A platform for Just Atomic Computations

– The Jena Atomic Calculator (JAC)

Stephan Fritzsche
Helmholtz-Institut Jena &
Theoretisch-Physikalisches Institut Jena

24th October 2022

A platform for Just Atomic Computations

superheavy elements

electric dipole moments

level energies

Fano profiles

electron affinities

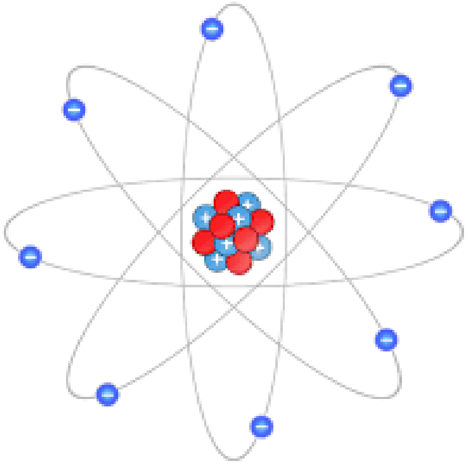
playground for
electronic structure theory

pair production

electron capture

polarizibilities

transition probabilities



plasma diagnostics

Rayleigh & Compton

autoionization & cascades

atomic form factors

ionization potentials

energy shifts in plasma

(multi-) photon ionization

DR plasma rates

Rydberg matter

bremsstrahlung

dispersion coefficients

internal conversion

Zeeman splittings & Lande factors

hyperfine splittings

Electron-impact processes

parity non-conservation

isotope shifts & nuclear moments

Coulomb excitation & ionization

alpha-variations

double (auto-) ionization

A platform for Just Atomic Computations

level energies

playground for
electronic structure theory

transition probab

Rayleig npton

ionization ials

(multi-) photon ion

Rydberg matter

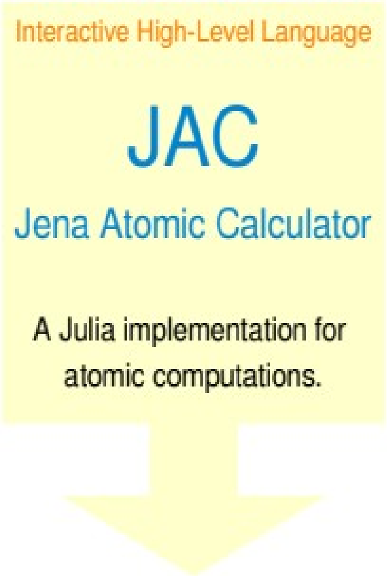
internal conversion

Electron-impact processes

isotope shifts & nuclear moments

alpha-variations

double (auto-) ionization



Interactive High-Level Language

JAC

Jena Atomic Calculator

A Julia implementation for
atomic computations.

electric dipole moments

electron affinities

electron capture

polarizibilities

plasm s

autoior & cascades

ato n factors

energy sh lasma

DR plasma rates

nts

ation

Coulom ion

Jena Atomic Calculator

(Fully) relativistic atomic structure code
with focus upon continuum processes & cascades;
for (almost) all shell structures and elements;
user-friendly, open & features on demand.

A platform for Just Atomic Computations

level energies

playground for
electronic structure theory

transition probab

Rayleig mpton

ionization ials

(multi-) photon ion

Rydberg matter

internal conversion

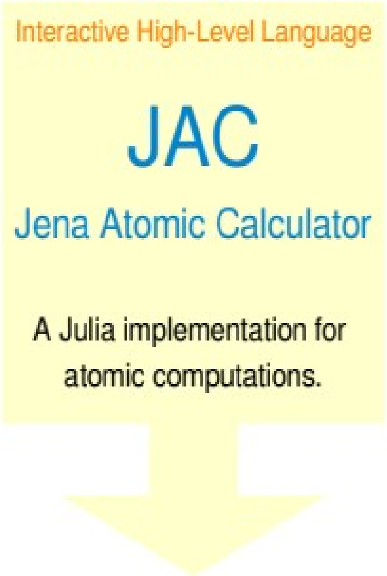
Zeeman

Electron-impact processes

isotope shifts & nuclear moments

alpha-variations

double (auto-) i


Interactive High-Level Language

JAC

Jena Atomic Calculator

A Julia implementation for
atomic computations.

electric dipole moments

electron affinities

electron capture

polarizibilities

plasm s

autoion & cascades

ato n factors

Plan of this talk

- Demands and available tools. What is needed ?
- JAC: An alternative implementation
- Atomic cascades & applications
- A first example: How to make use of the toolbox
- Need of a descriptive language: Data structures
- Summary

Established tools for atomic computations

-- including great physical insight & numerical expertise

- Cowan's HFX: support & semi-empirical adjustment of level structures, transition probabilities & cross sections.
- ATSP: Breit-Pauli approximation, level energies & properties.
- Grasp/Ratip/ MDGME: Large-scale computations of individual energies, rates, ...
- FAC: Modelling and diagnostics of astro- & plasma processes.
  Flexible Atomic Code
- CI-MBPT: Combines CI and MBPT methods for bound-state properties.
- ...
- “Home-made”: Large No. of tools for particular purposes.

→ huge number of applications in AMO physics, science & technology

Established tools for atomic computations

-- including great physical insight & numerical expertise

- Cowan's HFX: support & semi-empirical adjustment of level structures, transition probabilities & cross sections.
- ATSP: Breit-Pauli approximation, level energies f
- Grasp/Ratip/ MDGME Large-scale computatio
- FAC: Flexible Atomic C
- und-state properties.
- o. of tools for particular purposes.

Most often, Fortran (or C, C++) codes ... quite technical & with little use of the underlying 'physics language' as developed in atomic theory; difficult to extent towards new processes, coding is typically cumbersome.

Performance vs. productivity.

→ huge number of applications in AMO physics, science & technology

Established tools for atomic computations

-- including great physical insight & numerical expertise

- Cowan's HFX: support & semi-empirical adjustment of level structures, transition probabilities & cross sections.
- ATSP: Breit-Pauli approximation, level energies & properties.
- Grasp/Ratip/ MDGME Large-scale computations of individual energies, rates, ...
- FAC:

Central questions to any new implementation:

- Is a common (and community) platform for atomic computations desirable ?
- „Adhesive approach“ ? ... how to organize technically ?
- Representation of orbitals & wave functions ? ... modularity ?
- How can we benefit from a good ‘core machinery’ ?
- How simple and user-friendly can such a machinery be made ?
- How to combine productivity & performance in developing such a platform ?

What do we need in atomic structure and collision theory ?

– a descriptive language for doing atomic computations ...

Requirements:

- ▶ Data types close to atomic physics.

  Shell, Subshell, Configuration, Orbital, Basis, Level, Multiplet, Cascade, Pulse, ...

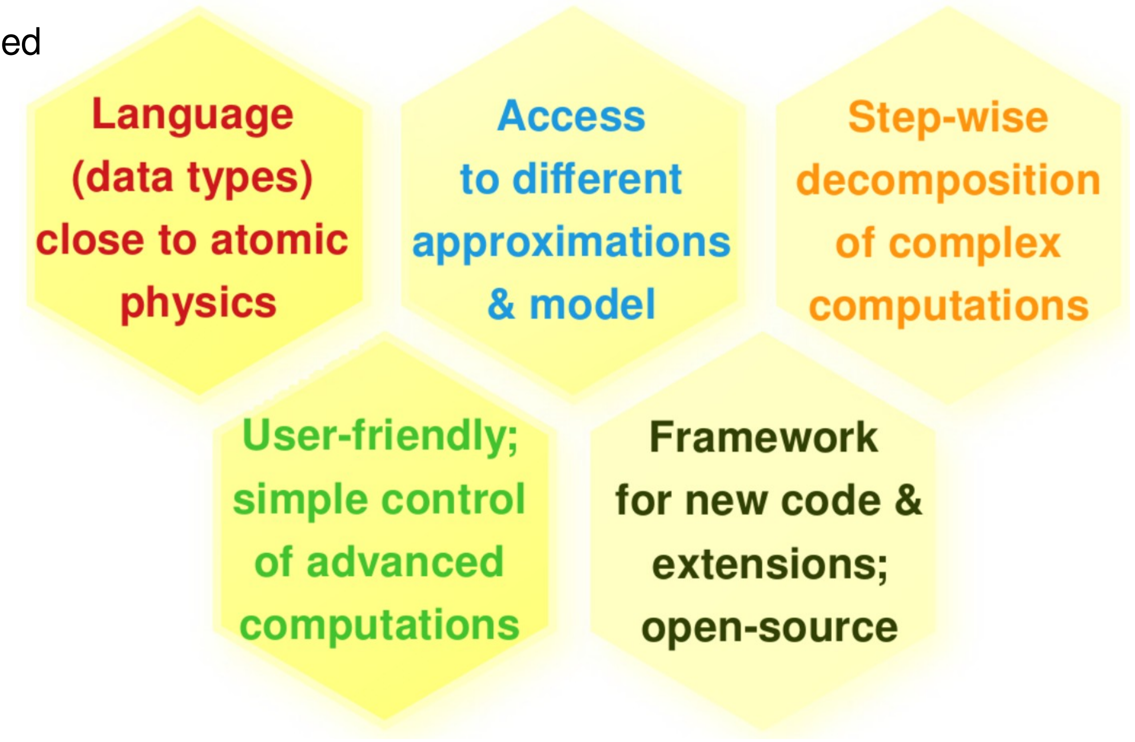
- ▶ Implementation and comparison of different models.

- ▶ Support a coarse-grained decomposition of most computational steps.

  A pseudo-code description should allow summarizing the major problem.

- ▶ Simple to learn and apply.

  With a simplified control; standard vs. advanced computations, complete active spaces; atomic cascades; ...



Language (data types) close to atomic physics

Access to different approximations & model

Step-wise decomposition of complex computations

User-friendly; simple control of advanced computations

Framework for new code & extensions; open-source

S. Fritzsche, Comput. Phys. Commun., **240** (2019) 1-14.

https://www.github.com/OpenJAC/JAC.jl

Atomic representations

- Configuration-based expansions
- Restricted active spaces (layer-by-layer)
- CI+perturbation theory; Gamov states
- Approximate Green functions, ...

Processes & properties

- Transition probabilities
- Excitation, ionization & recombination
- Auger, DR, Rayleigh-Compton, multi-γ
- Hyperfine & Zeeman splitting; plasma
- Isotope shifts, Lande & form factors

Atomic cascades

- Average singe-configuration approach
- Multiple-configuration approach
- Incorporation of shake-up & shake-off
- Ion & electron distributions, ...

Symbolic Racah algebra

- Wigner symbols, special values
- Symmetries & recursions
- Symbolic sum rule evaluation
- Spherical harmonics & tensors

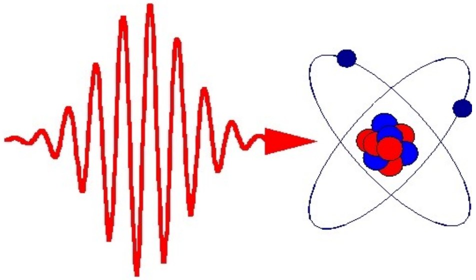
Interactive High-Level Language

JAC

Jena Atomic Calculator

A Julia implementation for atomic computations.

Open-source applications in physics, science and technology.

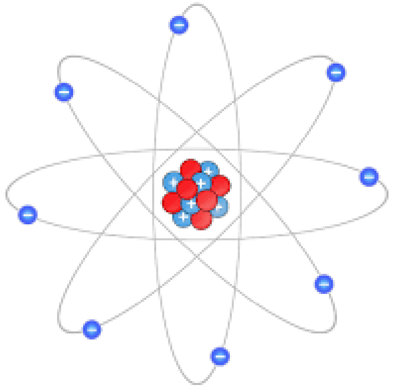


Interactive High-Level Language

JAC

Jena Atomic Calculator

A Julia implementation for
atomic computations.

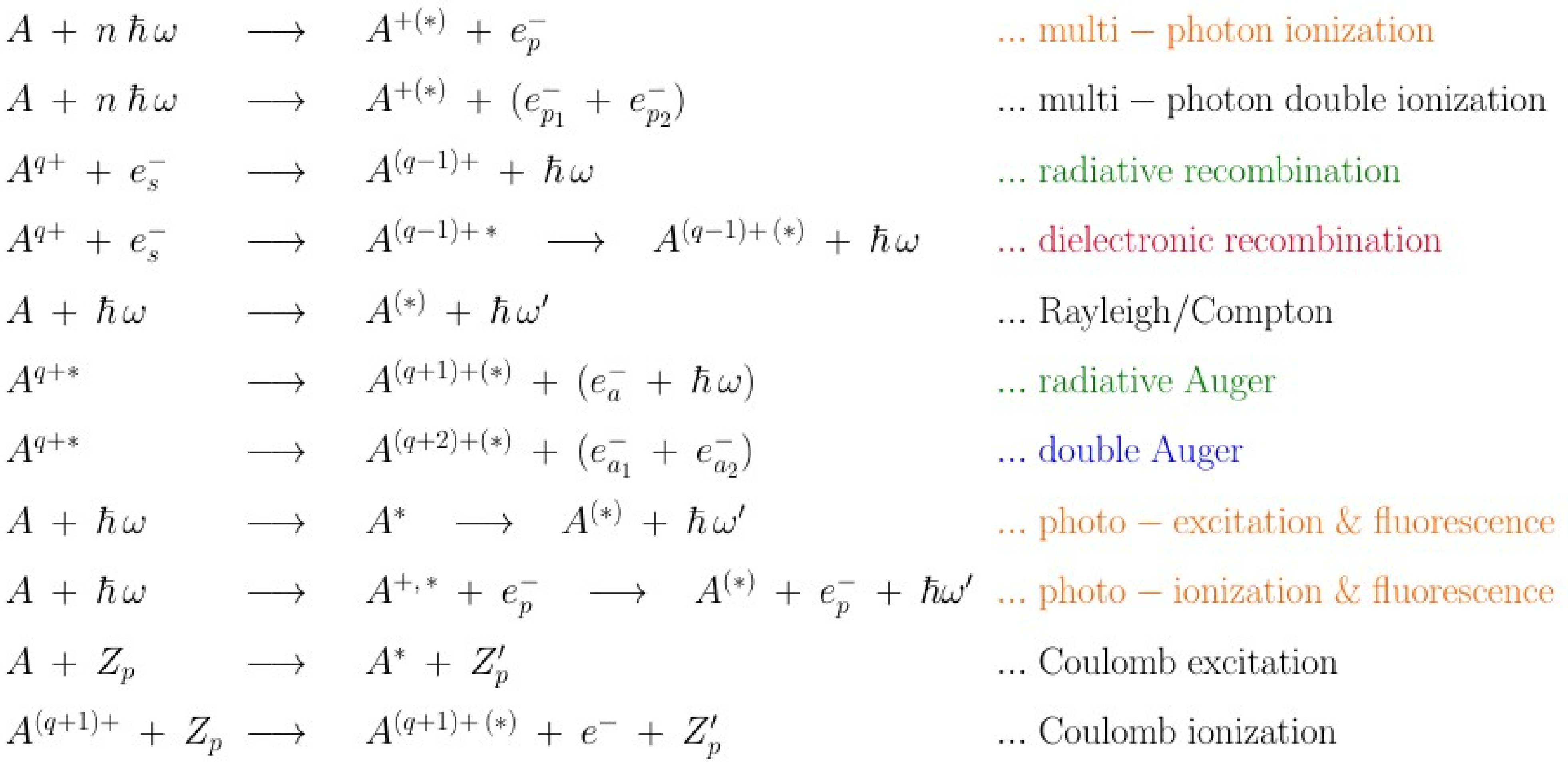


A few simple atomic processes:

