



FAC & cFAC

Evgeny Stambulchik

Faculty of Physics, Weizmann Institute of Science, Rehovot 7610001, Israel

Joint ICTP-IAEA School on Atomic Processes in Plasmas
February 27 – March 3, 2017
Trieste, Italy

Main features

- Calculates atomic structure, radiative transition rates, electron impact excitation and ionization, photo ionization and radiative recombination, autoionization and dielectronic capture.
- The atomic structure calculations are based on the relativistic configuration interaction (CI) with independent particle basis wavefunctions.
- The basis wavefunctions are derived from a local central potential, which is self-consistently determined to represent electronic screening of the nuclear potential.
- Relativistic effects are fully taken into account using the Dirac Coulomb Hamiltonian.
- Higher order QED effects are included: Breit interaction and self-energy and vacuum polarization effects.
- Continuum processes are treated in the distorted-wave (DW) approximation.
- Fast and reasonably accurate ($\sim 10^{-4}$ in level energies).

FAC vs. cFAC

Short history

FAC

is a software package for the calculation of various atomic processes, written by Ming Feng Gu at the Space Science Laboratory of Berkeley.

Latest release: v1.1.4 (9 Nov 2016).

cFAC

was started around 2010 (based on FAC-1.1.1^a, released in 2006), initially focusing on providing large volumes of data as required, e.g., for C-R plasma modeling, and eliminating reliance upon third-party Fortran numerical libraries with their C equivalents (hence the change in the package name).

Latest release: v1.6.1 (30 Nov 2016).

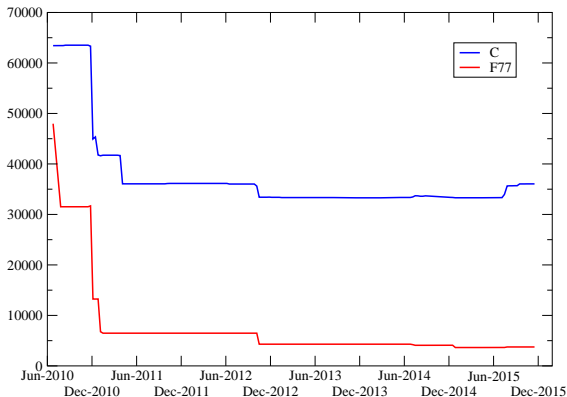
^aNot uniquely defined...

cFAC source lines of code (SLoC)

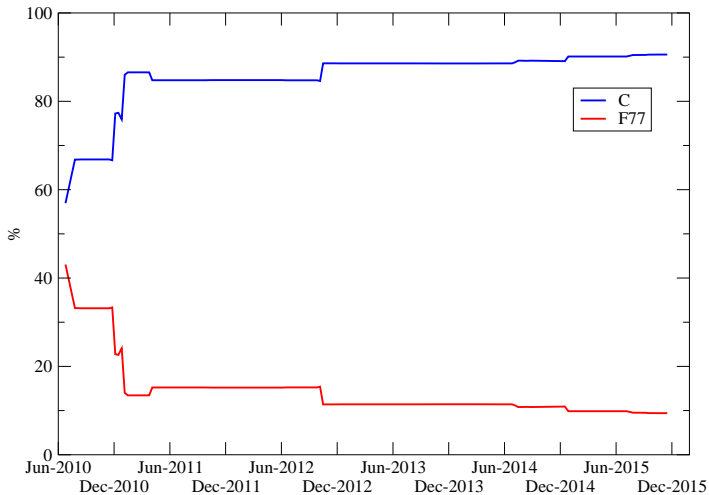
One of my most productive days was throwing away 1,000 lines of code.

Ken Thompson

cFAC source lines of code



cFAC source lines of code (relative)



FAC vs. cFAC

Availability

Home page for both FAC and cFAC

<https://www-amdis.iaea.org/FAC/>

Github repository for FAC

<https://github.com/fnevgeny/fac>

Maintainers: M. F. Gu and E. Stambulchik

Github repository for cFAC

<https://github.com/fnevgeny/cfac>

Maintainer: E. Stambulchik

<https://github.com/fnevgeny/cfac/issues>
to submit feature requests or report bugs.

