag{26}$$

We require that the error in integration be at least equal to the truncation error discussed in §2.3 or better. In order to match this truncation error, we have undertaken numerical experiments for a wide range of the parameters T_e and n_e. For values of these parameters, we found the minimum number of nodes required to obtain a result with a relative error $< 10^{-p}$, as shown in Figure 1.

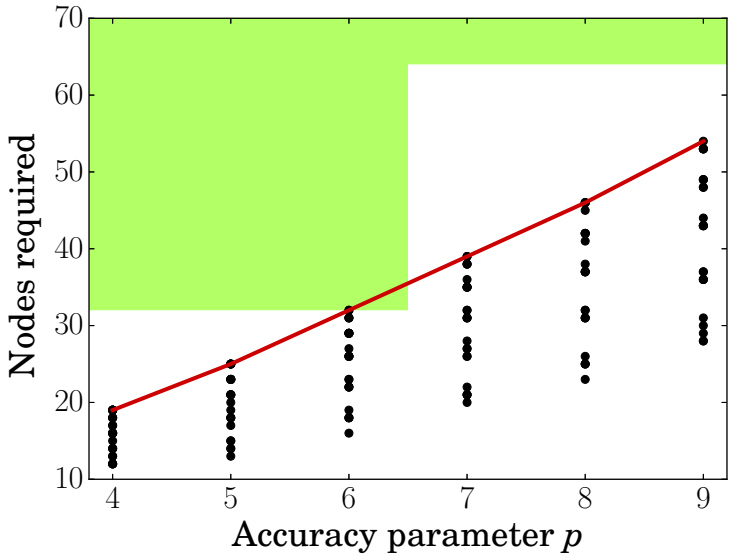

Figure 1: (color online). The number of nodes required to achieve a given value of the accuracy parameter p, required to ensure that the integrals are accurate to 10^{-p}, following the truncation formulas in §2.3. The circles correspond to different values of T_e and n_e, their maximum given by the red line. As CUDA warps typically execute 32 threads simultaneously, it is useful to use a multiple of 32 as the number of nodes for GPU implementation; this is indicated by the shaded region.

3.3. Implementation in CUDA

In order to evaluate the nodes and weights required for Gaussian quadrature, we have chosen to compute the eigenvalues and eigenvectors of the Jacobi matrix using the CPU LAPACKE routine `dstev`, which is optimised for this type of real, symmetric, tridiagonal matrix. A single set of nodes and weights are copied once to the GPU and used for every subsequent integral (once for both dimensions of the double integral). We typically choose $N = 32$ nodes, as this gives sufficient accuracy in accordance with the discussion in §3.2 and is the standard warp size; we provide tabulated values for nodes and weights for this case in Appendix C, if using `LAPACKE` is undesirable.

In a CUDA implementation of numerical quadrature, we therefore choose a block dimension of y warps ($32y$ threads per block), each thread in a warp evaluating the integrand at one node. The optimal choice of y depends on the type of GPU; the total time to evaluate 9600 $J^{\uparrow}$, $K^{\uparrow}$ and $L^{\uparrow}$ integrals on three GPUs described in Table 2 is given for different choices of y in Figure 2. We see that a choice of $y = 2$ or $y = 6$ minimizes the computation time.

Device	**Clock speed** [MHz]	**Driver Version**
GeForce GTX 580	772	7050
GeForce GTX 680	1058	8000
GeForce GTX TITAN Black	980	7050

Table 2: Types and details of GPUs used to benchmark numerical quadrature, as described in this section. The code was compiled with nVidia's compiler, `nvcc`, with CUDA toolkit version 6.0.

One advantage of evaluating the integrand on a GPU is the efficient summation of the integrands by a reduction operation via shared memory. This type of operation, unique to modern accelerators, allows a vector with N elements to be summed in a time $\propto \log(N)$ rather than $\propto N$ as may be expected when adding values sequentially to an accumulator. For two-dimensional integrals as in Equation (7), reduction is possible only over the first dimension, while summation over the second dimension may be carried out by using device memory to store intermediate partial sums. However, calculations performed on a variety of GPUs have shown that it is faster to evaluate the first dimension of the integral on each thread and then perform a reduction sum as for a one-dimensional integral.

