High Energy Density Physics 7 (2011) 43–46

Contents lists available at ScienceDirect

High Energy Density Physics

journal homepage: www.elsevier.com/locate/hedp

Efficient algorithm for generating spectra using line-by-line methods

Vijay Sonnad, Carlos A. Iglesias*

Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, CA 94550, USA

ARTICLE INFO

Article history:
Received 4 November 2010
Received in revised form
19 November 2010
Accepted 10 December 2010
Available online 9 January 2011

Keywords:
Plasma spectroscopy
Opacity
Emission spectra
Spectral lines

ABSTRACT

A method for efficient generation of spectra using line-by-line approaches is presented. The only approximation is replacing the line shape function with an interpolation procedure, which makes the method independent of the line profile functional form. The computational savings for a large number of lines is proportional to the number of frequency points in the spectral range. Therefore, for large-scale problems the method can reduce the operation count by several orders of magnitude.

1. Introduction

Calculations of spectra using line-by-line methods can entail large computational efforts [1]. That is, the bound–bound spectrum at photon energy $\hbar\omega$ is the sum of all the spectral lines weighted by their strength,

$$I(\omega) = \sum_a I_a(\omega) \tag{1.1}$$

$$I_a(\omega) = \sum_{k=1}^{N_{al}} s_{ak}\phi_a(\omega - \nu_{ak}) \tag{1.2}$$

where each a represents a group of N_{al} lines having the same normalized profile $\phi_a(x)$ with s_{ak} and ν_{ak} the strength and center of the kth line in the group, respectively. For example, in opacity calculations a line group typically consists of a configuration-to-configuration transition array where all the lines within the array are assigned the same profile [2].

The operation count to compute $I_a(\omega)$ directly from Eq. (1.2) given the atomic data is

$$N_{l_a} \propto O\left(N_{al}N_\omega N_\phi\right) \tag{1.3}$$

where N_ω is the number of frequency points and N_ϕ the number of operations to evaluate ϕ at a given frequency. Since $N_\phi \approx 10$ even for the simple Lorentz profile, $N_\omega = 10^{3-4}$ is typically required in opacity calculations, and configuration-to-configuration transition arrays in low- to mid-Z elements can often have $N_l = 10^{5-6}$, evaluation of $I_a(\omega)$ can take considerable computational effort [1]. Thus, efficient algorithms to generate the spectrum are welcome.

The purpose here is to present a procedure previously implemented in the TOPAZ opacity code [2] that is significantly faster than the explicit calculation of Eq. (1.2) and may benefit the growing number of efforts performing line-by-line calculations.

2. Method

The proposed procedure to generate spectra for myriad lines with identical profiles can be separated into three steps: interpolation, line accumulation, and convolution. It is assumed that the spectrum is to be computed on a uniform frequency mesh $\{\omega_i\}$ with $\omega_{i+1} - \omega_i = h$ for all i.

2.1. Interpolation

The first step is to replace explicit evaluation of the profile by interpolation, which is computationally faster and does not significantly impact accuracy. When many interpolations with the same data set and different arguments must be carried out, the Newton polynomial proves efficient. The line profile and divided-differences are computed on a uniform detuning mesh $\{x_j\}$ (frequency measured from line center, $x_o = 0$) with $x_{j+1} - x_j = h$ for all j. Then the interpolation for the profile at detuning x is given by [3]

* Corresponding author.
E-mail address: iglesias1@llnl.gov (C.A. Iglesias).

1574-1818/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.hedp.2010.12.002

$$\phi(x)=\phi^{(0)}[x_j]+\Delta_k\phi^{(1)}[x_j,x_{j+1}]+\Delta_k(h-\Delta_k)\phi^{(2)}[x_j,x_{j+1},x_{j+2}]+\cdots \tag{2.1.1}$$

where the divided-differences are obtained from the recursion formulas

$$\phi^{(n)}[x_j,\ldots,x_{j+n}]=\begin{cases}\phi(x), & n=0\\ \dfrac{\phi^{(n-1)}[x_{j+1},\ldots,x_{j+n}]-\phi^{(n-1)}[x_j,\ldots,x_{j+n-1}]}{x_{j+n}-x_j}, & n\geq 1\end{cases} \tag{2.1.2}$$

the kth line center displacement from the grid points is

$$\Delta_k = x-x_j \tag{2.1.3}$$

the index j is defined by

$$x_j\leq x<x_{j+1} \tag{2.1.4}$$

and only terms through second-order were included with the group index a suppressed.