A^*	$\longrightarrow$	$A^{(*)} \; + \; \hbar\omega$	... photon emission
$A \; + \; \hbar\omega$	$\longrightarrow$	A^*	... photon excitation
$A \; + \; \hbar\omega$	$\longrightarrow$	$A^{+*} \; + \; e_p^-$	... (atomic) photoionization
A^{q+*}	$\longrightarrow$	$A^{(q+1)+(*)} \; + \; e_a^-$	... Auger emission; autoionization
$e_s^- \; + \; A$	$\longrightarrow$	$A^* \; + \; e_s^{-'}$	... electron – impact excitation
$e_s^- \; + \; A$	$\longrightarrow$	$A^* \; + \; e_s^{-'} \; + \; e^-$	... electron – impact ionization
$A \; + \; n\,\hbar\omega$	$\longrightarrow$	A^*	... multi – photon excitation/decay

Quiz: Atomic processes in a nutshell

-- for “intermediates” in atomic physics



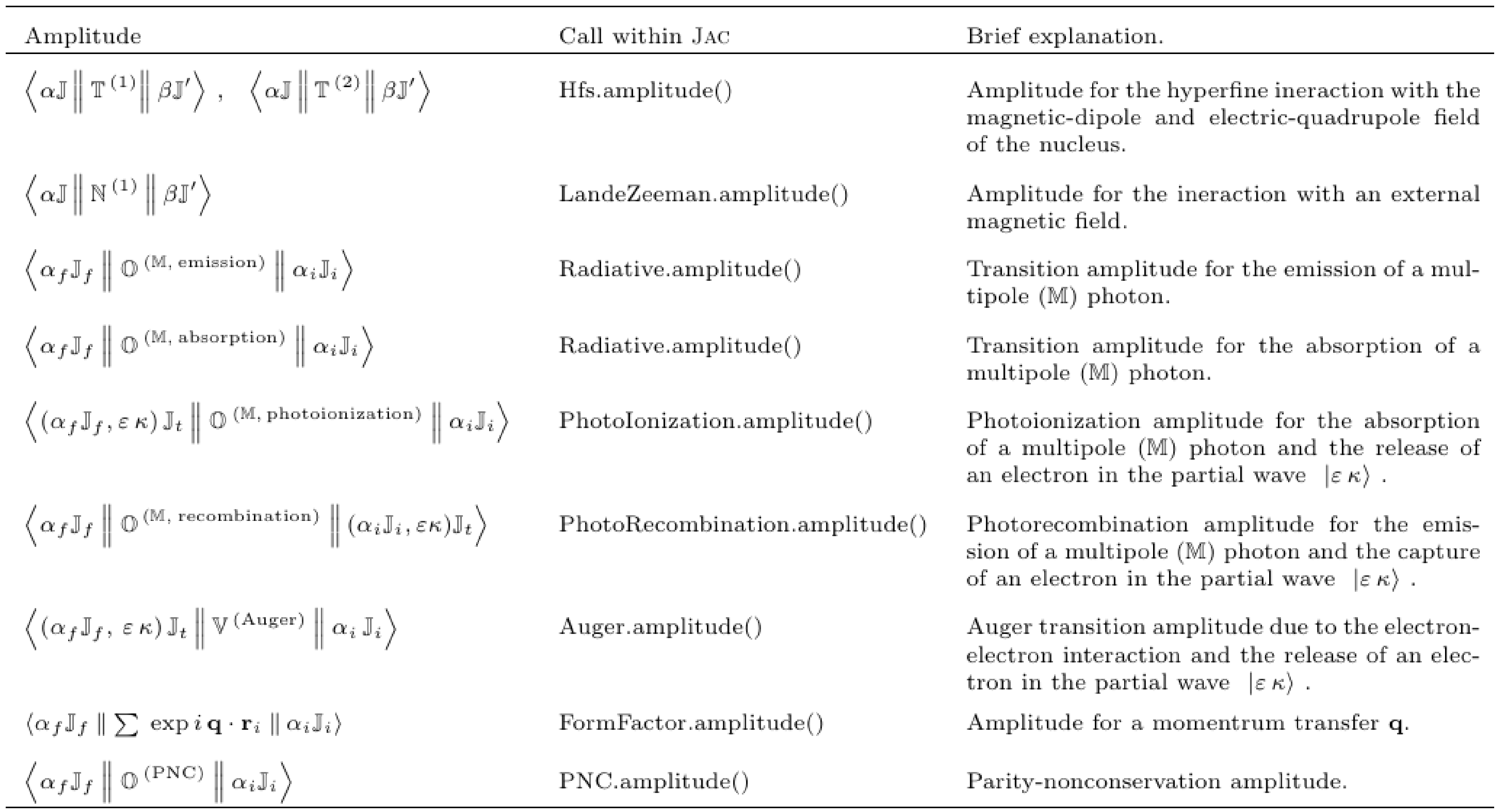
$A + n\,\hbar\omega$	$\longrightarrow$	$A^{+(*)} + e_p^-$	... multi − photon ionization
$A + n\,\hbar\omega$	$\longrightarrow$	$A^{+(*)} + (e_{p_1}^- + e_{p_2}^-)$	... multi − photon double ionization
$A^{q+} + e_s^-$	$\longrightarrow$	$A^{(q-1)+} + \hbar\omega$	... radiative recombination
$A^{q+} + e_s^-$	$\longrightarrow$	$A^{(q-1)+*} \longrightarrow A^{(q-1)+(*)} + \hbar\omega$	... dielectronic recombination
$A + \hbar\omega$	$\longrightarrow$	$A^{(*)} + \hbar\omega'$	... Rayleigh/Compton
A^{q+*}	$\longrightarrow$	$A^{(q+1)+(*)} + (e_a^- + \hbar\omega)$	... radiative Auger
A^{q+*}	$\longrightarrow$	$A^{(q+2)+(*)} + (e_{a_1}^- + e_{a_2}^-)$	... double Auger
$A + \hbar\omega$	$\longrightarrow$	$A^* \longrightarrow A^{(*)} + \hbar\omega'$	... photo − excitation & fluorescence
$A + \hbar\omega$	$\longrightarrow$	$A^{+,*} + e_p^- \longrightarrow A^{(*)} + e_p^- + \hbar\omega'$	... photo − ionization & fluorescence
$A + Z_p$	$\longrightarrow$	$A^* + Z'_p$	... Coulomb excitation
$A^{(q+1)+} + Z_p$	$\longrightarrow$	$A^{(q+1)+(*)} + e^- + Z'_p$	... Coulomb ionization

- Indeed, these and many other processes occur in atomic spectroscopy, astro and plasma physics as well as at various places elsewhere.
- How much help can atomic theory provide ? -- Which tools are available ?

https://www.github.com/OpenJAC/JAC.jl

Jena Atomic Calculator (JAC)

– atomic **amplitudes** implemented

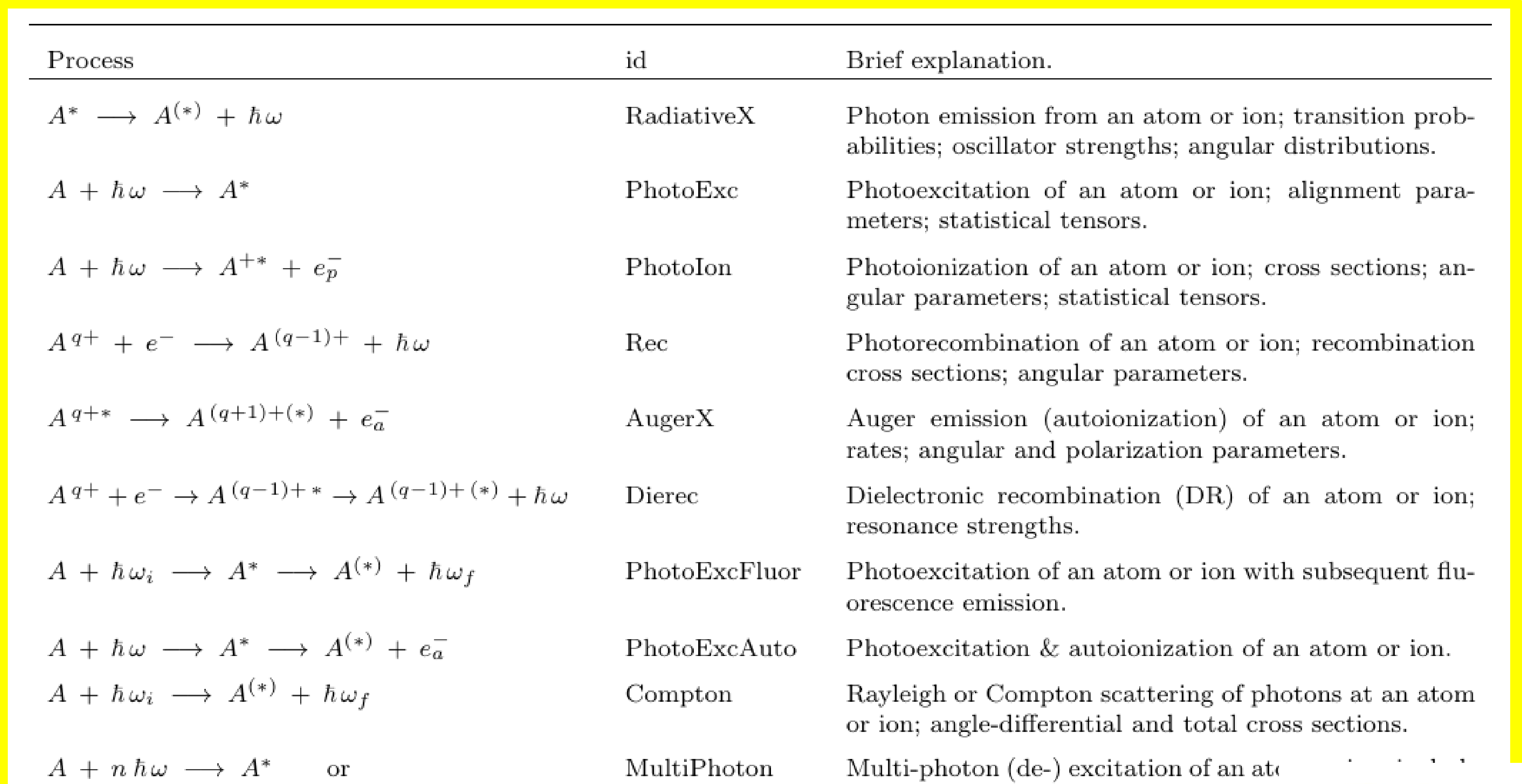


Amplitude	Call within Jac	Brief explanation.
$\left\langle \alpha\mathbb{J} \left\| \mathbb{T}^{(1)} \right\| \beta\mathbb{J}' \right\rangle$, $\left\langle \alpha\mathbb{J} \left\| \mathbb{T}^{(2)} \right\| \beta\mathbb{J}' \right\rangle$	Hfs.amplitude()	Amplitude for the hyperfine ineraction with the magnetic-dipole and electric-quadrupole field of the nucleus.
$\left\langle \alpha\mathbb{J} \left\| \mathbb{N}^{(1)} \right\| \beta\mathbb{J}' \right\rangle$	LandeZeeman.amplitude()	Amplitude for the ineraction with an external magnetic field.
$\left\langle \alpha_f\mathbb{J}_f \left\| \mathbb{O}^{(\mathbb{M},\text{ emission})} \right\| \alpha_i\mathbb{J}_i \right\rangle$	Radiative.amplitude()	Transition amplitude for the emission of a multipole ($\mathbb{M}$) photon.
$\left\langle \alpha_f\mathbb{J}_f \left\| \mathbb{O}^{(\mathbb{M},\text{ absorption})} \right\| \alpha_i\mathbb{J}_i \right\rangle$	Radiative.amplitude()	Transition amplitude for the absorption of a multipole ($\mathbb{M}$) photon.
$\left\langle (\alpha_f\mathbb{J}_f, \varepsilon\,\kappa)\,\mathbb{J}_t \left\| \mathbb{O}^{(\mathbb{M},\text{ photoionization})} \right\| \alpha_i\mathbb{J}_i \right\rangle$	PhotoIonization.amplitude()	Photoionization amplitude for the absorption of a multipole ($\mathbb{M}$) photon and the release of an electron in the partial wave $\lvert\varepsilon\,\kappa\rangle$.
$\left\langle \alpha_f\mathbb{J}_f \left\| \mathbb{O}^{(\mathbb{M},\text{ recombination})} \right\| (\alpha_i\mathbb{J}_i, \varepsilon\kappa)\mathbb{J}_t \right\rangle$	PhotoRecombination.amplitude()	Photorecombination amplitude for the emission of a multipole ($\mathbb{M}$) photon and the capture of an electron in the partial wave $\lvert\varepsilon\,\kappa\rangle$.
$\left\langle (\alpha_f\mathbb{J}_f,\ \varepsilon\,\kappa)\,\mathbb{J}_t \left\| \mathbb{V}^{(\text{Auger})} \right\| \alpha_i\ \mathbb{J}_i \right\rangle$	Auger.amplitude()	Auger transition amplitude due to the electron-electron interaction and the release of an electron in the partial wave $\lvert\varepsilon\,\kappa\rangle$.
$\langle\alpha_f\mathbb{J}_f \,\|\, \sum\ \exp i\,\mathbf{q}\cdot\mathbf{r}_i \,\|\, \alpha_i\mathbb{J}_i\rangle$	FormFactor.amplitude()	Amplitude for a momentrum transfer $\mathbf{q}$.
$\left\langle \alpha_f\mathbb{J}_f \left\| \mathbb{O}^{(\text{PNC})} \right\| \alpha_i\mathbb{J}_i \right\rangle$	PNC.amplitude()	Parity-nonconservation amplitude.

https://www.github.com/OpenJAC/JAC.jl

Atomic **processes** in JAC

– combining often (bound) levels with a different No. of electrons



Process	id	Brief explanation.
$A^* \longrightarrow A^{(*)} + \hbar\omega$	RadiativeX	Photon emission from an atom or ion; transition probabilities; oscillator strengths; angular distributions.
$A + \hbar\omega \longrightarrow A^*$	PhotoExc	Photoexcitation of an atom or ion; alignment parameters; statistical tensors.
$A + \hbar\omega \longrightarrow A^{+*} + e_p^-$	PhotoIon	Photoionization of an atom or ion; cross sections; angular parameters; statistical tensors.
$A^{q+} + e^- \longrightarrow A^{(q-1)+} + \hbar\omega$	Rec	Photorecombination of an atom or ion; recombination cross sections; angular parameters.
$A^{q+*} \longrightarrow A^{(q+1)+(*)} + e_a^-$	AugerX	Auger emission (autoionization) of an atom or ion; rates; angular and polarization parameters.
$A^{q+} + e^- \to A^{(q-1)+*} \to A^{(q-1)+(*)} + \hbar\omega$	Dierec	Dielectronic recombination (DR) of an atom or ion; resonance strengths.
$A + \hbar\omega_i \longrightarrow A^* \longrightarrow A^{(*)} + \hbar\omega_f$	PhotoExcFluor	Photoexcitation of an atom or ion with subsequent fluorescence emission.
$A + \hbar\omega \longrightarrow A^* \longrightarrow A^{(*)} + e_a^-$	PhotoExcAuto	Photoexcitation & autoionization of an atom or ion.
$A + \hbar\omega_i \longrightarrow A^{(*)} + \hbar\omega_f$	Compton	Rayleigh or Compton scattering of photons at an atom or ion; angle-differential and total cross sections.
$A + n\,\hbar\omega \longrightarrow A^*$ or $A^* \longrightarrow A^* + n\,\hbar\omega$	MultiPhoton	Multi-photon (de-) excitation of an at ing two-photon decay, etc.
$A + Z_p \longrightarrow A^* + Z_p$	CoulExc	Coulomb excitation of an atom or ic ions; energie-differential, partial and t citation cross sections.