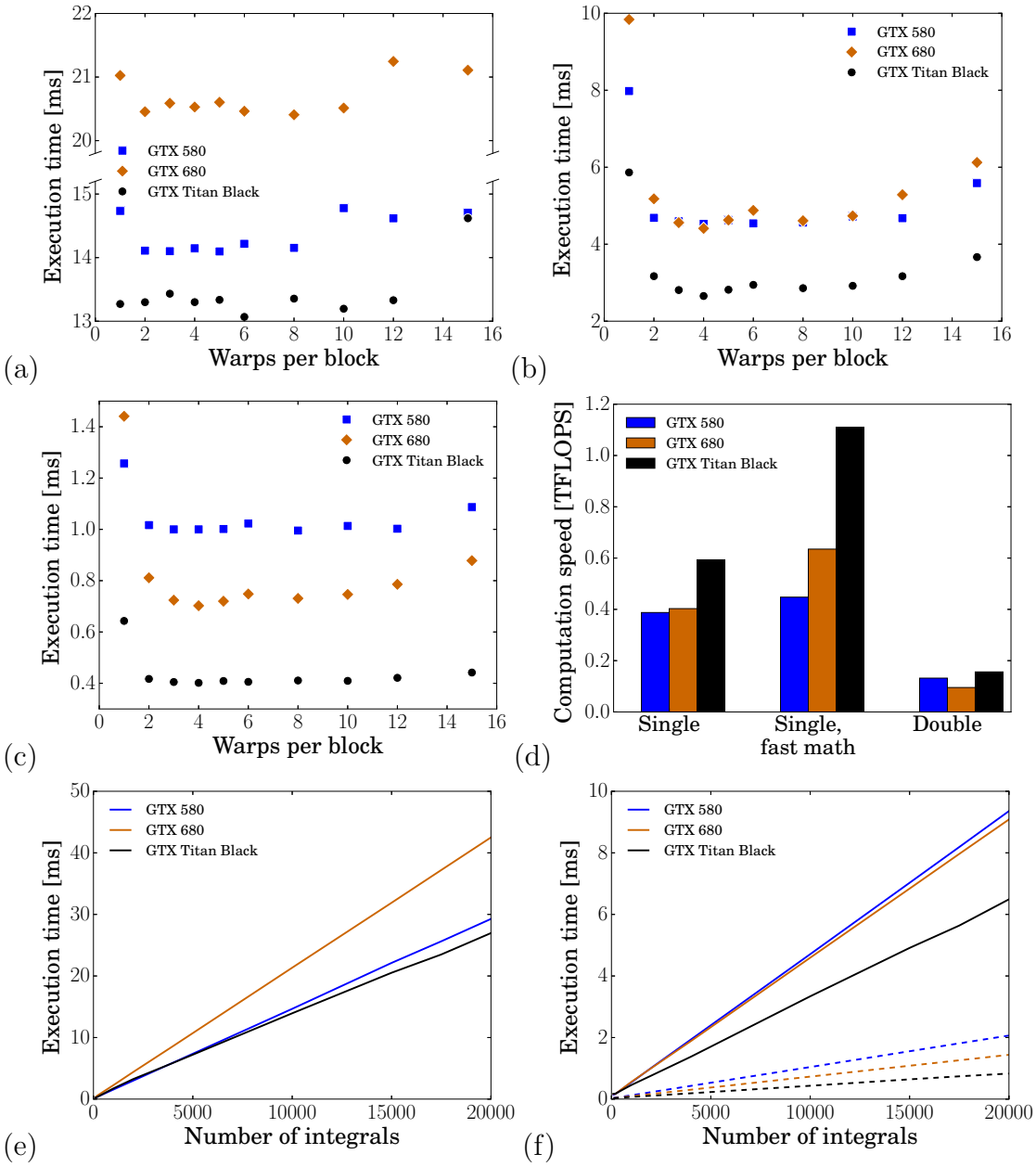

Figure 2: (color online). The total computation time (including memory copies) to carry out 9600 $J^{\uparrow}$, $K^{\uparrow}$ and $L^{\uparrow}$ integrals as a function of the number of warps per block y in (a) double precision (b) single precision and (c) single precision, with the `fast math` compiler option. (d) The computation speed in each case for the optimal choice of y. The total computation time for an optimal choice of y as a function of the number of integrals for (e) double precision and (f) single precision (solid) with `fast math` (dashed). The execution times were recorded using the nVidia profiler, `nvprof`.