FAC vs. cFAC

Licensing

FAC

Recently (summer 2015), Ming Feng Gu kindly agreed to release the FAC sources under the GPL (version 3 or higher) license:

<http://www.gnu.org/licenses/gpl-3.0.en.html>.

However, the GPL'ed sources (authored by MFG) are only part of the whole FAC package. Use at your own discretion!

cFAC

Some bits of the FAC sources, which are still used in cFAC, were published in *Computer Physics Communications*, and as such, are licensed for non-profit or academic use only, see

<http://cpc.cs.qub.ac.uk/licence/licence.html>.

To compile in these *optional* CPC-licensed modules, pass the `--with-cpc-module` configure flag and explicitly agree to the CPC licensing terms. As a result, the “sfac” executable will not be redistributable!

FAC vs. cFAC

Compatibility

FAC components

SFAC and PFAC—a thin Python wrapper around SFAC + CRM and POL modules.

cFAC components

SFAC, PFAC, CFACDB library, CFACDBU utility, CFACDB browser/explorer.

SFAC compatibility

In general, SFAC scripts that work with FAC should work unchanged with cFAC. However, MBPT and R-matrix modules were removed.

SFAC interface

SFAC can be used both in interactive and batch mode. When invoked without arguments, it runs in the interactive mode:

```
% sfac

>>> Info()
=====
cFAC-1.6.1 http://github.com/fnevgeny/cfac

Based on the Flexible Atomic Code (FAC)
by Ming Feng Gu

Maintained by Evgeny Stambulchik
=====
>>> Print('Hello!')
Hello!
>>> Exit()
%
```

Examples and manual

Let's copy a few demos to your home directory:

```
% cp -a /home/nfs3/smr3105/estambul/cfac-demo ~/
% cd ~/cfac-demo
%
```

It is a good opportunity to load the official documentation:

```
%acroread /home/nfs3/smr3105/share/doc/cfac/manual.pdf&
%
```

Electronic configurations

```
% gedit Fe_XVII.sf &
```

```
# Define the species
```

```
SetAtom('Fe')
```

```
# (as you see, lines beginning with # are comments)
```

```
# Ne-like ion:
```

```
# Ground state
```

```
Config('n2', '1s2 2*8')
```

```
# ... and a group of singly-excited to n=3 states
```

```
Config('n3', '1s2 2*7 3*1')
```

```
# It is a good idea to check the configurations explicitly:
```

```
ListConfig()
```

Electronic configurations (cont.)

Let's run sfac:

```
% sfac Fe_XVII.sf

n2      0    1s+2 2s+2 2p-2 2p+4
n3      1    1s+2 2s+2 2p-2 2p+3 3d+1
n3      2    1s+2 2s+2 2p-1 2p+4 3d+1
n3      3    1s+2 2s+1 2p-2 2p+4 3d+1
n3      4    1s+2 2s+2 2p-2 2p+3 3d-1
...
...
n3      15   1s+2 2s+1 2p-2 2p+4 3s+1

%
```

“ $n\ell\pm$ ” means one-electron $|n\ell j = \ell \pm 1/2\rangle$.

$$H = \sum_{i=1}^N H_D(i) + \sum_{i<j}^N \frac{1}{r_{ij}},$$

$H_D(i)$ is the single-electron Dirac Hamiltonian.