There are many other atomic processes and related entities.

https://www.github.com/OpenJAC/JAC.jl

Atomic representations

- Configuration-based expansions
- Restricted active spaces (layer-by-layer)
- CI+perturbation theory; Gamov states
- Approximate Green functions, ...

Processes & properties

- Transition probabilities
- Excitation, ionization & recombination
- Auger, DR, Rayleigh-Compton, multi-γ
- Hyperfine & Zeeman splitting; plasma
- Isotope shifts, Lande & form factors

Atomic cascades

- Average singe-configuration approach
- Multiple-configuration approach
- Incorporation of shake-up & shake-off
- Ion & electron distributions, ...

Symbolic Racah algebra

- Wigner symbols, special values
- Symmetries & recursions
- Symbolic sum rule evaluation
- Spherical harmonics & tensors

Interactive High-Level Language

JAC

Jena Atomic Calculator

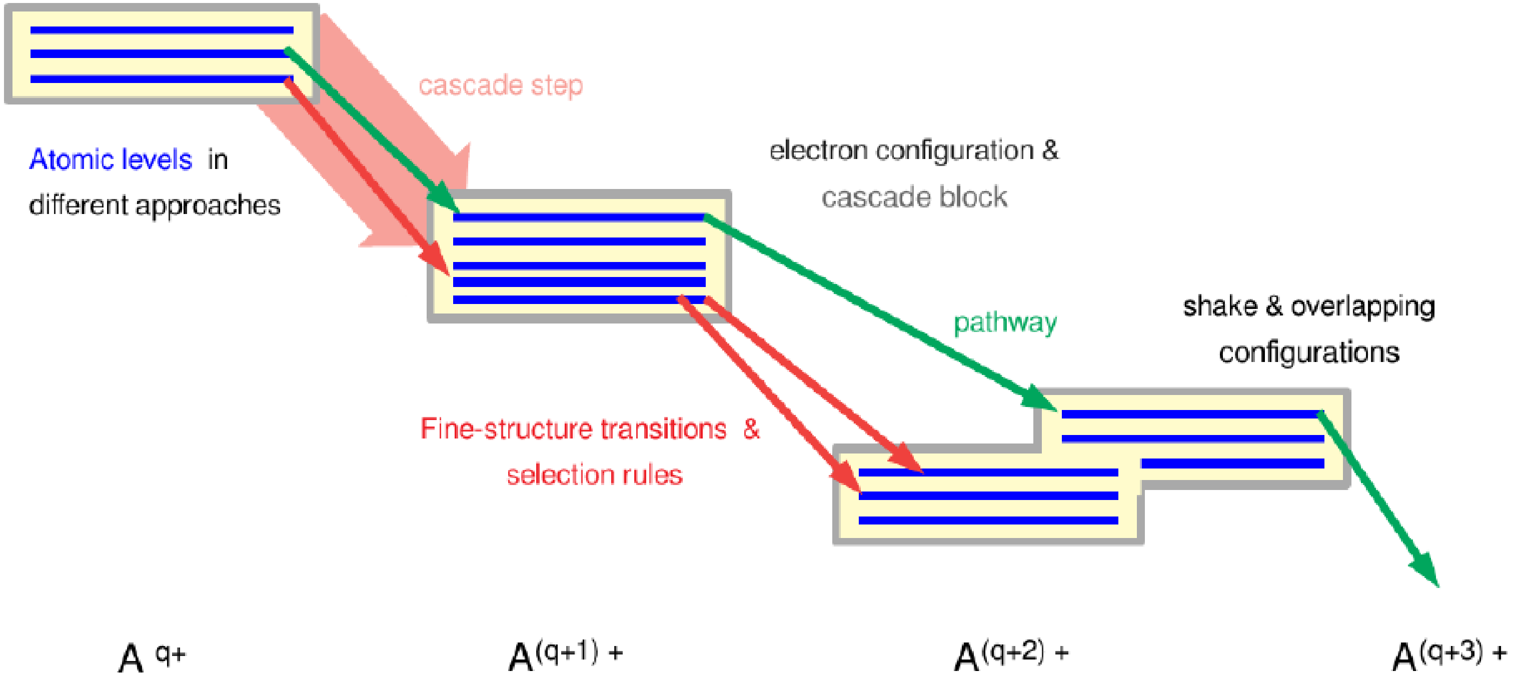
A Julia implementation for atomic computations.

Open-source applications in physics, science and technology.

SF, P. Palmeri & S. Schippers, Symmetry **13** (2021) 520.

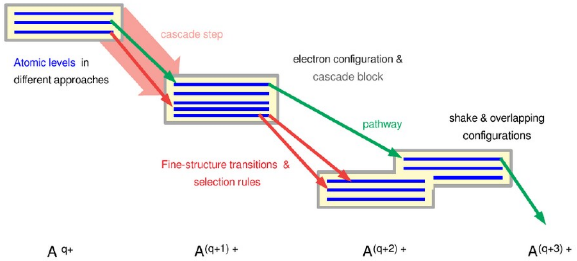
Atomic casades

– key to many spectroscopic observations

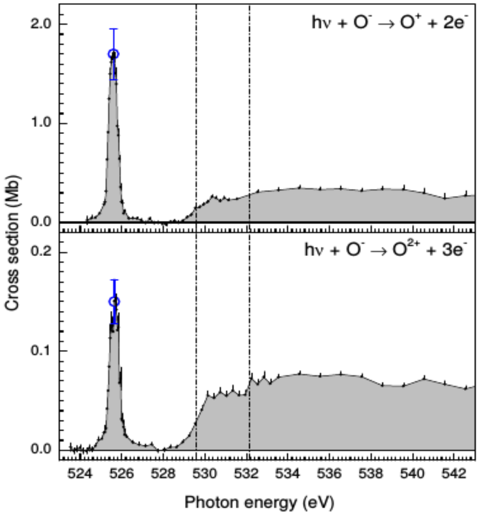


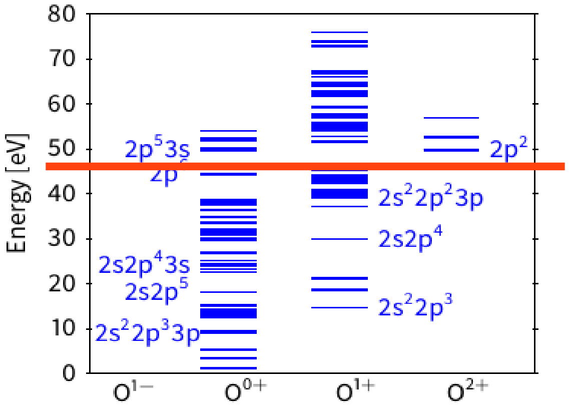
Ion & absorptoin spectra after inner-shell excitations

– recorded at PIPE (Hamburg)

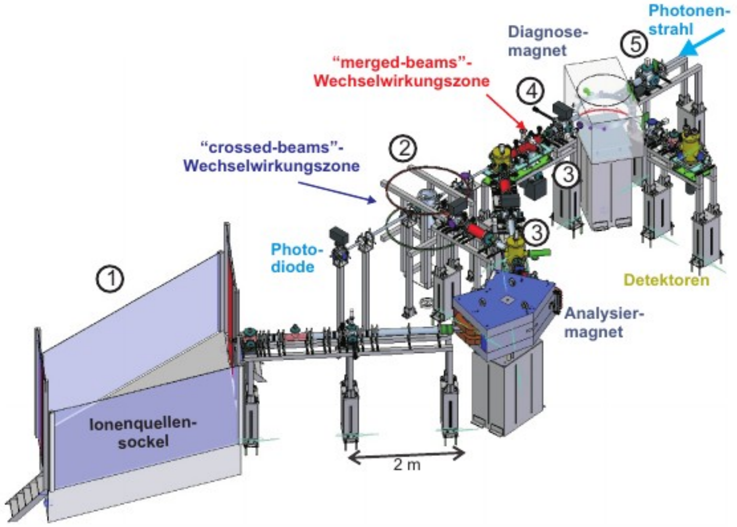

cascade step
Atomic levels in different approaches
electron configuration & cascade block
pathway
shake & overlapping configurations
Fine-structure transitions & selection rules
A q+
A(q+1) +
A(q+2) +
A(q+3) +

$$O^- (1s^2\, 2s^2\, 2p^5) \;+\; \gamma \;\rightarrow\; O^-(1s\, 2s^2\, 2p^6\;\, ^2S_{1/2}) \;\rightarrow\; O^{(m-1)+} (1s^2\, 2l^{7-m}) \;+ m\, e^-$$





merge-beam experiments



Double & triple detachments by Auger electron emission are energetically forbidden.

→ possible only due to shake-up

2s → 3s, 2p → 3p

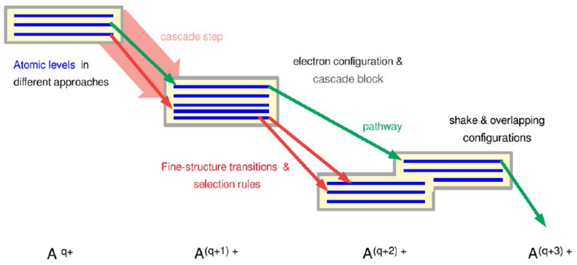
S. Schippers et al, PRA 94 (2016) 052412.

PIPE – Photon-Ion spectrometer at PETRA III (Hamburg)

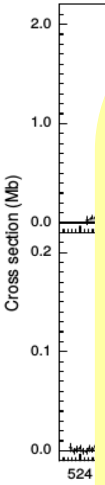
S. Schippers and coworkers, Gießen, Frankfurt, Hamburg collaboration (since 2014).

Ion & absorptoin spectra after inner-shell excitations

– recorded at PIPE (Hamburg)


cascade step
Atomic levels in different approaches
electron configuration & cascade block
pathway
shake & overlapping configurations
Fine-structure transitions & selection rules
A q+
A(q+1) +
A(q+2) +
A(q+3) +

$$O^- (1s^2\, 2s^2\, 2p^5) + \gamma \rightarrow O^-(1s\, 2s^2\, 2p^6\ ^2S_{1/2}) \rightarrow O^{(m-1)+} (1s^2\, 2l^{7-m}) + m\, e^-$$



Dou

emis

S. S

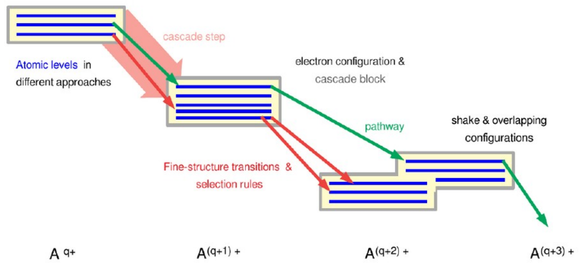
14).

Recent applications

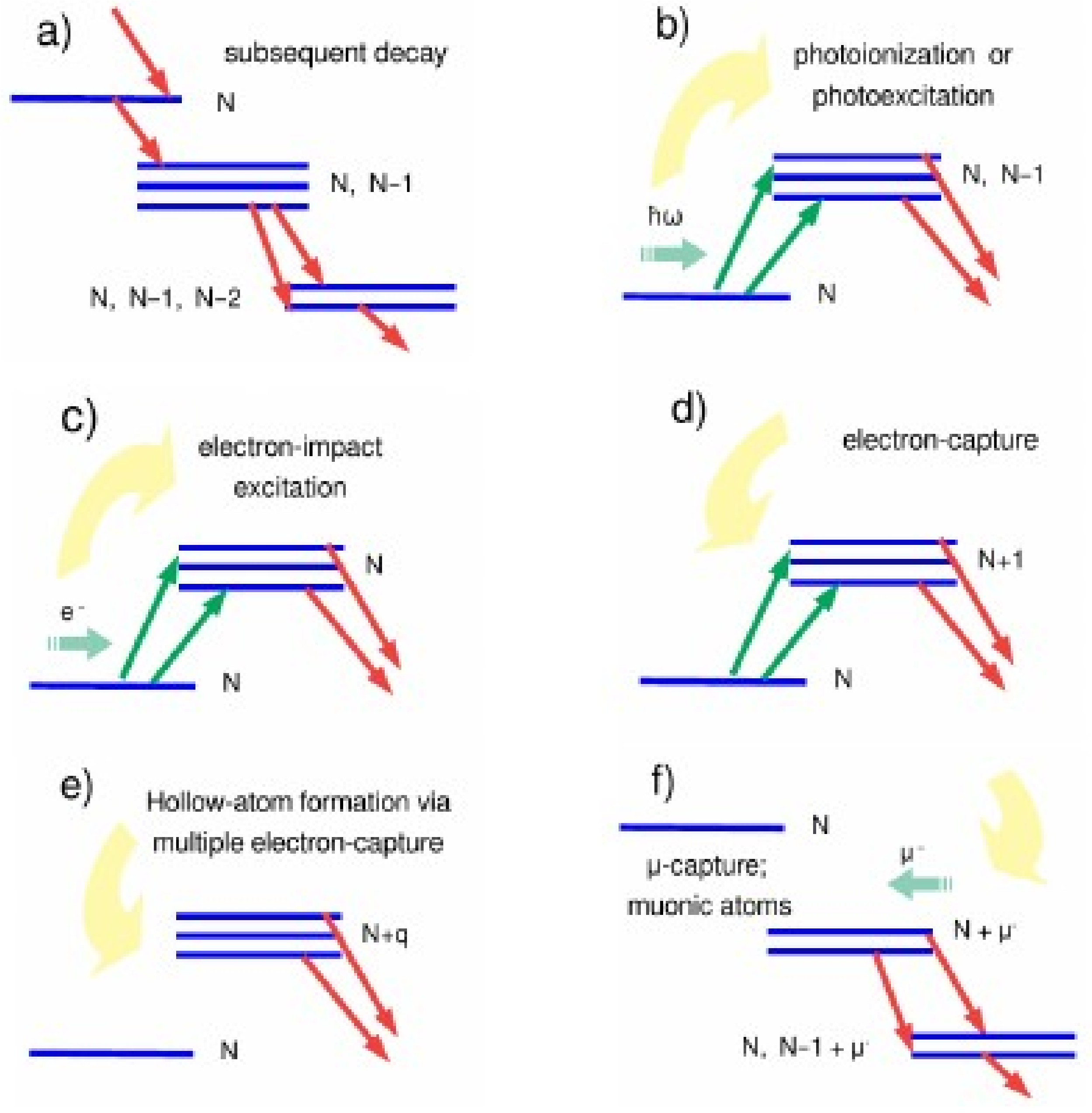
- Near L-edge photoionization of Fe^+: Schippers *et al.*, ApJ 849 (2017) 5.
- Near L-edge photoionization of Fe^{3+}: Beerwerth *et al.*, ApJ 887 (2019) 189.
- Multiple photodetachment of C^-: Perry-Sassmannshausen *et al.*, PRL 124 (2020) 083203.
- 3p photoelectron spectrum of Kr: Kosogi *et al.*, PRA 101 (2020) 042505.
- Multiple photodetachment of atomic anions: Schippers *et al.*, JPB 53 (2021) 192001.
- Near L-edge photoionization of Fe^{2+}: Schippers *et al.*, ApJ 908 (2021) 52.
- L-shell single/double core-hole production of Ar^+: Müller *et al.*, PRA 104 (2021) 042505.
- Multiple photodetachment of Si^-: Sassmannshausen *et al.*, PRA 104 (2021) 053101.
- ...