All atomic rate coefficients as defined in §2.2 depend on only a few shared variables (such as electron density and temperature) and do not affect one another. This implies that the computation time is far larger than the memory copy latency, as a typical atomic model has between 10^2 and 10^5 atomic transitions, each with a corresponding integral. We show the calculation time when varying the size of the problem in Figures 2e and 2f. We confirm that the calculation time is linear with the size of the problem above some small threshold, corresponding to the memory copies.

Depending on the desired accuracy and experimental uncertainty in the cross-sectional data, it may be worthwhile to perform calculations in single precision. In this case, the observed speed-up in Figure 2 is consistent with the greater single precision processing power of GPUs. In addition, we see a speed-up in single precision when using the `fast math` compiler option, which reduces the number of floating point operations required to evaluate mathematical functions. It should be noted that this theoretically decreases the precision, though this has not been found to be noticeable. Calculating atomic rates using the Maxwell-Boltzmann distribution or by assuming a constant differential cross-section for collisional ionization may be necessary for performance issues, but produces errors of the order of tens of percent[11]; such errors are much larger than the loss in accuracy switching from double to single precision for the sake of the speed-up conferred.

This analysis shows a high level of utilization of the GPU for numerical quadrature in both double and single precision. The inherent parallelism of numerical quadrature applies equally well to CPUs, but the relatively high utilization of the GPU guarantees an advantage over a comparable number of CPUs for this type of calculation, as the former have far higher theoretical maximum performance.

4. Example collisional-radiative calculation

4.1. Model description

In order to demonstrate the use of GPUs to evaluate integrals in a realistic scenario, we have constructed a simple collisional-radiative model. With this aim in mind, we have reduced the physical and computational complexity of the model outside the calculation of rate coefficients. We consider an aluminum plasma close to solid density, as this element is well studied. Furthermore, as it is metallicaly bonded, the valence electrons may be modelled by a Fermi-Dirac distribution leading to a significant electron density (and

therefore degree of degeneracy) even at low temperatures. The cross-section data is taken from the ALADDIN database[12] and from the Los Alamos atomic physics code database[13].

We consider charge states states from 3+ to 6+, with a total of 14 superconfiguration levels (those, for which reliable atomic data was readily available); the lower ionization stages are not considered for simplicity, because aluminum is a metalically bonded solid whose valence electrons are taken to be free. Autoionization, dielectronic and radiative recombination are not included in this demonstrative model, because they do not involve the evaluation of an integral and therefore do not stand to benefit from GPU implementation. These processes are also typically of secondary importance for this type of plasma. We take a background radiation field to be that of a black body at $T_r = 250$ eV.

The collisional radiative rates of §2.2 are assembled into a rate matrix M and supplementary matrix M', so that the rates of change of the atomic populations $\vec{N}$ and electrons' kinetic energy ε are given by

$$\frac{d\vec{N}}{dt} = M\vec{N} \tag{27}$$

and

$$\frac{d\varepsilon}{dt} = \sum M'\vec{N} + \left(\frac{d\varepsilon}{dt}\right)_{\text{IB}} \tag{28}$$

respectively. Collisional-radiative codes have well-known approaches to solve these equations, particularly when they are "stiff" and the rate matrices are sparsely populated, which are beyond the scope of this work. We solve these coupled equations using the RK-4 method, keeping the timestep fixed for simplicity; for more details of our approach, see[14]. On the host CPU, matrix operations are carried out using standard BLAS routines, while the chemical potential and heat capacity are evaluated using Padé approximations. The latter cannot be readily parallelized, meaning that the total speed-up is smaller than might be expected from the speed of numerical quadrature in Section 3.3.

