MERL

Multi-**E**lectron **R**adiator **L**ine-shape

M. F. Gu and I. E. Golovkin
Prism Computational Sciences, WI, USA
R. C. Mancini
Physics Department, Univ. of Nevada, Reno, USA

7th Spectral Line Shape Code Comparison Workshop, Gran Canaria, SPAIN, Sep 30-Oct 4 2024

Theory and model

- MERL is based on the theory and formalism of Stark-broadened line shapes developed by the late Professor Chuck Hooper and his students at the University of Florida
- Reference framework: standard Stark broadening theory approximation
- Static ions and dynamic electrons:

$$I(\omega) = \int P(\varepsilon) J(\omega, \varepsilon) d\varepsilon$$

Electron broadening

- MERL implements the physics model for H-like He (He^{+1}) developed by John O'Brien (PhD 1970)[1] and extended for highly-charged ions by Richard J. Tighe (PhD 1978)[2]
- Electron broadening: second order quantum mechanical relaxation theory with a dipole interaction potential
- Perturbing (plasma) electrons represented by Coulomb wave functions (Karzas and Latter free-free Gaunt factor[3]) and the thermal average performed over a Maxwellian electron distribution
- Cut-off at plasma frequency to approximate the effect of electron correlations[4]

1. J. O'Brien and C.F. Hooper, Jr., J. Quant. Spectrosc. and Radiative Transfer **14**, 479 (1974)
2. R.J. Tighe and C.F. Hooper, Jr., Physical Review A **17**, 410 (1978)
3. W.J. Karzas and R. Latter, Astrophysical Journal Supp. **55**, 167 (1961)
4. E.W. Smith, Physical Review Letters **18**, 990 (1967)

Ion microfield distribution

- The static-ion microfield distibution function is computed with either the Tighe-Hooper (TH) perturbation model[1] or the APEX[2] model
- TH: radiator net charge (χ) and two different types of perturbing ions (Z_1 and Z_2); arbitrary relative concentration $N_R=N_1/N_2$; differing electron and ion temperatures ($T_R=T_e/T_i$)
- APEX: radiator net charge and perturbing ion of type 1 (Z_1), and perturbing ion of type 2 (Z_2); differing electron and ion temperatures ($T_R=T_e/T_i$)
- Fitting formula developed by Gu-Beiersdorfer[3] can aslo be used to generate microfield distribution

1. R.J. Tighe and C.F. Hooper, Jr., Physical Review A **15**, 1773 (1977)
2. C.A. Iglesias, H.E. DeWitt, J.L. Lebowitz, D. Mac Gowan and W.B. Hubbard, Physical Review A **31**, 1698 (1985)
3. M. F. Gu, P. Beiersdorfer, Physical Review A 101, 032501 (2020)

Code details and numerics[1]

- The code is flexible and general, i.e. no “hard-wiring”
- The case is defined via a series of input files
- The microfield distribution input file is computed with either TH, APEX, or computed with a new fitting formula
- The atomic physics input file is computed with Cowan’s atomic structure code (or FAC, HULLAC)
- The original code is written in FORTRAN 77: a pre-processor program is used to read the atomic physics input file, calculate the size of the block matrices and set values in the PARAMETER statements of the code
- MERL is a stable and robust production code

1. R.C. Mancini, D.P. Kilcrease, L.A. Woltz and C.F. Hooper, Computer Physics Comm. **63**, 314 (1991)

Code details and numerics (II)

- The code reads (fine structure) energy level structure and reduce dipole matrix elements from the atomic physics input file and performs block diagonalization to reduce the size of the matrices as much as possible
- The size of the matrices are determined by:
  - complexity of atomic physics, i.e. energy level structure and matrix elements and selection rules,
  - and how complete is the electron broadening calculation
- Electron broadening options:
  - default: include upper level Stark shift and broadening,
  - include lower level Stark shift and broadening,
  - include interference term
- In general, the more complete the electron broadening calculation is the larger the sizes and the smaller the number of block matrices

Code details and numerics (III)

- The outer do-loop performs the microfield integration
- For every microfield ε, best ω-grid is not known apriori
- For every microfield ε, J(ω,ε) is first computed for a prescribed number of ω-points; this calculation gives a preliminary J(ω,ε)
- A peak-finding routine optimizes the ω-grid in order to have all peaks well resolved and described
- The peak-finding routine yields the best J(ω,ε) typically described in a non-uniform and customized ω-grid
- This J(ω,ε) is then mapped into a dense and uniform ω-grid by interpolation that is used in the microfield integration

Code details and numerics (IV)

- Other available options:
  - remove the non-quenching approximation, i.e. allow mixing between upper and lower levels, to treat intra-shell transitions
  - include a NLTE distribution populations for the upper levels read in from an input file, i.e. default is to use an LTE distribution computed inline via a Boltzmann factor[1]
  - include Doppler broadening via convolution with a Gaussian
  - include natural broadening (spontaneous radiative decay, autoionization) via augmentation of diagonal matrix elements of the electron broadening operator
  - Include ion quadrupole – field gradient interaction[2]
  - include ion dynamics effects via BID model[3,4]
  - Include dense plasma line shifts[5]

1. L.A. Woltz, V.L. Jacobs, C.F. Hooper, Jr. and R.C. Mancini, Physical Review A **44**, 1281 (1991)
2. D.P. Kilcrease, R.C. Mancini and C.F. Hooper, Jr. Physical Review E **48**, 3901 (1993)
3. D.B. Boercker, C.A. Iglesias and J.W. Dufty, Physical Review A **36**, 2254 (1987)
4. D.A. Haynes, Jr., D.T. Garber, C.F. Hooper, Jr., R.C. Mancini et al, Physical Review E **53**, 1042 (1996)
5. G.C. Junkel, M.A. Gunderson, C.F. Hooper, Jr. and D.A. Haynes, Jr., Physical Review E **62**, 5584 (2000)

Recent Updates

- Dynamic memory allocation
- Automated atomic data preparation with FAC
- Micro-field generation with APEX or inline fitting formula
- Faster linear algebra with mkl (intel, apple)
- Automated building procedure with Makefile
- Parallel execution, partition on photon energy and micro-field points to different processors
- Scheduling with ssh on heterogeneous network
- PBS on Clusters, can be extended for SLURM
- Command line interface with Python scripts

Details of Case 11 Calculation

- Lower and Upper level broadening
- Perturbing states within 2s->2p, 3l->3d, 3l->4l’
- With and without interference
- Te=100 eV, Ne = 1E24, 3E24, and 1E25