SF, P. Palmeri & S. Schippers, Symmetry **13** (2021) 520.

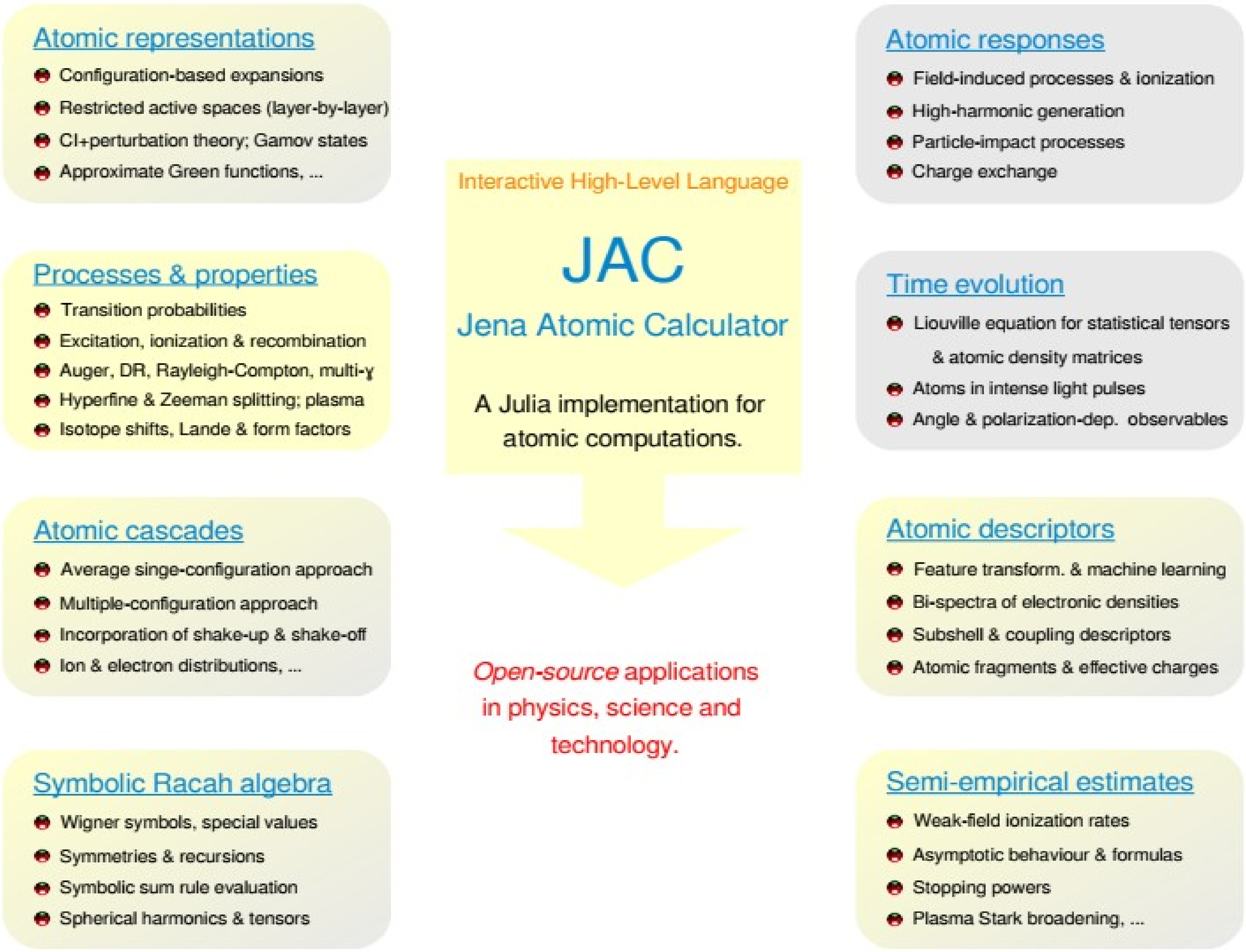
Atomic casades



– key to many spectroscopic observations



plasma processes &
rate coefficients



Atomic representations

- Configuration-based expansions
- Restricted active spaces (layer-by-layer)
- CI+perturbation theory; Gamov states
- Approximate Green functions, ...

Processes & properties

- Transition probabilities
- Excitation, ionization & recombination
- Auger, DR, Rayleigh-Compton, multi-γ
- Hyperfine & Zeeman splitting; plasma
- Isotope shifts, Lande & form factors

Atomic cascades

- Average singe-configuration approach
- Multiple-configuration approach
- Incorporation of shake-up & shake-off
- Ion & electron distributions, ...

Symbolic Racah algebra

- Wigner symbols, special values
- Symmetries & recursions
- Symbolic sum rule evaluation
- Spherical harmonics & tensors

Interactive High-Level Language

JAC

Jena Atomic Calculator

A Julia implementation for atomic computations.

Open-source applications in physics, science and technology.

Atomic responses

- Field-induced processes & ionization
- High-harmonic generation
- Particle-impact processes
- Charge exchange

Time evolution

- Liouville equation for statistical tensors & atomic density matrices
- Atoms in intense light pulses
- Angle & polarization-dep. observables

Atomic descriptors

- Feature transform. & machine learning
- Bi-spectra of electronic densities
- Subshell & coupling descriptors
- Atomic fragments & effective charges

Semi-empirical estimates

- Weak-field ionization rates
- Asymptotic behaviour & formulas
- Stopping powers
- Plasma Stark broadening, ...

1

?

https://www.github.com/OpenJAC/JAC.jl

https://www.github.com/OpenJAC/JAC.jl

Jena Atomic Calculator (JAC)

-- A first example and encounter

JAC ... Jena atomic calculator provides tools for performing atomic (structure) calculations at various degrees of complexity and sophistication. ... JAC also facilitates interactive computations, the simulation of atomic cascades, the time-evolution of statistical tensors as well as various semi-empirical estimates of atomic properties. In addition, the Jac module supports the graphical representation of level energies, electron and photon spectra, radial orbitals and others.

Example: Einstein A and B coefficients for the Fe X spectrum;

$$\mathrm{Fe}^{9+} \quad [\mathrm{Ne}]\ 3s^2\ 3p^5 \rightarrow [\mathrm{Ne}]\ 3s\ 3p^6 \ + \ 3s^2\ 3p^4\ 3d$$

```
> wa = Atomic.Computation("Fe X: Einstein",  NuclearModel(26.), ...,
            [Configuration("[Ne] 3s^2 p^5")], ...,
            [Configuration("[Ne ] 3s 3p^6"), Configuration("[Ne] 3s^2 3p^4 3d") ], ...,
            Radiative.Settings([E1, M2], [UseCoulomb, UseBabushkin], false, false, ...  )
> perform(wa)
```

```
... in perform('computation: SCF', ...)
Compute CI matrix of dimension 1 x 1 for the symmetry 1/2^+ ...   done.
Compute CI matrix of dimension 1 x 1 for the symmetry 3/2^+ ...   done.
...
```

https://www.github.com/OpenJAC/JAC.jl

Jena Atomic Calculator (JAC)

-- A first example and encounter

JAC ... Jena atomic calculator provides tools for performing atomic (structure) calculations at various degrees of complexity and sophistication. ... JAC also facilitates interactive computations, the simulation of atomic cascades, the time-evolution of statistical tensors as well as various semi-empirical estimates of atomic properties. In addition, the Jac module supports the graphical representation of level energies, electron and photon spectra, radial orbitals and others.

Example: Einstein A and B coefficients for the Fe X spectrum;

$$\mathrm{Fe}^{9+} \quad [\mathrm{Ne}]\, 3s^2\, 3p^5 \rightarrow [\mathrm{Ne}]\, 3s\, 3p^6 \;+\; 3s^2\, 3p^4\, 3d$$

- Generation of start orbitals
- Computation of angular coefficients (on fly)
- Self-Consistent-Field (SCF) iteration
- Set-up and diagonalization of Hamiltonian matrix
- Breit, QED, many-body corrections, ...
- Compute all (many-electron) transition amplitudes

Treated often individually (GRASP , FAC, ...)

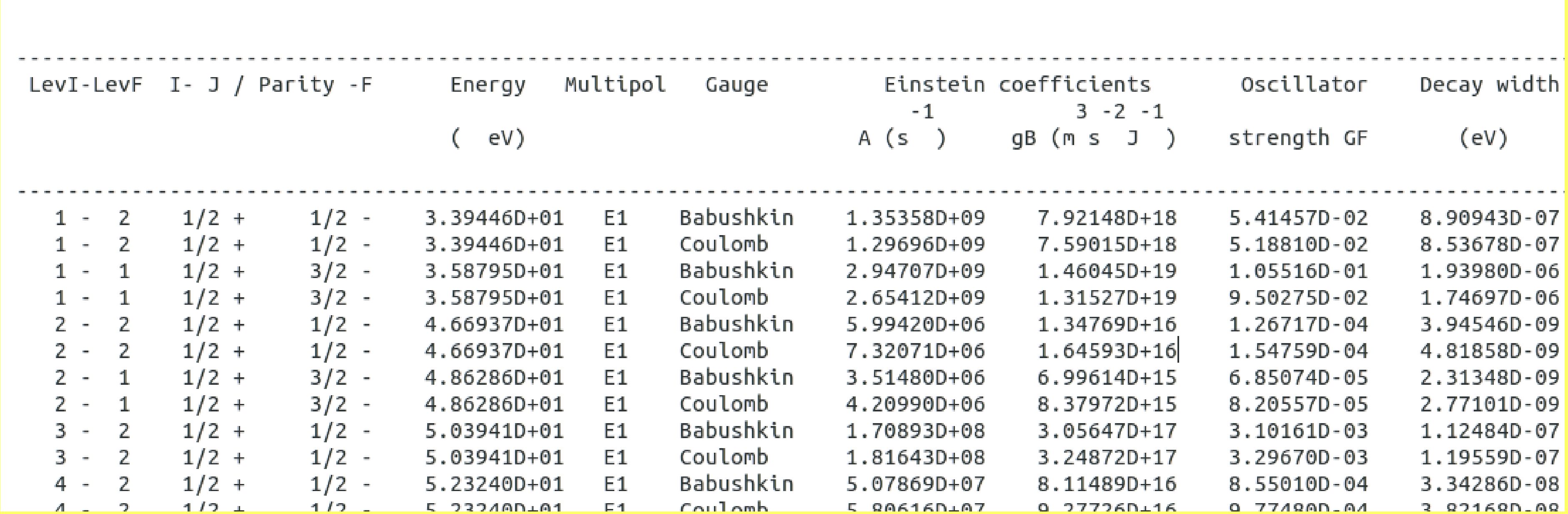
https://www.github.com/OpenJAC/JAC.jl

Jena Atomic Calculator (JAC)

-- A first example and encounter

JAC ... Jena atomic calculator provides tools for performing atomic (structure) calculations at various degrees of complexity and sophistication. ... JAC also facilitates interactive computations, the simulation of atomic cascades, the time-evolution of statistical tensors as well as various semi-empirical estimates of atomic properties. In addition, the Jac module supports the graphical representation of level energies, electron and photon spectra, radial orbitals and others.

Example: Einstein A and B coefficients for the Fe X spectrum;



```
 ---------------------------------------------------------------------------------------------------------------------
  LevI-LevF  I- J / Parity -F      Energy   Multipol   Gauge          Einstein coefficients        Oscillator   Decay width
                                                                        -1             3 -2 -1
                                   (  eV)                             A (s  )      gB (m s  J  )  strength GF      (eV)

 ---------------------------------------------------------------------------------------------------------------------
    1 -  2    1/2 +     1/2 -   3.39446D+01   E1     Babushkin    1.35358D+09    7.92148D+18    5.41457D-02    8.90943D-07
    1 -  2    1/2 +     1/2 -   3.39446D+01   E1     Coulomb      1.29696D+09    7.59015D+18    5.18810D-02    8.53678D-07
    1 -  1    1/2 +     3/2 -   3.58795D+01   E1     Babushkin    2.94707D+09    1.46045D+19    1.05516D-01    1.93980D-06
    1 -  1    1/2 +     3/2 -   3.58795D+01   E1     Coulomb      2.65412D+09    1.31527D+19    9.50275D-02    1.74697D-06
    2 -  2    1/2 +     1/2 -   4.66937D+01   E1     Babushkin    5.99420D+06    1.34769D+16    1.26717D-04    3.94546D-09
    2 -  2    1/2 +     1/2 -   4.66937D+01   E1     Coulomb      7.32071D+06    1.64593D+16    1.54759D-04    4.81858D-09
    2 -  1    1/2 +     3/2 -   4.86286D+01   E1     Babushkin    3.51480D+06    6.99614D+15    6.85074D-05    2.31348D-09
    2 -  1    1/2 +     3/2 -   4.86286D+01   E1     Coulomb      4.20990D+06    8.37972D+15    8.20557D-05    2.77101D-09
    3 -  2    1/2 +     1/2 -   5.03941D+01   E1     Babushkin    1.70893D+08    3.05647D+17    3.10161D-03    1.12484D-07
    3 -  2    1/2 +     1/2 -   5.03941D+01   E1     Coulomb      1.81643D+08    3.24872D+17    3.29670D-03    1.19559D-07
    4 -  2    1/2 +     1/2 -   5.23240D+01   E1     Babushkin    5.07869D+07    8.11489D+16    8.55010D-04    3.34286D-08
    4 -  2    1/2 +     1/2 -   5.23240D+01   E1     Coulomb      5.80616D+07    9.27726D+16    9.77480D-04    3.82168D-08
```

What do we need in atomic structure and collision theory ?

– a descriptive language for doing atomic computations ...