The code used in this work is available here[4]. It is split into two parts, so that the CPU-only code may be run without a working GPU or nVidia's compiler. The functions contained therein are ready to be included into an established collisional-radiative code.

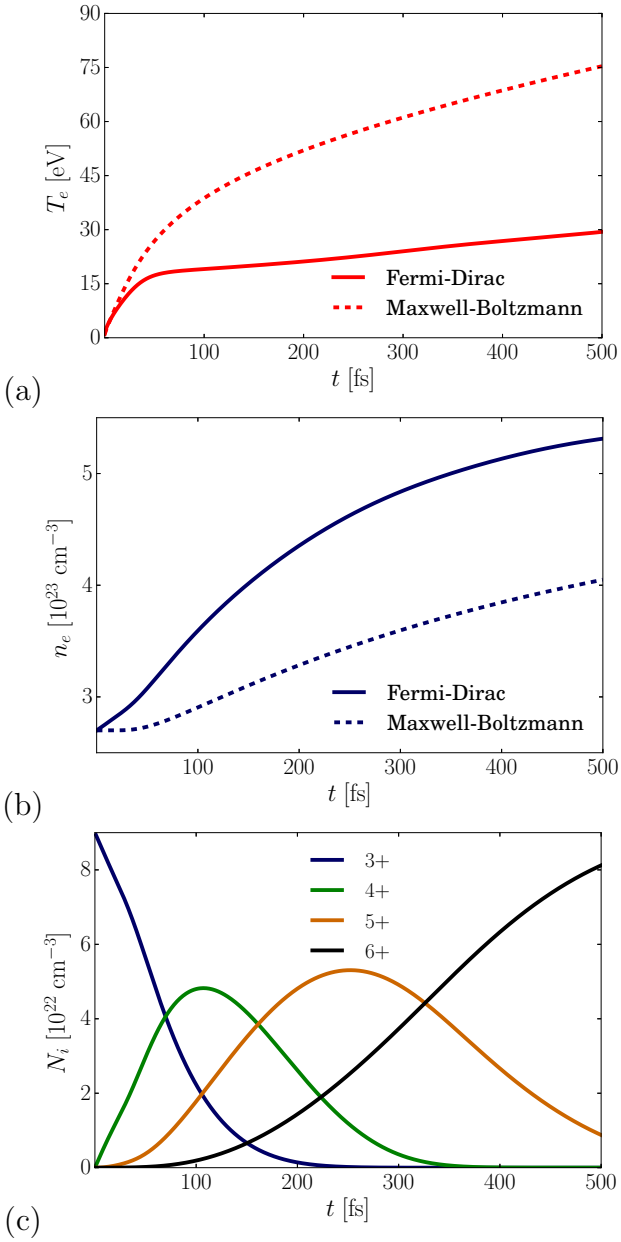

Figure 3: (color online). Results of a time dependent collisional-radiative simulation of near-solid density aluminum (total ion density 9×10^{22} cm^{-3}) irradiated by isotropic black body photons at $T_r = 250$ eV, comparing rate coefficients and heat capacity for free electrons with Fermi-Dirac (solid) and Maxwell-Boltzmann (dashed) statistics. (a) The electron temperature T_e and (b) density n_e. (c) Evolution of the ion populations for the Fermi-Dirac distribution.

4.2. Results

A sample calculation with 5×10^6 timesteps is shown in Figure 3. We compare a model with the degenerate rate coefficients (integrating over the Fermi-Dirac distribution) to one with analytical integrals over the Maxwell-Boltzmann distribution. Running a serial CPU-only version of the degenerate calculation took over 10 hours (on an Intel E5-2680 v2 processor), while the GPU-assisted version (with the GTX 580, as outlined in §3.3) took approximately 40 minutes. A notable speedup is therefore achieved, even comparable to the case where all 10 cores of the given CPU were to be fully utilized.