By diagonalizing the total Hamiltonian, one obtains

$$\psi = \sum_{\nu} b_{\nu} \Phi_{\nu},$$

with the configuration state functions (CSF's) Φ_{ν} being the Slater constructs of N one-electron Dirac spinors $\varphi_{n\kappa m}$

$$\varphi_{n\kappa m} = \frac{1}{r} \begin{pmatrix} P_{n\kappa}(r) \chi_{\kappa m}(\theta, \phi, \sigma) \\ iQ_{n\kappa}(r) \chi_{-\kappa m}(\theta, \phi, \sigma) \end{pmatrix}$$

κ is the relativistic angular quantum number:

$$\kappa = (l - j)(2j + 1).$$

In the Dirac-Fock-Slater method, $P_{n\kappa}$ and $Q_{n\kappa}$ satisfy

$$\begin{aligned}\left(\frac{d}{dr} + \frac{\kappa}{r}\right)P_{n\kappa} &= \alpha\left(\varepsilon_{n\kappa} - V + \frac{2}{\alpha^2}\right)Q_{n\kappa} \\ \left(\frac{d}{dr} - \frac{\kappa}{r}\right)Q_{n\kappa} &= \alpha(-\varepsilon_{n\kappa} + V)P_{n\kappa},\end{aligned}$$

for an effective central field $V(r) = V^N(r) + V^{ee}(r)$.

For a given multipole operator O_M^L :

$$S_{fi} = |\langle \psi_f || O_M^L || \psi_i \rangle|^2 \equiv |M_{fi}|^2$$

$$gf_{fi} = (2L + 1)^{-1} \omega (\alpha \omega)^{2L-2} |M_{fi}|^2$$

$$gA_{fi} = 2\alpha^3 \omega^2 gf_{fi}$$

Note: M_{fi} differ from reduced matrix elements as defined, e.g., in the Cowan's book, by a factor of $\sqrt{2}$.

Electron impact excitation (EIE) I

Different treatments of the continuum wavefunctions:

- The plane-wave (PW) Born approximation uses an unperturbed plane wave for free orbitals.
- The Coulomb-wave (CW) Born approximation takes into account the distortion of the continuum due to a pure Coulomb potential.
- The most accurate of this class is the distorted-wave (DW) approximation, in which the free orbitals are calculated in a more realistic potential taking into account the electronic structure of the target ion.

By default, FAC uses DW for calculating EIE cross sections.

Note: DW is not accurate for neutral atoms; $Z \gtrsim 4$ should be fine.

Electron impact excitation (EIE) II

The EIE cross subsection σ_{01} from the initial state ψ_0 to the final state ψ_1 is

$$\sigma_{01} = \frac{\pi}{k_0^2 g_0} \Omega_{01},$$

g_0 is the statistical weight of the initial state, and k_0 is the kinetic momentum of the incident electron,

$$k_0^2 = 2\varepsilon_0 \left(1 + \frac{\alpha^2}{2} \varepsilon_0 \right),$$

and the collision strength is

$$\Omega_{01} = 2 \sum_{\kappa_0 \kappa_1} \sum_{J_T} (2J_T + 1) | \langle \psi_0 \kappa_0, J_T M_T | \sum_{i < j} \frac{1}{r_{ij}} | \psi_1 \kappa_1, J_T M_T \rangle |^2$$

Electron impact ionization (EII) I

$$\sigma(\varepsilon_0, \varepsilon) = \frac{1}{k_0^2 g_0} \Omega_{01}$$

(note a different from EIE numerical factor - π is missing).

DW calculations of EII processes are rather slow; by default, Binary-Encounter-Dipole (BED) approach is used.

Photoionization and Radiative Recombination

$$\sigma_{PI} = 2\pi\alpha \frac{df}{dE}$$
$$\sigma_{RR} = \frac{\alpha^2 g_i}{2 g_f \varepsilon} \frac{\omega^2}{(1 + 0.5\alpha^2\varepsilon)} \sigma_{PI},$$

The differential oscillator strength:

$$\frac{df}{dE} = \frac{\omega}{g_i} (2L + 1)^{-1} (\alpha\omega)^{2L-2} S,$$

where L is the rank of the multipole operator inducing the transition, and the generalized line strength is

$$S = \sum_{\kappa J_T} |\langle \psi_f, \kappa; J_T \| O^L \| \psi_i \rangle|^2,$$

and O^L is the multipole operator inducing the transition.