Requirements:

- **Data types close to atomic physics.**

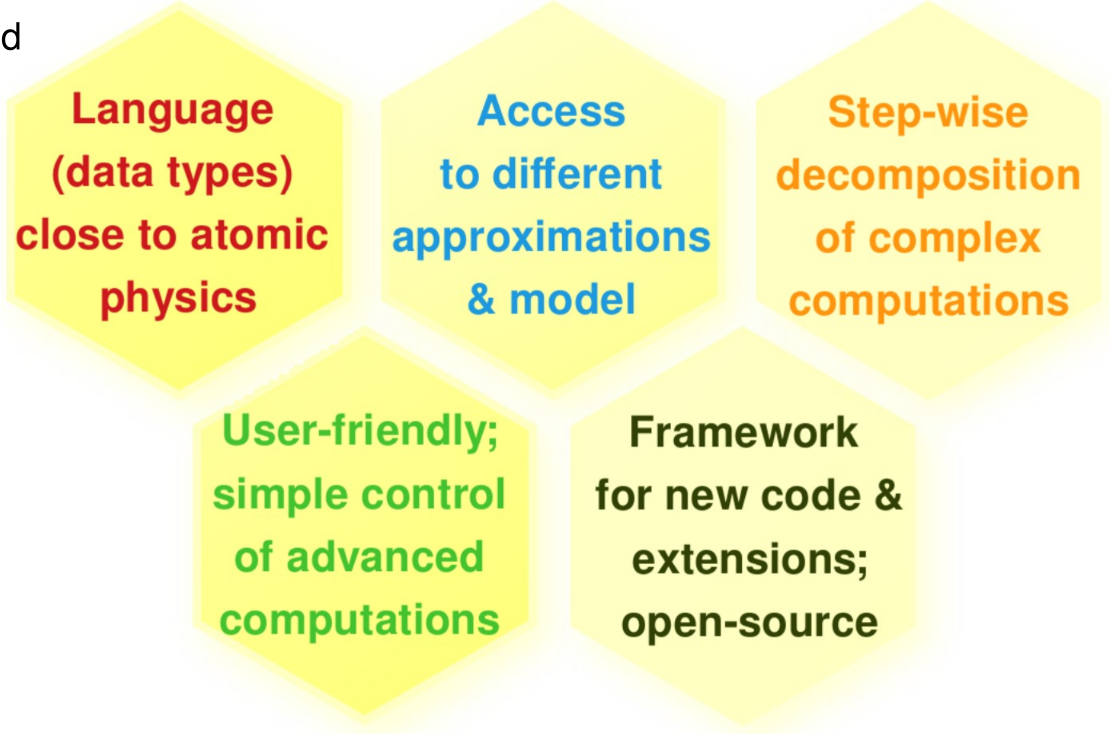
  Shell, Subshell, Configuration, Orbital, Basis, Level, Multiplet, Cascade, Pulse, ...

- Implementation and comparison of different models.
- Support a coarse-grained decomposition of most computational steps.

  A pseudo-code description should allow summarizing the major problem.

- Simple to learn and apply.

  With a simplified control; standard vs. advanced computations, complete active spaces; atomic cascades; ...



Language (data types) close to atomic physics

Access to different approximations & model

Step-wise decomposition of complex computations

User-friendly; simple control of advanced computations

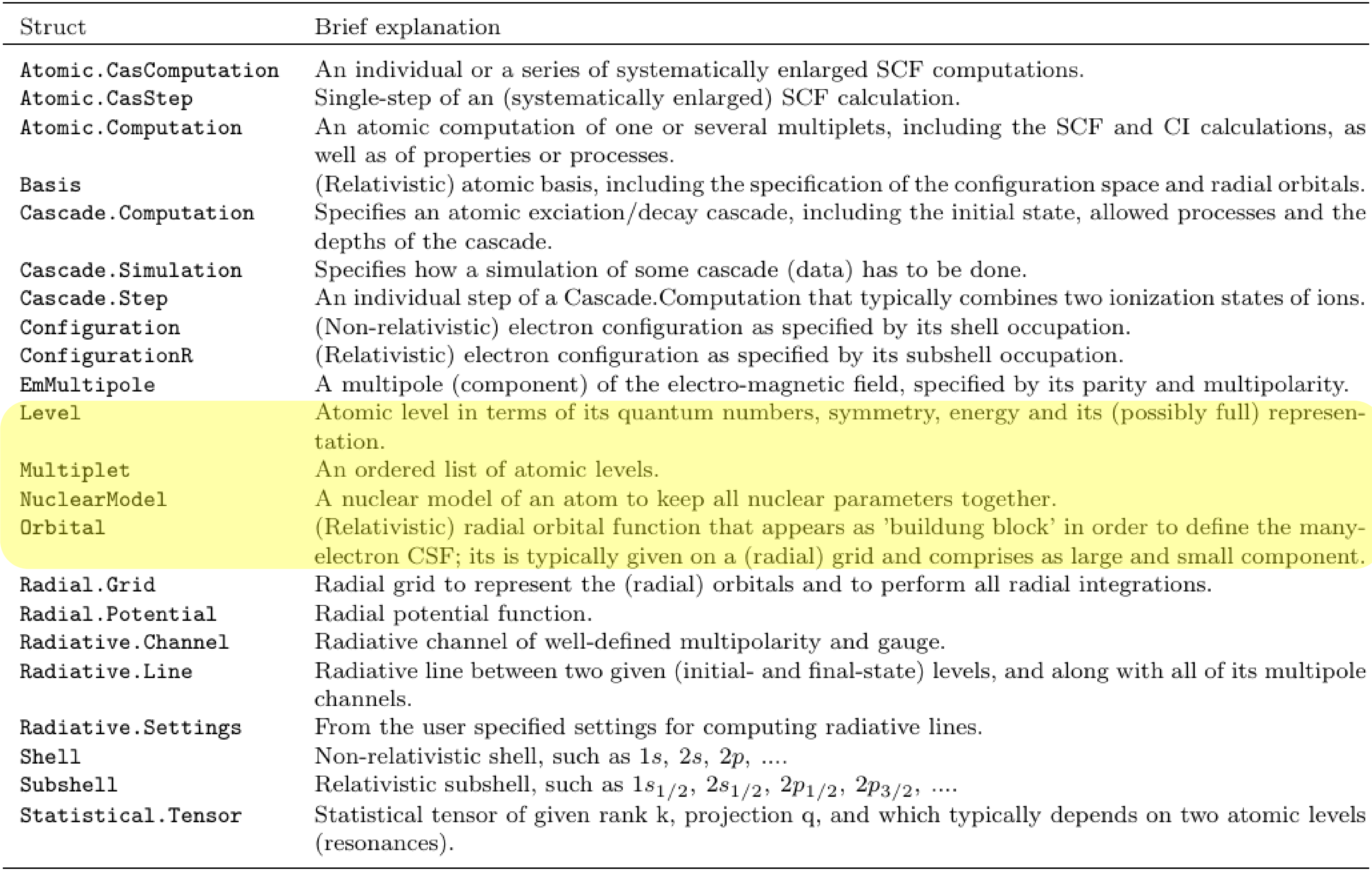
Framework for new code & extensions; open-source

S. Fritzsche, Comput. Phys. Commun., **240** (2019) 1-14.

https://www.github.com/OpenJAC/JAC.jl

Jena Atomic Calculator (JAC)

– a descriptive language for doing atomic computations ...



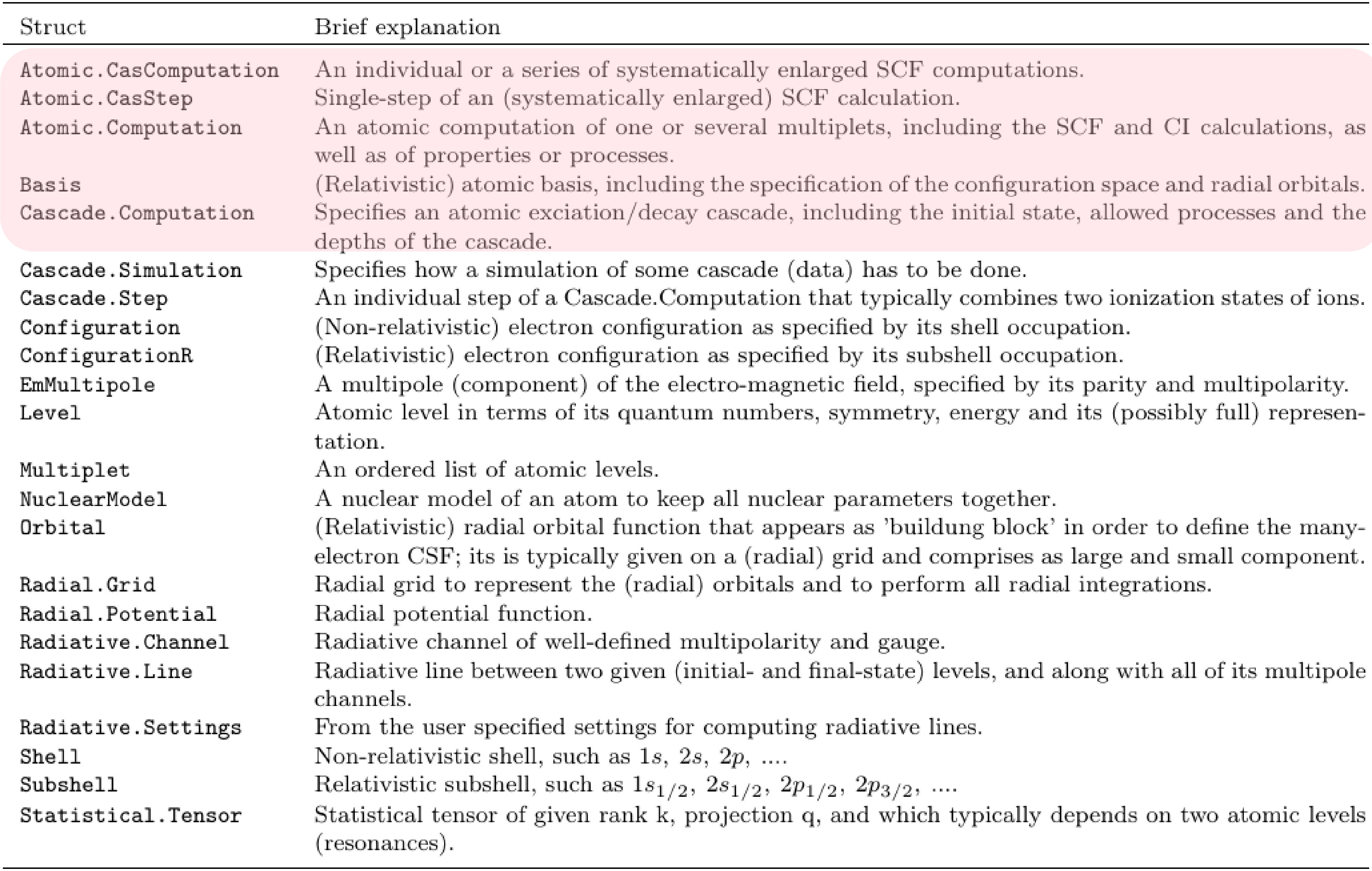
Struct	Brief explanation
`Atomic.CasComputation`	An individual or a series of systematically enlarged SCF computations.
`Atomic.CasStep`	Single-step of an (systematically enlarged) SCF calculation.
`Atomic.Computation`	An atomic computation of one or several multiplets, including the SCF and CI calculations, as well as of properties or processes.
`Basis`	(Relativistic) atomic basis, including the specification of the configuration space and radial orbitals.
`Cascade.Computation`	Specifies an atomic exciation/decay cascade, including the initial state, allowed processes and the depths of the cascade.
`Cascade.Simulation`	Specifies how a simulation of some cascade (data) has to be done.
`Cascade.Step`	An individual step of a Cascade.Computation that typically combines two ionization states of ions.
`Configuration`	(Non-relativistic) electron configuration as specified by its shell occupation.
`ConfigurationR`	(Relativistic) electron configuration as specified by its subshell occupation.
`EmMultipole`	A multipole (component) of the electro-magnetic field, specified by its parity and multipolarity.
`Level`	Atomic level in terms of its quantum numbers, symmetry, energy and its (possibly full) representation.
`Multiplet`	An ordered list of atomic levels.
`NuclearModel`	A nuclear model of an atom to keep all nuclear parameters together.
`Orbital`	(Relativistic) radial orbital function that appears as 'buildung block' in order to define the many-electron CSF; its is typically given on a (radial) grid and comprises as large and small component.
`Radial.Grid`	Radial grid to represent the (radial) orbitals and to perform all radial integrations.
`Radial.Potential`	Radial potential function.
`Radiative.Channel`	Radiative channel of well-defined multipolarity and gauge.
`Radiative.Line`	Radiative line between two given (initial- and final-state) levels, and along with all of its multipole channels.
`Radiative.Settings`	From the user specified settings for computing radiative lines.
`Shell`	Non-relativistic shell, such as $1s$, $2s$, $2p$, ...
`Subshell`	Relativistic subshell, such as $1s_{1/2}$, $2s_{1/2}$, $2p_{1/2}$, $2p_{3/2}$, ...
`Statistical.Tensor`	Statistical tensor of given rank k, projection q, and which typically depends on two atomic levels (resonances).

S. Fritzsche, Comput. Phys. Commun., **240** (2019) 1-14.

https://www.github.com/OpenJAC/JAC.jl

Jena Atomic Calculator (JAC)

-- A fresh approach to the computation of atoms, ...



Struct	Brief explanation
`Atomic.CasComputation`	An individual or a series of systematically enlarged SCF computations.
`Atomic.CasStep`	Single-step of an (systematically enlarged) SCF calculation.
`Atomic.Computation`	An atomic computation of one or several multiplets, including the SCF and CI calculations, as well as of properties or processes.
`Basis`	(Relativistic) atomic basis, including the specification of the configuration space and radial orbitals.
`Cascade.Computation`	Specifies an atomic exciation/decay cascade, including the initial state, allowed processes and the depths of the cascade.
`Cascade.Simulation`	Specifies how a simulation of some cascade (data) has to be done.
`Cascade.Step`	An individual step of a Cascade.Computation that typically combines two ionization states of ions.
`Configuration`	(Non-relativistic) electron configuration as specified by its shell occupation.
`ConfigurationR`	(Relativistic) electron configuration as specified by its subshell occupation.
`EmMultipole`	A multipole (component) of the electro-magnetic field, specified by its parity and multipolarity.
`Level`	Atomic level in terms of its quantum numbers, symmetry, energy and its (possibly full) representation.
`Multiplet`	An ordered list of atomic levels.
`NuclearModel`	A nuclear model of an atom to keep all nuclear parameters together.
`Orbital`	(Relativistic) radial orbital function that appears as 'buildung block' in order to define the many-electron CSF; its is typically given on a (radial) grid and comprises as large and small component.
`Radial.Grid`	Radial grid to represent the (radial) orbitals and to perform all radial integrations.
`Radial.Potential`	Radial potential function.
`Radiative.Channel`	Radiative channel of well-defined multipolarity and gauge.
`Radiative.Line`	Radiative line between two given (initial- and final-state) levels, and along with all of its multipole channels.
`Radiative.Settings`	From the user specified settings for computing radiative lines.
`Shell`	Non-relativistic shell, such as $1s$, $2s$, $2p$, ...
`Subshell`	Relativistic subshell, such as $1s_{1/2}$, $2s_{1/2}$, $2p_{1/2}$, $2p_{3/2}$, ...
`Statistical.Tensor`	Statistical tensor of given rank k, projection q, and which typically depends on two atomic levels (resonances).

S. Fritzsche, Comput. Phys. Commun., **240** (2019) 1-14.

What do we need in atomic structure and collision theory ?

– a descriptive language for doing atomic computations ...

Requirements:

3

- Data types close to atomic physics.

  Shell, Subshell, Configuration, Orbital, Basis, Level, Multiplet, Cascade, Pulse, ...

- Implementation and co
- Support a coarse-g

  A pseudo-code descr

- Simple to learn and

  With a simplified cor
  computations, compl
  atomic cascades; ...

