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.


cFAC
was started around 2010 (based on FAC-1.1.1\textsuperscript{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).


\textsuperscript{a}Not uniquely defined...
One of my most productive days was throwing away 1,000 lines of code.

---

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

```bash
% 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:

```bash
% 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&
%```
% 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()
Let's run sfac:

```bash
% sfac Fe_XVII.sf
```

<table>
<thead>
<tr>
<th>n</th>
<th>ℓ ±</th>
<th>1s+</th>
<th>2s±</th>
<th>2p±</th>
<th>3d±</th>
</tr>
</thead>
<tbody>
<tr>
<td>n2</td>
<td>0</td>
<td>1s+</td>
<td>2s+</td>
<td>2p-</td>
<td>2p+</td>
</tr>
<tr>
<td>n3</td>
<td>1</td>
<td>1s+</td>
<td>2s+</td>
<td>2p-</td>
<td>2p+ 3d+</td>
</tr>
<tr>
<td>n3</td>
<td>2</td>
<td>1s+</td>
<td>2s+</td>
<td>2p-</td>
<td>2p+ 3d+</td>
</tr>
<tr>
<td>n3</td>
<td>3</td>
<td>1s+</td>
<td>2s+</td>
<td>2p-</td>
<td>2p+ 3d+</td>
</tr>
<tr>
<td>n3</td>
<td>4</td>
<td>1s+</td>
<td>2s+</td>
<td>2p-</td>
<td>2p+ 3d-</td>
</tr>
</tbody>
</table>

... ... ...

| n3 | 15  | 1s+ | 2s+ | 2p- | 2p+ 3s+ |

```

“nl±” means one-electron \(|nlj = ℓ ± 1/2\).
\[ H = \sum_{i=1}^{N} H_D(i) + \sum_{i<j} \frac{1}{r_{ij}}, \]

\( H_D(i) \) 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) \( \Phi_\nu \) being the Slater constructs of \( N \) one-electron Dirac spinors \( \varphi_{n\kappa m} \)

\[ \varphi_{n\kappa m} = \frac{1}{r} \left( \begin{array}{c} P_{n\kappa}(r)\chi_{\kappa m}(\theta, \phi, \sigma) \\ iQ_{n\kappa}(r)\chi_{-\kappa m}(\theta, \phi, \sigma) \end{array} \right) \]

\( \kappa \) 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

\[
\left( \frac{d}{dr} + \frac{\kappa}{r} \right) P_{n\kappa} = \alpha \left( \epsilon_{n\kappa} - V + \frac{2}{\alpha^2} \right) Q_{n\kappa} \\
\left( \frac{d}{dr} - \frac{\kappa}{r} \right) Q_{n\kappa} = \alpha \left( -\epsilon_{n\kappa} + V \right) P_{n\kappa},
\]

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

For a given multipole operator $O^L_M$:

$$S_{fi} = \left| \langle \psi_f | O^L_M | \psi_i \rangle \right|^2 \equiv |M_{fi}|^2$$

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

$$g A_{fi} = 2 \alpha^3 \omega^2 g f_{fi}$$

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.
The EIE cross subsection $\sigma_{01}$ from the initial state $\psi_0$ to the final state $\psi_1$ 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

$$\Omega_{01} = 2 \sum_{\kappa_0\kappa_1} \sum_{J_T} (2J_T + 1) |<\psi_0\kappa_0, J_T M_T| \sum_{i<j} \frac{1}{r_{ij}} |\psi_1\kappa_1, J_T M_T>|^2$$
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 - \( \pi \) is missing).

DW calculations of EII processes are rather slow; by default, Binary-Encounter-Dipole (BED) approach is used.
\[\sigma_{PI} = 2\pi\alpha \frac{df}{dE}\]

\[\sigma_{RR} = \frac{\alpha^2 g_i}{2 g_f \epsilon (1 + 0.5\alpha^2 \epsilon)} \sigma_{PI},\]

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

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

\( E_{if} \) 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**

- Calculations
- *Table() → Binary → PrintTable() → ASCII

**cFAC**

- Calculations
- *Table() → Binary
- StoreTable() → SQLite
- PrintTable() → ASCII

**Future**

- Calculations
- ???? → SQLite
- ???? → ASCII
# An ultra-simple demo

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

# He-like

```python
Config('2.gc', '1s2')
Config('2.ex', '1s1 2*')
Config('2.ai', '2*2')
```

# H-like