2.2. Line accumulation

The second step makes no approximation and involves replacing the sum over lines by a sum over frequencies effectively accumulating the lines. Substitute the results of Section 2.1 into Eq. (1.2) at the grid points $\{\omega_i\}$ to obtain for a given line group

$$I_a(\omega_i)\approx\tilde{I}_a(\omega_i)=\sum_{k=1}^{N_\ell}s_k\Big\{\phi^{(0)}[\omega_i-\omega_j]+\Delta_k\phi^{(1)}[\omega_i-\omega_j,\omega_{i+1}-\omega_j]+\Delta_k(h-\Delta_k)\phi^{(2)}[\omega_i-\omega_j,\omega_{i+1}-\omega_j,\omega_{i+2}-\omega_j]\Big\} \tag{2.2.1}$$

where now the line center displacement and frequency index j are given by

$$\Delta_k = \omega_j-\nu_k \tag{2.2.2}$$

$$\omega_{j-1}<\nu_k\leq\omega_j \tag{2.2.3}$$

A pictorial representation of the interpolation scheme leading to Eq. (2.2.1) is shown in Fig. 1 for the kth line. In effect, the interpolating profile is shifted so that the line center, $\phi(0)$, coincides with the kth line center. The interpolated value at photon frequency ω_i is obtained using the appropriate segment of the Newton polynomial corresponding to detuning $x=\omega_i-\nu_k$.

It is possible to write the sum over lines in Eq. (2.2.1) as a sum over frequency points by noting that for a given j all the lines that satisfy Eq. (2.2.3) involve the same $\phi^{(n)}$ coefficients. Thus, Eq. (2.2.1) can be exactly rewritten as

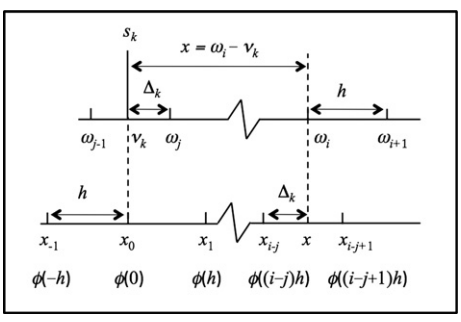

Fig. 1. Interpolation using Newton divided-differences for a line with strength, center, and displacement denoted by s_k, ν_k, and Δ_k.

$$\tilde{I}_a(\omega_i)=\sum_{j=1}^{N_\omega}\Big\{S_0(\omega_j)\phi^{(0)}[\omega_i-\omega_j]+S_1(\omega_j)\phi^{(1)}[\omega_i-\omega_j,\omega_{i+1}-\omega_j]+S_2(\omega_j)\phi^{(2)}[\omega_i-\omega_j,\omega_{i+1}-\omega_j,\omega_{i+2}-\omega_j]\Big\} \tag{2.2.4}$$

where the accumulated quantities are defined by

$$S_0(\omega_j)=\sum_{k=1}^{N_\ell}{}'s_k, \tag{2.2.5a}$$

$$S_1(\omega_j)=\sum_{k=1}^{N_\ell}{}'s_k\Delta_k, \tag{2.2.5b}$$

$$S_2(\omega_j)=\sum_{k=1}^{N_\ell}{}'s_k\Delta_k(h-\Delta_k) \tag{2.2.5c}$$

and $\sum'$ denotes a restricted sum so that j satisfies Eq. (2.2.3) otherwise $S_n(\omega_j)=0$.

2.3. Convolution

The final step recognizes the expression in Eq. (2.2.4) as the discrete convolution of the functions S_n and $\phi^{(n)}$. In terms of indices, Eq. (2.2.4) is symbolically written as

$$\tilde{I}_a(i)=\sum_{j=1}^{N_\omega}\Big\{S_0(j)\phi^{(0)}[i-j]+S_1(j)\phi^{(1)}[i-j]+S_2(j)\phi^{(2)}[i-j]\Big\} \tag{2.3.1}$$

and can be efficiently performed using fast Fourier transform (FFT) methods [4].

2.4. Operation count

The calculation of $I_a(\omega)$ with the proposed method requires $O(N_\omega N_\phi)$ operations to compute the profile and divided-differences on the detuning mesh. The line accumulation is performed by going through the group line list, determining for each line the j index satisfying Eq. (2.2.3) (simple arithmetic for a uniform mesh), and adding the line contribution to the accumulated quantity at index j. Therefore, the line accumulation in Eq. (2.2.5) is $O(N_l)$ operations. Although explicit convolution is $O(N_\omega^2)$, taking advantage of the FFT methods makes it $O(N_\omega\log_2 N_\omega)$ [4]. All together the operation count to evaluate $I_a(\omega)$ with the present method is then

$$N_{I_a}^{\mathrm{FFT}}\approx O(N_\ell)+O(N_\omega\log_2 N_\omega)+O\Big(N_\omega N_\phi\Big) \tag{2.4.1}$$

Comparing the results in Eqs. (1.3) and (2.4.1) yields the ratio

$$\frac{N_{I_a}}{N_{I_a}^{\mathrm{FFT}}}\approx\frac{O\Big(N_\ell N_\omega N_\phi\Big)}{O(N_\ell)+O(N_\omega\log_2 N_\omega)+O\Big(N_\omega N_\phi\Big)}\underset{N_\ell>>N_\omega}{\rightarrow}O(N_\omega)+O\left(\frac{N_\omega}{N_\ell}\right) \tag{2.4.2}$$

which for $N_\omega << N_\ell$ is $O(N_\omega)$ leading to large computational savings.

3. Example

An example is presented to illustrate the method. The line group is assumed to have 3 lines and $I_a(\omega)$ is evaluated on 6 uniformly