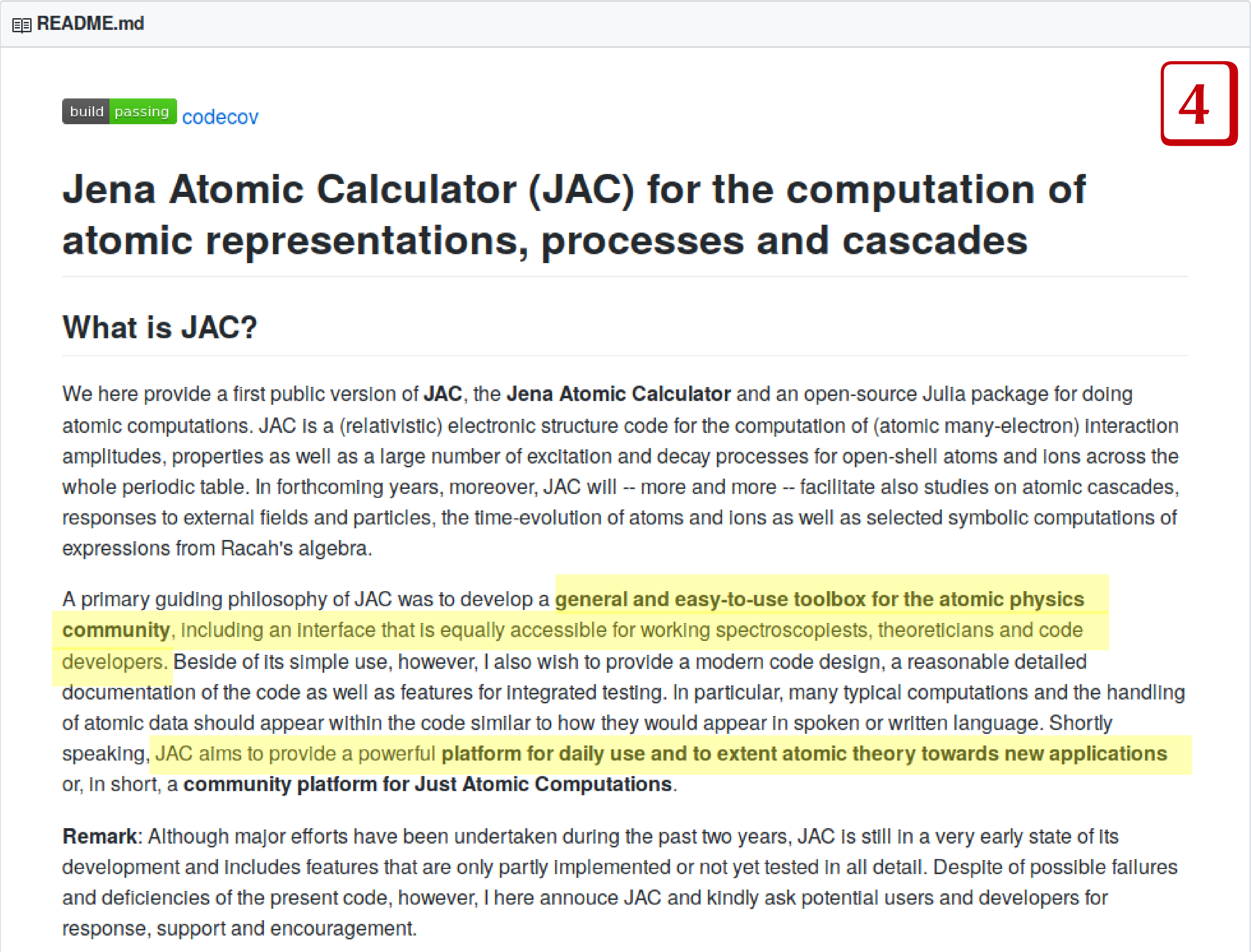
Why Julia ?

- (Very) fast, high-level language (from MIT, since ~ 2012).
- Combines productivity and performance.
- Multiple dispatch ... to distinguish generic code, still dynamic.
- Powerful data type hierarchy: Abstract types.
- Just in-time (JIT) compilation, fast loops.
- Rapid code development: no linkage; in-built benchmarking.
- Most code & macros are written in Julia.
- Extensive list of packages.
- No storage management, little declaration; type stability.
- Easy documentation, ...

S. Fritzsche, Comput. Phys. Commun

https://www.github.com/OpenJAC/JAC.jl

4



README.md

build passing codecov

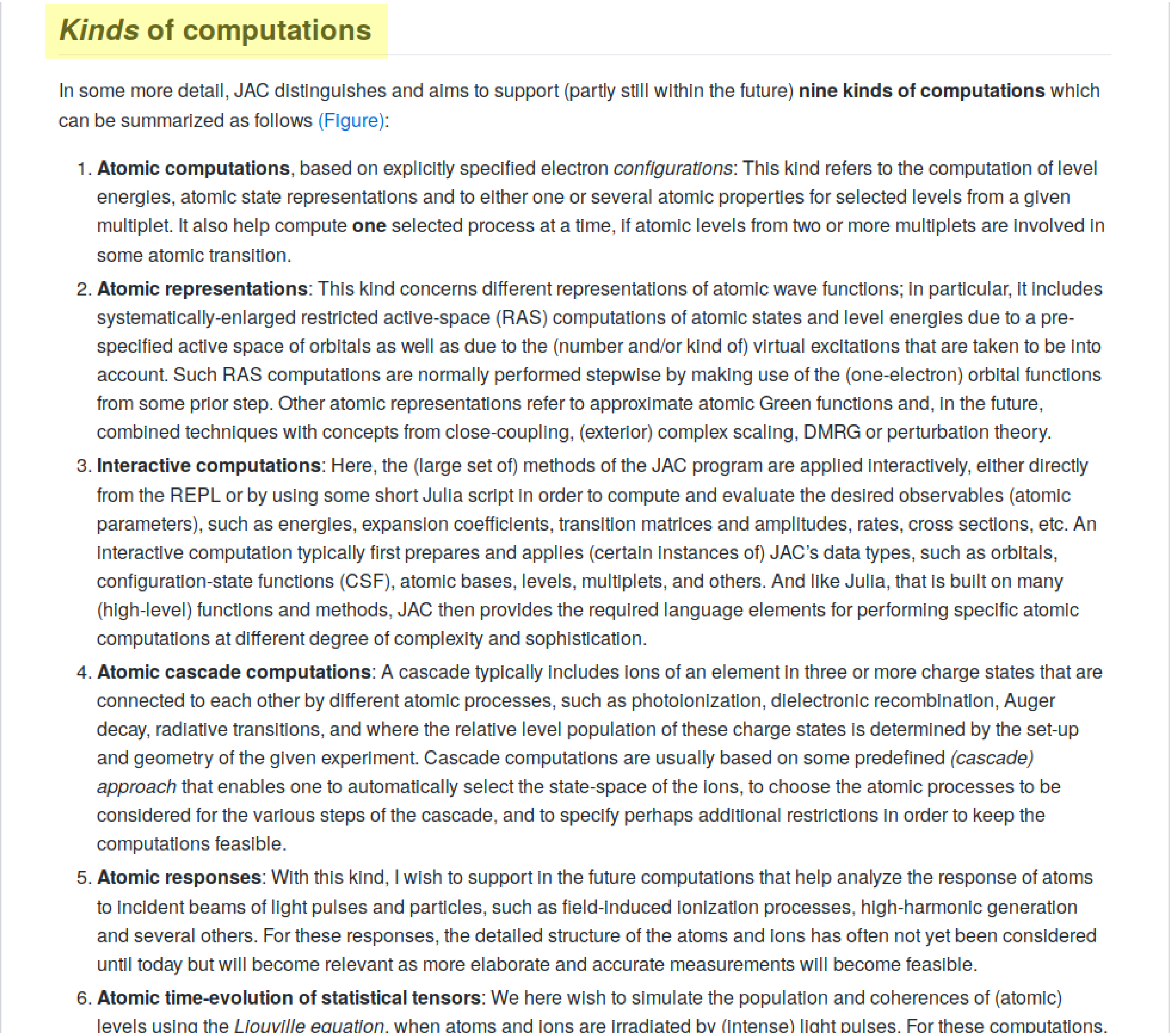
Jena Atomic Calculator (JAC) for the computation of atomic representations, processes and cascades

What is JAC?

We here provide a first public version of **JAC**, the **Jena Atomic Calculator** and an open-source Julia package for doing atomic computations. JAC is a (relativistic) electronic structure code for the computation of (atomic many-electron) interaction amplitudes, properties as well as a large number of excitation and decay processes for open-shell atoms and ions across the whole periodic table. In forthcoming years, moreover, JAC will -- more and more -- facilitate also studies on atomic cascades, responses to external fields and particles, the time-evolution of atoms and ions as well as selected symbolic computations of expressions from Racah's algebra.

A primary guiding philosophy of JAC was to develop a **general and easy-to-use toolbox for the atomic physics community**, including an interface that is equally accessible for working spectroscopiests, theoreticians and code developers. Beside of its simple use, however, I also wish to provide a modern code design, a reasonable detailed documentation of the code as well as features for integrated testing. In particular, many typical computations and the handling of atomic data should appear within the code similar to how they would appear in spoken or written language. Shortly speaking, JAC aims to provide a powerful **platform for daily use and to extent atomic theory towards new applications** or, in short, a **community platform for Just Atomic Computations**.

Remark: Although major efforts have been undertaken during the past two years, JAC is still in a very early state of its development and includes features that are only partly implemented or not yet tested in all detail. Despite of possible failures and deficiencies of the present code, however, I here annouce JAC and kindly ask potential users and developers for response, support and encouragement.

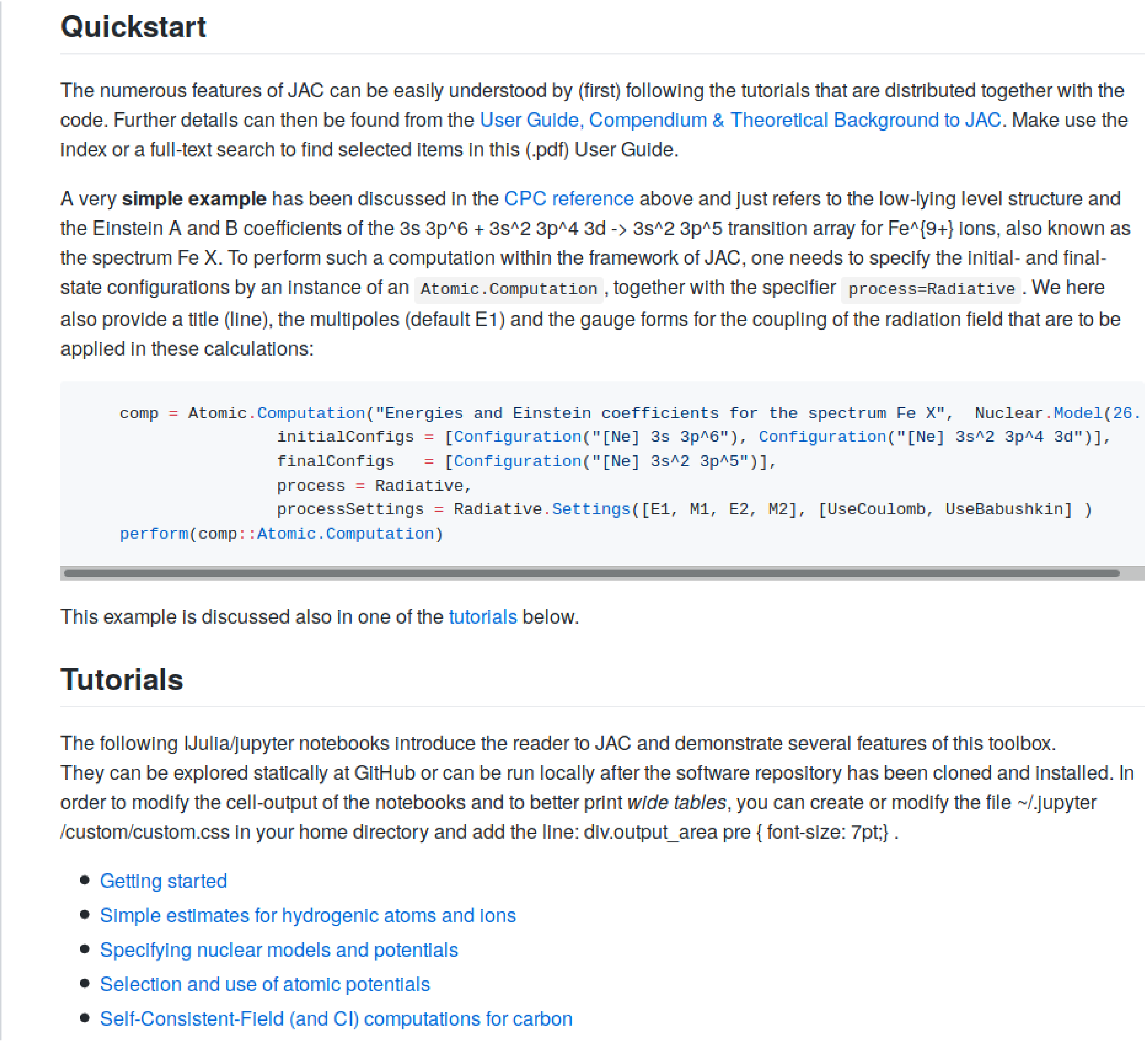


Kinds of computations

In some more detail, JAC distinguishes and aims to support (partly still within the future) **nine kinds of computations** which can be summarized as follows (Figure):

1. **Atomic computations**, based on explicitly specified electron *configurations*: This kind refers to the computation of level energies, atomic state representations and to either one or several atomic properties for selected levels from a given multiplet. It also help compute **one** selected process at a time, if atomic levels from two or more multiplets are involved in some atomic transition.
2. **Atomic representations**: This kind concerns different representations of atomic wave functions; in particular, it includes systematically-enlarged restricted active-space (RAS) computations of atomic states and level energies due to a pre-specified active space of orbitals as well as due to the (number and/or kind of) virtual excitations that are taken to be into account. Such RAS computations are normally performed stepwise by making use of the (one-electron) orbital functions from some prior step. Other atomic representations refer to approximate atomic Green functions and, in the future, combined techniques with concepts from close-coupling, (exterior) complex scaling, DMRG or perturbation theory.
3. **Interactive computations**: Here, the (large set of) methods of the JAC program are applied interactively, either directly from the REPL or by using some short Julia script in order to compute and evaluate the desired observables (atomic parameters), such as energies, expansion coefficients, transition matrices and amplitudes, rates, cross sections, etc. An interactive computation typically first prepares and applies (certain instances of) JAC's data types, such as orbitals, configuration-state functions (CSF), atomic bases, levels, multiplets, and others. And like Julia, that is built on many (high-level) functions and methods, JAC then provides the required language elements for performing specific atomic computations at different degree of complexity and sophistication.
4. **Atomic cascade computations**: A cascade typically includes ions of an element in three or more charge states that are connected to each other by different atomic processes, such as photoionization, dielectronic recombination, Auger decay, radiative transitions, and where the relative level population of these charge states is determined by the set-up and geometry of the given experiment. Cascade computations are usually based on some predefined *(cascade) approach* that enables one to automatically select the state-space of the ions, to choose the atomic processes to be considered for the various steps of the cascade, and to specify perhaps additional restrictions in order to keep the computations feasible.
5. **Atomic responses**: With this kind, I wish to support in the future computations that help analyze the response of atoms to incident beams of light pulses and particles, such as field-induced ionization processes, high-harmonic generation and several others. For these responses, the detailed structure of the atoms and ions has often not yet been considered until today but will become relevant as more elaborate and accurate measurements will become feasible.
6. **Atomic time-evolution of statistical tensors**: We here wish to simulate the population and coherences of (atomic) levels using the *Liouville equation*, when atoms and ions are irradiated by (intense) light pulses. For these computations,

https://www.github.com/OpenJAC/JAC.jl



Quickstart

The numerous features of JAC can be easily understood by (first) following the tutorials that are distributed together with the code. Further details can then be found from the User Guide, Compendium & Theoretical Background to JAC. Make use the index or a full-text search to find selected items in this (.pdf) User Guide.