Autoionization and Dielectronic Recombination

In the first order perturbation theory, the AI rate is

$$A^a = 2 \sum_{\kappa} \left| \langle \psi_{f, \kappa}; J_T M_T | \sum_{i < j} \frac{1}{r_{ij}} | \psi_i \rangle \right|^2,$$

The inverse process is the dielectronic capture (DC). The DC strength is

$$S_{DC} = \frac{g_i}{2g_f} \frac{\pi^2}{E_{if}} A^a,$$

E_{if} is the resonance energy.

An AI state formed by DC may either autoionize, or radiatively decay. In the later case, one talks about dielectronic recombination (DR).

Note: FAC does not calculate the DR rate coefficients; one needs to loop over all the possible channels.

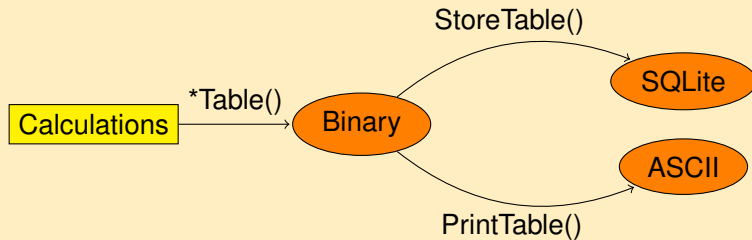
FAC vs. cFAC

Output format options

FAC



cFAC



Future



cFAC

An example of SFAC config file

An ultra-simple demo

SetAtom('Fe')

He-like

Config('2.gc', '1s2')

Config('2.ex', '1s1 2*')

Config('2.ai', '2*2')

H-like

Config('1.gc', '1s')

Config('1.ex', '2*1')

Bare nucleus

Config('0.gc', '')

cFAC

An example of SFAC config file (cont.)

```
# Start from the lowest charge state of interest and go up

# Keep user updated
Print('nele = 2 (He)')

ConfigEnergy(0)
OptimizeRadial(['2.gc'])
ConfigEnergy(1)

# Include configuration interactions between all
# 2-electron states
Structure('le.bin', 2)

# All radiative transitions of this (2-electron) charge state
# Calculate E1, M1, and E2
TRTable('tr.bin', 2, -1)
TRTable('tr.bin', 2, +1)
TRTable('tr.bin', 2, -2)
```

cFAC

An example of SFAC config file (cont.)

Again, all collisional transitions of He-like

CETable('ce.bin', 2)

Same for H

Print('nele = 1 (H)')

ConfigEnergy(0)

OptimizeRadial(['1.gc'])

ConfigEnergy(1)

Structure('le.bin', 1)

TRTable('tr.bin', 1, -1)

CETable('ce.bin', 1)

*# Now we have also ionization processes; He-like wavefunctions
are already calculated with the optimized (for He!) potential*

AITable('ai.bin', 2)

CITable('ci.bin', 2)

RRTable('rr.bin', 2)

cFAC

An example of SFAC config file (cont.)

```
Print('nele = 0')  
Structure('le.bin', 0)  
# For the bare nucleus, only CI and RR (from H) exist  
CITable('ci.bin', 1)  
RRTable('rr.bin', 1)  
  
# Initialize the SQLite database store, erasing if exists  
StoreInit('Fe.db', 1)  
  
# Store the binary data into the DB  
StoreTable('le.bin')  
StoreTable('tr.bin')  
StoreTable('ai.bin')  
StoreTable('ci.bin')  
StoreTable('rr.bin')  
StoreTable('ce.bin')  
  
# Close the DB  
StoreClose()
```

cFAC

An example of SFAC config file (cont.)