Physically, we see a large difference in results for the two distribution functions. There is a significantly higher degree of ionization (and hence electron density) in the case of the Fermi-Dirac distribution, which leads (if the absorbed energy is roughly comparable) to a lower temperature since the heat capacity is invariably positive. The main reason for this large difference in the degree of ionization is the effect of blocking factors. The rates of collisional ionization and three body recombination are both significantly slowed particularly at the relatively low temperatures considered here. Meanwhile, the rate of photoionization is affected far less by blocking factors due to the large photon energies of the Bose-Einstein distribution at $T_r = 250$ eV. Therefore, the rate of photoionization is proportionately higher than three body recombination in the case of the Fermi-Dirac distribution, leading to a higher degree of ionization.

5. Summary and conclusions

We have shown that a large practical speedup can be achieved by employing the large inherent degree of parallelism of GPUs to perform many simultaneous integrals. We have achieved a reasonable fraction of the theoretical maximum GPU speed, which far outperforms a single comparable CPU. The inherent latency of host/device memory transfer becomes negligible when many integrals are carried out with the same small set of input parameters. Further work implementing the evaluation of degenerate atomic rate coefficients using GPUs into existing collisional-radiative codes may yield additional increases in speed-up on heterogeneous clusters.

We have shown that GPUs can make tractable the problem of performing double integrals over the differential collisional ionization cross-section, which may be necessary for simulations of dense plasmas. We have presented the results of a time-dependent collisional-radiative model of aluminum, where

the valence electrons are treated as free, irradiated by black body radiation. The results show a significant difference between Fermi-Dirac and Maxwell-Boltzmann statistics when the free electron temperature is low and density is high.

Acknowledgements

Many thanks to Ben Dudson for excellent advice and to both Richard Evans and Rob Akers for advice about GPUs. The authors acknowledge UK funding from AWE Plc. and the Engineering and Physical Science Research Council (grant EP/J0194021).

Appendix A. General form of atomic rates

A generalised rate coefficient for a process involving m incoming and l outgoing interacting particles is given by

$$R = \int_0^\infty d\epsilon_0 \cdots \int_0^\infty d\epsilon_n \delta(S) \mathbb{B} \prod_{j=0}^{m} v_j f(\epsilon_j) \left[\prod_{k=0}^{l-1} \frac{d}{d\epsilon_k} \right] \sigma(\epsilon_0 \ldots \epsilon_n), \tag{A.1}$$

where ϵ is the kinetic energy of each particle, $v = \sqrt{2\epsilon/M}$ their velocity, $f(\epsilon)$ the energy probability distribution and σ the relevant quantum mechaincal cross-section. The δ represents the Dirac delta function with the argument being the sum over all energies

$$S = \sum_{j=0}^{m} \epsilon_j - E_{threshold} - \sum_{k=0}^{l} \epsilon_k, \tag{A.2}$$

which ensures energy conservation. The cross-section has been differentiated $l-1$ times, where the resulting differential cross-section is a measure of the distribution of the outgoing particles. The symbol $\mathbb{B}$ corresponds to the total effect of the Pauli blocking factors, given in a degenerate plasma by

$$\mathbb{B} = \prod_{k=0}^{l} \tilde{F}(\epsilon_k, T_e), \tag{A.3}$$

while in a classical Maxwellian plasma $\mathbb{B} = 1$. To obtain the total rate of transfer of ions from a given energy state, we multiply by the density of the given ions and the densities of all incoming particles,

$$\frac{dN_i}{dt} = -RN_i \prod_{j=0}^{m} N_j \tag{A.4}$$

The inverse rates of a given process are usually found through micro-reversibility relations. This allows a single cross-section to be used for a process and its inverse, only by exchanging the incoming particles with the outgoing.

While this formulation of the often familiar rate coefficients may seem needlessly complicated, it is necessary in order to take full account of all the interacting particles and their energies; the outgoing particles, which are ignored in the classical calculation, are particularly important when the plasma is degenerate, as detailed below. It should be noted that the kinetic energies of participating ions are typically not included in Equation (A.1), because their speed is unchanged by the collision due to their large mass. It is worth noting that one of the integrals in Equation (A.1) must be chosen to be carried out over the delta function, so the constraint of energy conservation naturally reduces the dimensionality of the integral by one.