A very **simple example** has been discussed in the CPC reference above and just refers to the low-lying level structure and the Einstein A and B coefficients of the 3s 3p^6 + 3s^2 3p^4 3d -> 3s^2 3p^5 transition array for Fe^{9+} ions, also known as the spectrum Fe X. To perform such a computation within the framework of JAC, one needs to specify the initial- and final-state configurations by an instance of an `Atomic.Computation`, together with the specifier `process=Radiative`. We here also provide a title (line), the multipoles (default E1) and the gauge forms for the coupling of the radiation field that are to be applied in these calculations:

```
comp = Atomic.Computation("Energies and Einstein coefficients for the spectrum Fe X",  Nuclear.Model(26.
                initialConfigs = [Configuration("[Ne] 3s 3p^6"), Configuration("[Ne] 3s^2 3p^4 3d")],
                finalConfigs   = [Configuration("[Ne] 3s^2 3p^5")],
                process = Radiative,
                processSettings = Radiative.Settings([E1, M1, E2, M2], [UseCoulomb, UseBabushkin] )
perform(comp::Atomic.Computation)
```

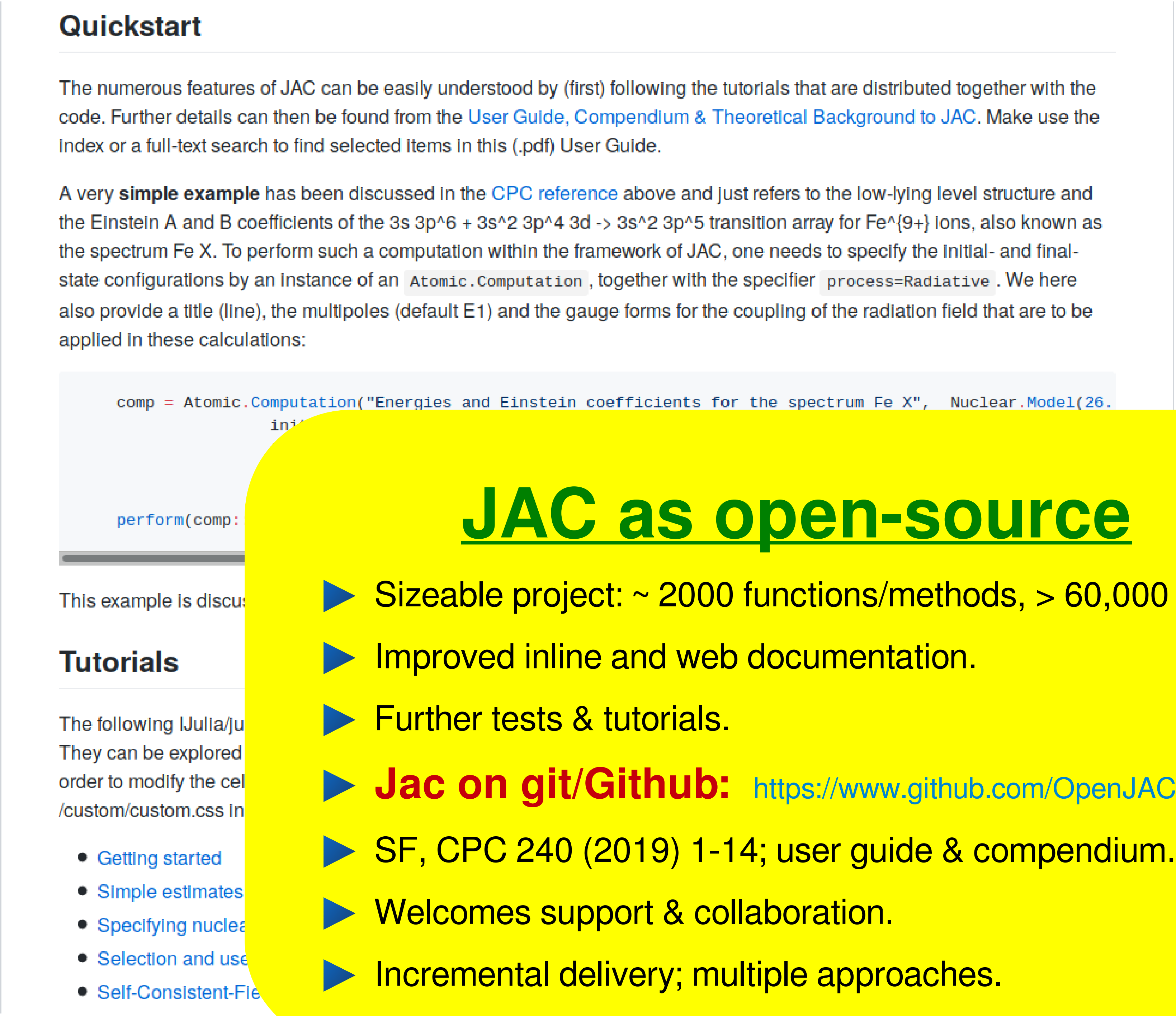
This example is discussed also in one of the tutorials below.

Tutorials

The following IJulia/jupyter notebooks introduce the reader to JAC and demonstrate several features of this toolbox. They can be explored statically at GitHub or can be run locally after the software repository has been cloned and installed. In order to modify the cell-output of the notebooks and to better print *wide tables*, you can create or modify the file ~/.jupyter /custom/custom.css in your home directory and add the line: div.output_area pre { font-size: 7pt;} .

- Getting started
- Simple estimates for hydrogenic atoms and ions
- Specifying nuclear models and potentials
- Selection and use of atomic potentials
- Self-Consistent-Field (and CI) computations for carbon

https://www.github.com/OpenJAC/JAC.jl



Quickstart

The numerous features of JAC can be easily understood by (first) following the tutorials that are distributed together with the code. Further details can then be found from the User Guide, Compendium & Theoretical Background to JAC. Make use the index or a full-text search to find selected items in this (.pdf) User Guide.

A very **simple example** has been discussed in the CPC reference above and just refers to the low-lying level structure and the Einstein A and B coefficients of the 3s 3p^6 + 3s^2 3p^4 3d -> 3s^2 3p^5 transition array for Fe^{9+} ions, also known as the spectrum Fe X. To perform such a computation within the framework of JAC, one needs to specify the initial- and final-state configurations by an instance of an `Atomic.Computation` , together with the specifier `process=Radiative` . We here also provide a title (line), the multipoles (default E1) and the gauge forms for the coupling of the radiation field that are to be applied in these calculations:

```
    comp = Atomic.Computation("Energies and Einstein coefficients for the spectrum Fe X",  Nuclear.Model(26.
                             ini



    perform(comp:
```

This example is discu

Tutorials

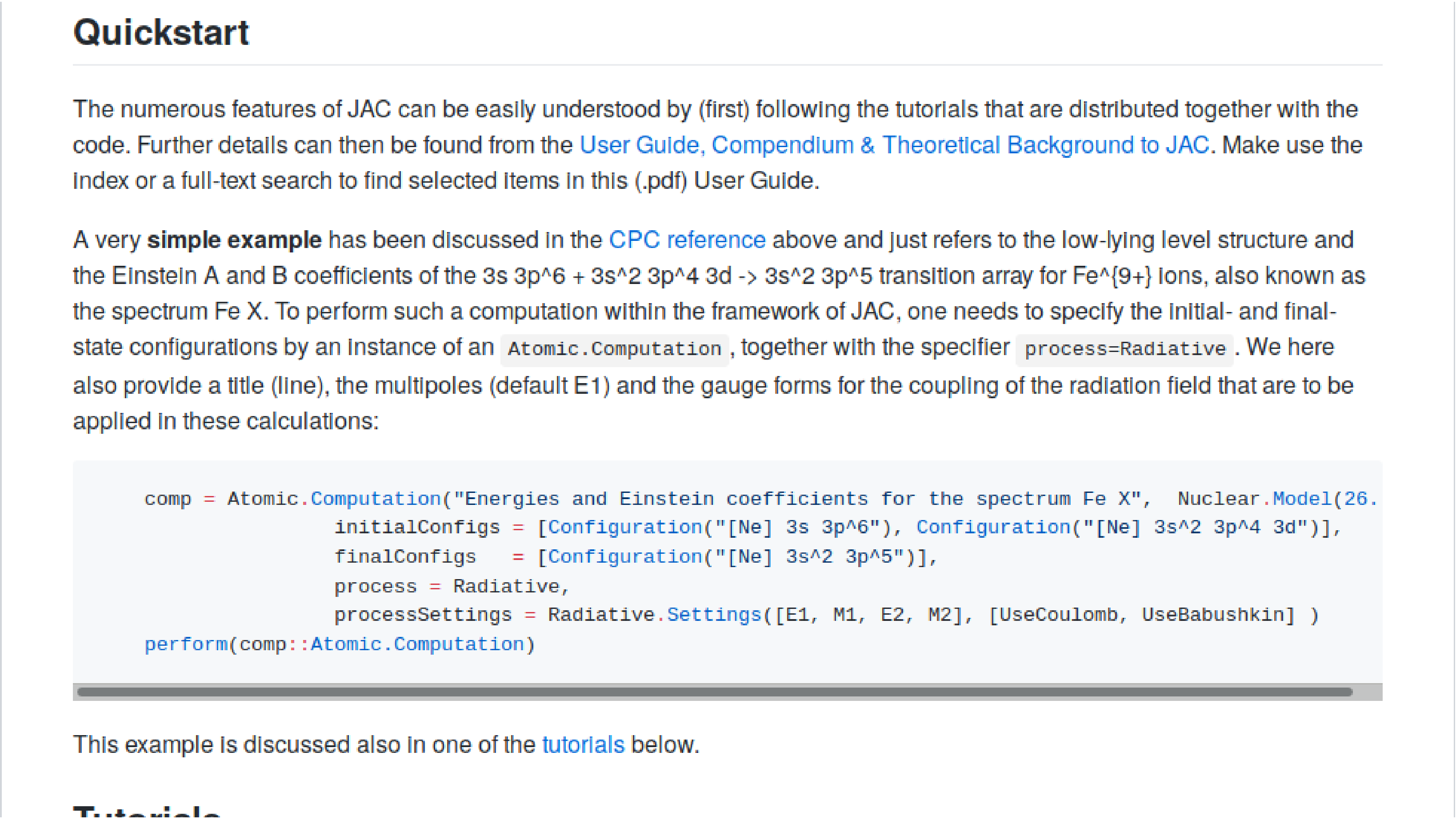
The following IJulia/ju
They can be explored
order to modify the cel
/custom/custom.css in

- Getting started
- Simple estimates
- Specifying nucle
- Selection and use
- Self-Consistent-Fie

JAC as open-source

- ▶ Sizeable project: ~ 2000 functions/methods, > 60,000 lines
- ▶ Improved inline and web documentation.
- ▶ Further tests & tutorials.
- ▶ **Jac on git/Github:** https://www.github.com/OpenJAC/JAC.jl
- ▶ SF, CPC 240 (2019) 1-14; user guide & compendium.
- ▶ Welcomes support & collaboration.
- ▶ Incremental delivery; multiple approaches.

https://www.github.com/OpenJAC/JAC.jl



Quickstart

The numerous features of JAC can be easily understood by (first) following the tutorials that are distributed together with the code. Further details can then be found from the User Guide, Compendium & Theoretical Background to JAC. Make use the index or a full-text search to find selected items in this (.pdf) User Guide.

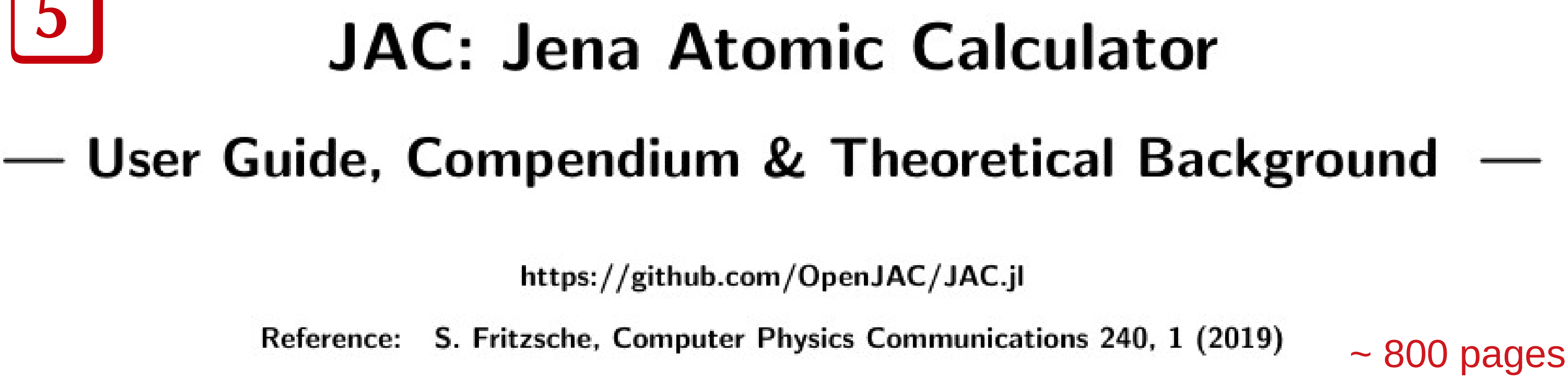
A very **simple example** has been discussed in the CPC reference above and just refers to the low-lying level structure and the Einstein A and B coefficients of the 3s 3p^6 + 3s^2 3p^4 3d -> 3s^2 3p^5 transition array for Fe^{9+} ions, also known as the spectrum Fe X. To perform such a computation within the framework of JAC, one needs to specify the initial- and final-state configurations by an instance of an `Atomic.Computation`, together with the specifier `process=Radiative`. We here also provide a title (line), the multipoles (default E1) and the gauge forms for the coupling of the radiation field that are to be applied in these calculations:

```
comp = Atomic.Computation("Energies and Einstein coefficients for the spectrum Fe X",  Nuclear.Model(26.
                initialConfigs  = [Configuration("[Ne] 3s 3p^6"), Configuration("[Ne] 3s^2 3p^4 3d")],
                finalConfigs    = [Configuration("[Ne] 3s^2 3p^5")],
                process = Radiative,
                processSettings = Radiative.Settings([E1, M1, E2, M2], [UseCoulomb, UseBabushkin] )
perform(comp::Atomic.Computation)
```

This example is discussed also in one of the tutorials below.

Tutorials

5



JAC: Jena Atomic Calculator

— User Guide, Compendium & Theoretical Background —

https://github.com/OpenJAC/JAC.jl

Reference: S. Fritzsche, Computer Physics Communications 240, 1 (2019)

~ 800 pages

(I) Using JAC: Level structure of Th^{2+}

-- SCF + CI computations; QED estimates, ...