*# Alternatively (or in addition), you may want to use
the old FAC ASCII output format:*

MemENTable('le.bin')

PrintTable('le.bin', 'le.asc', 1)

PrintTable('tr.bin', 'tr.asc', 1)

PrintTable('ai.bin', 'ai.asc', 1)

PrintTable('rr.bin', 'rr.asc', 1)

PrintTable('ci.bin', 'ci.asc', 1)

PrintTable('ce.bin', 'ce.asc', 1)

- CFACDB is a set of application programming interface (API) calls for accessing cFAC-generated databases in the SQLite format from C or Fortran codes.
- In addition, collisional rates in a Maxwellian plasma can be obtained on-the-fly, using inter- and extrapolation between/beyond calculated data points.
- The data are returned via user-provided “sink” callback routines, with negligible CPU and memory overheads.

CFACDB browser

For manual inspection of up to moderate-size databases

A Graphical User Interface (GUI) tool. It is built as a Firefox extension, making it usable in any modern computer environment.

CFACDB browser

Example: the "mini" Fe case database

CFACDB

File Help

Species	Atomic number	Mass
Fe	26	56

Number of electrons: Δ Nele: 0 1 Transition ΔE = 251.893576750703

Initial level			Final level		
ID	Name	Energy	ID	Name	Energy
0	1s+2(0)0	-665.320073873914	0	1s+2(0)0	-665.320073873914
1	1s+1(1)1 2s+1(1)2	-421.481465706087	1	1s+1(1)1 2s+1(1)2	-421.481465706087
2	1s+1(1)1 2p-1(1)0	-420.412730525578	2	1s+1(1)1 2p-1(1)0	-420.412730525578
3	1s+1(1)1 2p-1(1)2	-420.336954319012	3	1s+1(1)1 2p-1(1)2	-420.336954319012
4	1s+1(1)1 2s+1(1)0	-420.33315114923	4	1s+1(1)1 2s+1(1)0	-420.33315114923
5	1s+1(1)1 2p+1(3)4	-419.797939722527	5	1s+1(1)1 2p+1(3)4	-419.797939722527
6	1s+1(1)1 2p+1(3)2	-419.121171173246	6	1s+1(1)1 2p+1(3)2	-419.121171173246
7	2s+2(0)0	-167.51846811663	7	2s+2(0)0	-167.51846811663
8	2s+1(1)1 2p-1(1)0	-167.404609824857	8	2s+1(1)1 2p-1(1)0	-167.404609824857
9	2s+1(1)1 2p-1(1)2	-167.227594422543	9	2s+1(1)1 2p-1(1)2	-167.227594422543
10	2s+1(1)1 2p+1(3)4	-166.653098546333	10	2s+1(1)1 2p+1(3)4	-166.653098546333
11	2p-2(0)0	-166.221355266761	11	2p-2(0)0	-166.221355266761
12	2p-1(1)1 2p+1(3)2	-165.860013871478	12	2p-1(1)1 2p+1(3)2	-165.860013871478
13	2p-1(1)1 2p+1(3)4	-165.613962660843	13	2p-1(1)1 2p+1(3)4	-165.613962660843
14	2s+1(1)1 2p+1(3)2	-165.310459209609	14	2s+1(1)1 2p+1(3)2	-165.310459209609
15	2p+2(4)4	-164.844863470214	15	2p+2(4)4	-164.844863470214
16	2p+2(0)0	-163.550783997367	16	2p+2(0)0	-163.550783997367

Radiative transitions

Multipole	Reduced matrix element	Calculation mode
1 (M1)	-0.0000298641389235854	0 (FR)
-2 (E2)	-4.55679725064329e-7	1 (NR)

Collisional processes

Type	Energy	Strength
1 (CE)	12.5442322469137	0.000086081272456795
1 (CE)	97.4950829824434	0.000067879242124036
1 (CE)	196.180063384014	0.0000547504496353213
1 (CE)	307.343101386054	0.000045575467083836
1 (CE)	429.521428025898	0.0000392289693991188
1 (CE)	561.254699998286	0.0000348375106113963
1 (CE)	701.200760037383	0.0000317872545565479