Appendix B. Derivations of integrals' upper limits

As discussed in §2.3, we must truncate integrals at some finite upper limit while preserving their accuracy. Consider the integral

$$I = \int_{E_i}^{\infty} \frac{\epsilon_\gamma - E_i}{\exp(\epsilon_\gamma / T_r) - 1} \tilde{F}(\epsilon_\gamma - E_i, T_e) d\epsilon_\gamma, \tag{B.1}$$

which appears in the rate of heating due to photoionization by a black-body radiation spectrum of temperature T_r. It is the part of the integral in Equation (13) over the part of the cross-section containing the D_1 coefficient. Consider the case when $\mu \leq 0$, so that $\frac{1}{2} \leq \tilde{F}(\epsilon_\gamma - E_i, T_e) \leq 1$. In general, we have that $[\exp(\epsilon_\gamma / T_r) - 1]^{-1} > \exp(-\epsilon_\gamma / T_r)$. We therefore have a lower bound for the integral,

$$I \geq \frac{1}{2} \int_{E_i}^{\infty} (\epsilon_\gamma - E_i) \exp(-\epsilon_\gamma / T_r) d\epsilon_\gamma, \tag{B.2}$$

$$I \geq \frac{T_r^2}{2} \exp(-E_i / T_r). \tag{B.3}$$

We can break up the integral I into two parts (denoting the integrand by $H(\epsilon_\gamma)$ for short),

$$I = \int_{E_i}^{E_i+A} H(\epsilon_\gamma)d\epsilon_\gamma + \int_{E_i+A}^{\infty} H(\epsilon_\gamma)d\epsilon_\gamma, \tag{B.4}$$

$$I = \quad I_1 \quad + \quad I_2, \tag{B.5}$$

where I_1 will be our approximation to the integral and I_2 is an error term which we must minimise by appropriate selection of A. If we assume that $A \geq T_r$, then $\exp(\epsilon_\gamma/T_r) - 1 > \frac{1}{2}\exp(\epsilon_\gamma/T_r)$ over the entire domain of integration of I_2. Consequently,

$$I_2 \leq \int_{E_i+A}^{\infty} \frac{(\epsilon_\gamma - E_i)}{\exp(\epsilon_\gamma/T_r) - 1} d\epsilon_\gamma, \tag{B.6}$$

$$I_2 \leq 2\int_{E_i+A}^{\infty} (\epsilon_\gamma - E_i)\exp(-\epsilon_\gamma/T_r)d\epsilon_\gamma, \tag{B.7}$$

$$I_2 \leq 2T_r \exp\left(-\frac{E_i + A}{T_r}\right)(A + T_r). \tag{B.8}$$

It is clear that the relative error of the integral is given by I_2/I. Substituting our lower bound for I from Equation (B.3) and upper bound for I_2 from Equation (B.8), we have that

$$\frac{I_2}{I} \leq \frac{2T_r \exp\left(-\frac{E_i+A}{T_r}\right)(A + T_r)}{\frac{T_r^2}{2}\exp(-E_i/T_r)}. \tag{B.9}$$

Requiring that this error term be smaller than 10^{-p} and by introducing $x = A/T_r$, it follows that the solution to

$$4(x + 1)\exp(-x) = 10^{-p} \tag{B.10}$$

satifies this requirement.

In the case when $\mu > 0$, we can split the integral into three parts,

$$I = \int_{E_i}^{E_i+\mu} H(\epsilon_\gamma)d\epsilon_\gamma + \int_{E_i+\mu}^{E_i+\mu+A} H(\epsilon_\gamma)d\epsilon_\gamma + \int_{E_i+\mu+A}^{\infty} H(\epsilon_\gamma)d\epsilon_\gamma, \tag{B.11}$$

$$I = \quad I_0 \quad + \quad I_1 \quad + \quad I_2. \tag{B.12}$$

The lower bound estimate of Equation (B.3) now holds for I_1 and clearly $I \geq I_1$. Therefore, we can choose as the upper limit $A + \mu$, where A is the result derived above. We simultaneously evaluate the two photoionization integrals of Equations (11) and (13), each with two cross-section terms. Repeating this procedure, we find that this limit is larger than the other three, so we select it to allow simultaneous evaluation of both integrals.