```python
Config('1.gc', '1s')
Config('1.ex', '2*1')
```

# Bare nucleus

```python
Config('0.gc', '')
```
# 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)
# 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)
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()
# 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.
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

<table>
<thead>
<tr>
<th>Species</th>
<th>Atomic number</th>
<th>Mass</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fe</td>
<td>26</td>
<td>56</td>
</tr>
</tbody>
</table>

Number of electrons: 2, \[ \text{\Delta} \text{Rule: } 0 \circlearrowleft 1 \] Transition \[ \Delta E = 251.893576750703 \]

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1s+2(0)0</td>
<td>-665.320073873914</td>
</tr>
<tr>
<td>1</td>
<td>1s+1(1)1 2s+1(1)2</td>
<td>-421.481465706087</td>
</tr>
<tr>
<td>2</td>
<td>1s+1(1)1 2p-1(1)0</td>
<td>-420.412730525578</td>
</tr>
<tr>
<td>3</td>
<td>1s+1(1)1 2p-1(1)2</td>
<td>-420.336954319012</td>
</tr>
<tr>
<td>4</td>
<td>1s+1(1)1 2s+1(1)0</td>
<td>-420.33315114923</td>
</tr>
<tr>
<td>5</td>
<td>1s+1(1)1 2p-1(3)4</td>
<td>-419.79739372527</td>
</tr>
<tr>
<td>6</td>
<td>1s+1(1)1 2p+1(3)2</td>
<td>-419.12117173246</td>
</tr>
<tr>
<td>7</td>
<td>2s+2(0)0</td>
<td>-167.51846811663</td>
</tr>
<tr>
<td>8</td>
<td>2s+1(1)1 2p-1(1)0</td>
<td>-167.40460824857</td>
</tr>
<tr>
<td>9</td>
<td>2s+1(1)1 2p-1(1)2</td>
<td>-167.227594422543</td>
</tr>
<tr>
<td>10</td>
<td>2s+1(1)1 2p+1(3)4</td>
<td>-166.653098546333</td>
</tr>
<tr>
<td>11</td>
<td>2p-2(0)0</td>
<td>-166.221355266761</td>
</tr>
<tr>
<td>12</td>
<td>2p-1(1)1 2p+1(3)2</td>
<td>-165.86003871478</td>
</tr>
<tr>
<td>13</td>
<td>2p-1(1)1 2p+1(3)4</td>
<td>-165.61952660843</td>
</tr>
<tr>
<td>14</td>
<td>2s+1(1)1 2p+1(3)2</td>
<td>-165.31045920609</td>
</tr>
<tr>
<td>15</td>
<td>2p+2(4)4</td>
<td>-164.844863470214</td>
</tr>
<tr>
<td>16</td>
<td>2p+2(0)0</td>
<td>-163.550783997367</td>
</tr>
</tbody>
</table>

Radiative transitions

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1s+2(0)0</td>
<td>-665.320073873914</td>
</tr>
<tr>
<td>1</td>
<td>1s+1(1)1 2s+1(1)2</td>
<td>-421.481465706087</td>
</tr>
<tr>
<td>2</td>
<td>1s+1(1)1 2p-1(1)0</td>
<td>-420.412730525578</td>
</tr>
<tr>
<td>3</td>
<td>1s+1(1)1 2p-1(1)2</td>
<td>-420.336954319012</td>
</tr>
<tr>
<td>4</td>
<td>1s+1(1)1 2s+1(1)0</td>
<td>-420.33315114923</td>
</tr>
<tr>
<td>5</td>
<td>1s+1(1)1 2p-1(3)4</td>
<td>-419.79739372527</td>
</tr>
<tr>
<td>6</td>
<td>1s+1(1)1 2p+1(3)2</td>
<td>-419.12117173246</td>
</tr>
<tr>
<td>7</td>
<td>2s+2(0)0</td>
<td>-167.51846811663</td>
</tr>
<tr>
<td>8</td>
<td>2s+1(1)1 2p-1(1)0</td>
<td>-167.40460824857</td>
</tr>
<tr>
<td>9</td>
<td>2s+1(1)1 2p-1(1)2</td>
<td>-167.227594422543</td>
</tr>
<tr>
<td>10</td>
<td>2s+1(1)1 2p+1(3)4</td>
<td>-166.653098546333</td>
</tr>
<tr>
<td>11</td>
<td>2p-2(0)0</td>
<td>-166.221355266761</td>
</tr>
<tr>
<td>12</td>
<td>2p-1(1)1 2p+1(3)2</td>
<td>-165.86003871478</td>
</tr>
<tr>
<td>13</td>
<td>2p-1(1)1 2p+1(3)4</td>
<td>-165.61952660843</td>
</tr>
<tr>
<td>14</td>
<td>2s+1(1)1 2p+1(3)2</td>
<td>-165.31045920609</td>
</tr>
<tr>
<td>15</td>
<td>2p+2(4)4</td>
<td>-164.844863470214</td>
</tr>
<tr>
<td>16</td>
<td>2p+2(0)0</td>
<td>-163.550783997367</td>
</tr>
</tbody>
</table>

Collisional processes

<table>
<thead>
<tr>
<th>Type</th>
<th>Energy</th>
<th>Strength</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ( CE )</td>
<td>12.5442322469137</td>
<td>0.0000086081272456795</td>
</tr>
<tr>
<td>1 ( CE )</td>
<td>97.4950829824424</td>
<td>0.000008679242124036</td>
</tr>
<tr>
<td>1 ( CE )</td>
<td>196.189053384014</td>
<td>0.00005470549635213</td>
</tr>
<tr>
<td>1 ( CE )</td>
<td>307.34301086054</td>
<td>0.000004557567083636</td>
</tr>
<tr>
<td>1 ( CE )</td>
<td>429.521282985989</td>
<td>0.0000039228693991188</td>
</tr>
<tr>
<td>1 ( CE )</td>
<td>561.25469998286</td>
<td>0.000003837510613963</td>
</tr>
<tr>
<td>1 ( CE )</td>
<td>701.200758037383</td>
<td>0.0000031787254556579</td>
</tr>
</tbody>
</table>