The strategy for obtaining upper limits for the other integrals is similar to the example given here. We use analytically integrable functions to obtain a lower bound for the total integral I and an upper bound for the error term I_2. Their ratio allows an upper limit to be calculated given a required accuracy, similarly to Equation (B.9). Such an expression may then be simplified for the sake of increasing computation speed by narrowing the range of possible parameters; the case above may be simplified, for example, if we could guarantee that $\mu \leq 0$.

Appendix C. Nodes and weights for 32-point Gaussian quadrature

x_j	w_j
1.36806908E-03	3.50930500E-03
7.19424423E-03	8.13719737E-03
1.76188722E-02	1.26960327E-02
3.25469620E-02	1.71369315E-02
5.18394221E-02	2.14179490E-02
7.53161931E-02	2.54990296E-02
1.02758102E-01	2.93420467E-02
1.33908941E-01	3.29111114E-02
1.68477867E-01	3.61728971E-02
2.06142121E-01	3.90969479E-02
2.46550046E-01	4.16559621E-02
2.89324362E-01	4.38260465E-02
3.34065699E-01	4.55869393E-02
3.80356319E-01	4.69221995E-02
4.27764019E-01	4.78193600E-02
4.75846167E-01	4.82700443E-02
5.24153833E-01	4.82700443E-02
5.72235981E-01	4.78193600E-02
6.19643681E-01	4.69221995E-02
6.65934301E-01	4.55869393E-02
7.10675638E-01	4.38260465E-02
7.53449954E-01	4.16559621E-02
7.93857879E-01	3.90969479E-02
8.31522133E-01	3.61728971E-02
8.66091059E-01	3.29111114E-02
8.97241898E-01	2.93420467E-02
9.24683807E-01	2.54990296E-02
9.48160578E-01	2.14179490E-02
9.67453038E-01	1.71369315E-02
9.82381128E-01	1.26960327E-02
9.92805756E-01	8.13719737E-03
9.98631931E-01	3.50930500E-03

Table 3: The nodes and weights, as defined for use in Equation (22), for 32-point Gaussian quadrature to 8 decimal places.

References

[1] B. Deschaud, O. Peyrusse, and F. B. Rosmej, EPL **108,** 53001 (2014).

[2] B. Deschaud, O. Peyrusse, and F. B. Rosmej, HEDP **15,** 22 (2015).

[3] V. Aslanyan and G. J. Tallents, Phys. Rev. E **91,** 063106 (2015).

[4] *https://github.com/Valentin-Aslanyan/Fermi-Dirac-GPU*

[5] C. Kittel, *Introduction to Solid State Physics 8th Edition* (Wiley, 1996).

[6] V. V. Karasiev, D. Chakraborty and S. B. Trickey, Computer Physics Communications **192,** 114 (2015).

[7] J. Oxenius, *Kinetic Theory of Particles and Photons: Theoretical Foundations of Non-LTE Plasma Spectroscopy* (Springer-Verlag, 1986).

[8] K. L. Bell, H. B. Gilbody, J. G. Hughes, A. E. Kingston, and F. J. Smith, J. Phys. Chem. Ref. Data **12**, 891 (1983).

[9] G. H. Golub and J. H. Welsch, Mathematics of Computation **23,** 221 (1969).

[10] L. N. Trefethen, SIAM Review **50,** 67 (2008).

[11] G. J. Tallents, HEDP **20,** 9 (2016).

[12] Web interface of the IAEA's ALADDIN database, *https://www-amdis.iaea.org/ALADDIN/*

[13] Web interface of the LANL atomic physics codes, *http://aphysics2.lanl.gov/tempweb/lanl/*

[14] V. Aslanyan and G. J. Tallents, Phys. Plasmas **21,** 062702 (2014).