Example: Low-lying levels of Th^{2+} (Z=90) from the excited configurations

[Rn] (5f6d + 5f7s + 5f7d + 6d7p + 7s7p + ...)

... together with **QED corrections** and **jj → LS transformation**

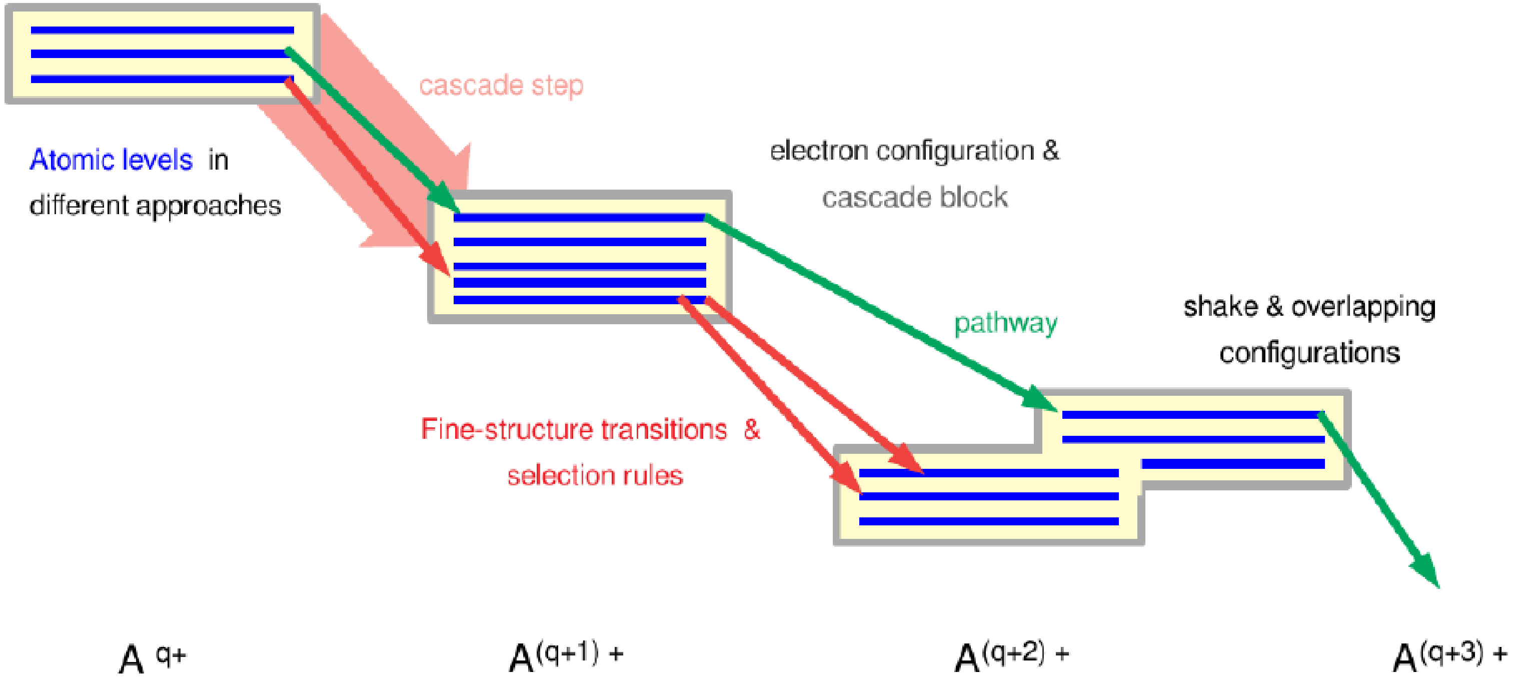
```
> wa = Atomic.Computation("Th^2+ QED estimate + jj-LS level transformation",  Nuclear.Model(90.);
             configs=[Configuration("[Rn] 5f6d"), Configuration("[Rn] 5f7s"), Configuration("[Rn] 6d7p"),
                    Configuration("[Rn] 7s7"), ... ],
             asfSettings=AsfSettings(true, false, "meanDFS", "hydrogenic", ..., [1],   0, 1.0e-6,
                                   Subshell[],  true, false, Petersburg(), LSjjSettings(true),
                                   False, [ i for i=1:10 ], false, LevelSymmetry[] ) )

> perform(wa)
```

... in perform('computation: SCF', ...)

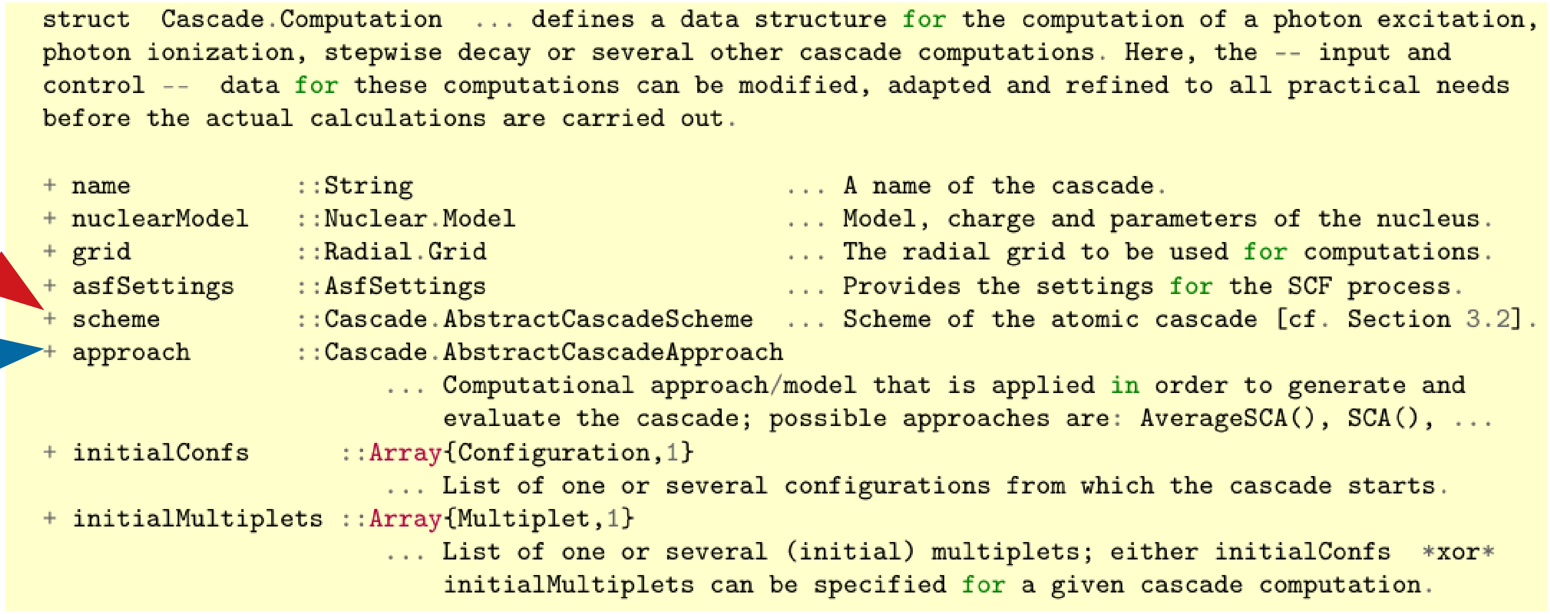
SF, P. Palmeri & S. Schippers, Symmetry **13** (2021) 520.

(II) Atomic cascades



Physics/ stabilization

Technical realization

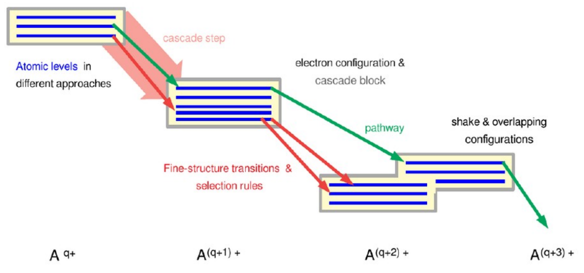


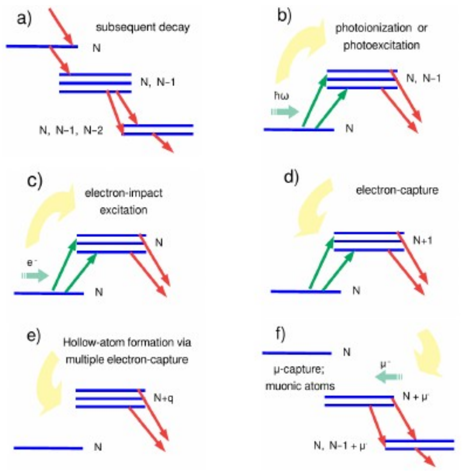
```
struct  Cascade.Computation  ... defines a data structure for the computation of a photon excitation,
photon ionization, stepwise decay or several other cascade computations. Here, the -- input and
control --  data for these computations can be modified, adapted and refined to all practical needs
before the actual calculations are carried out.

+ name             ::String                          ... A name of the cascade.
+ nuclearModel     ::Nuclear.Model                   ... Model, charge and parameters of the nucleus.
+ grid             ::Radial.Grid                     ... The radial grid to be used for computations.
+ asfSettings      ::AsfSettings                     ... Provides the settings for the SCF process.
+ scheme           ::Cascade.AbstractCascadeScheme   ... Scheme of the atomic cascade [cf. Section 3.2].
+ approach         ::Cascade.AbstractCascadeApproach
                        ... Computational approach/model that is applied in order to generate and
                            evaluate the cascade; possible approaches are: AverageSCA(), SCA(), ...
+ initialConfs       ::Array{Configuration,1}
                        ... List of one or several configurations from which the cascade starts.
+ initialMultiplets ::Array{Multiplet,1}
                        ... List of one or several (initial) multiplets; either initialConfs  *xor*
                            initialMultiplets can be specified for a given cascade computation.
```

SF, P. Palmeri & S. Schippers, Symmetry **13** (2021) 520.

(II) Atomic cascades





Example: Calculation of the Mg 1s $2s^2$ $2p^6$ $3s^2$ decay cascade after 1s ionization

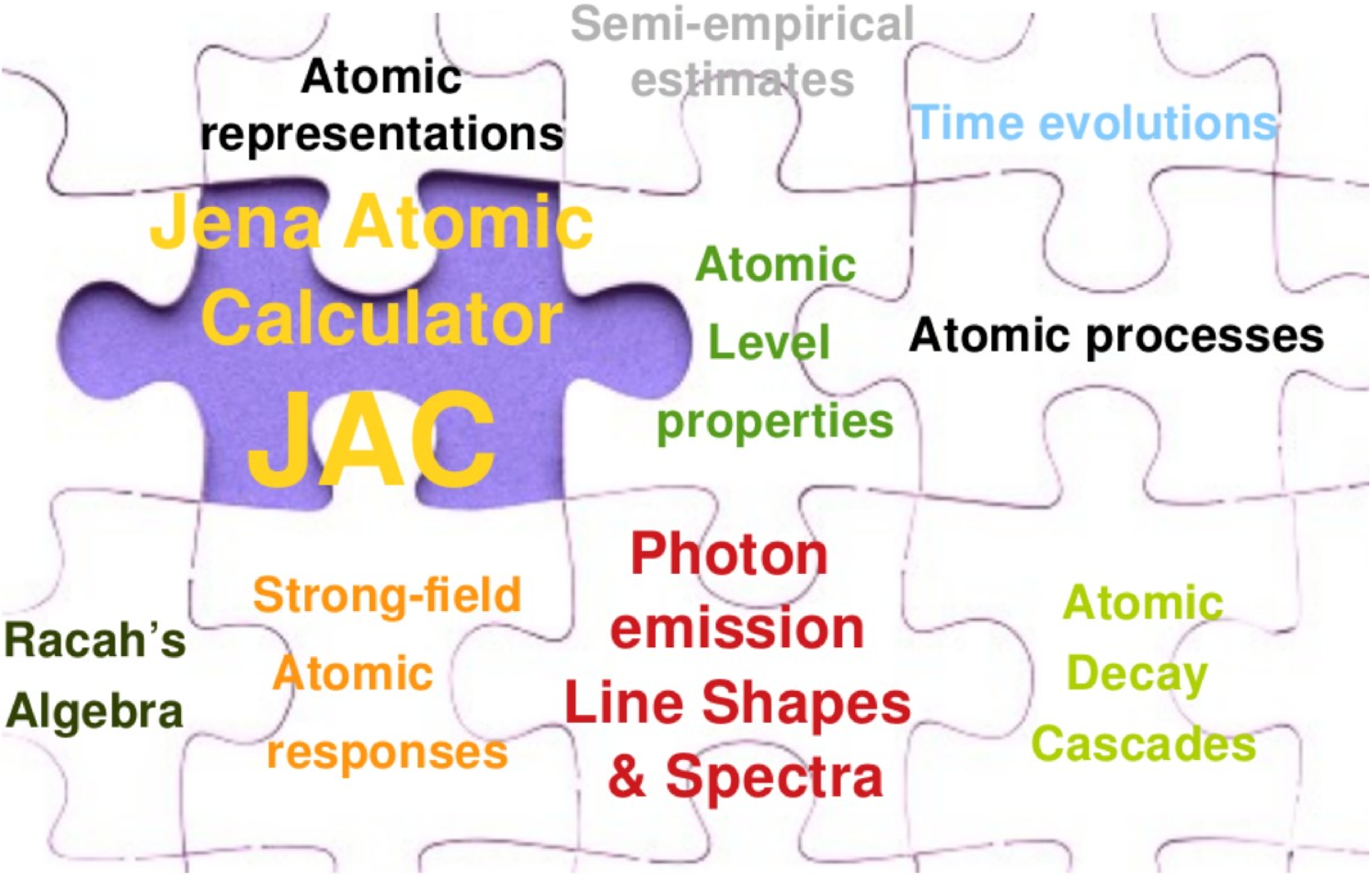
```
> decayScheme = Cascade.StepwiseDecayScheme([Auger(), Radiative()], 3, ...)

> grid = Radial.Grid(Radial.Grid(false), rnt = 4.0e-6, h = 5.0e-2, hp = 1.0e-2, rbox = 10.0)
> name = "Computation of the Mg  1s 2s^2 2p^6 3s^2 decay cascade after 1s ionization"

> wa   = Cascade.Computation(Cascade.Computation(); name = name,
                    NuclearModel = Nuclear.Model(12.),  grid = grid,
                   approach = Cascade.AverageSCA(), scheme = decayScheme,
                    InitialConfigs = [Configuration("1s 2s^2 2p^6 3s^2")] )

> wb   = perform(wa; output=true)
```

https://www.github.com/OpenJAC/JAC.jl



Jena Atomic Calculator

JAC

Atomic representations

Semi-empirical estimates

Time evolutions

Atomic Level properties

Atomic processes

Racah's Algebra

Strong-field Atomic responses

Photon emission Line Shapes & Spectra

Atomic Decay Cascades

https://www.github.com/OpenJAC/JAC.jl

Summary

- Large: … sizeable toolbox & code for a wide number of appliations.
- User-friendly: … for spectroscopy (experiment), theory and code developers.
- Open software: … new features by demand & search for collaboration.
- Necessary: … many modern applications in astro physics & elsewhere.
- Useful: … consistent data for different systems, processes & interactions.

Future of atomic theory ? ... How to continue ?

... How to benefit from each other ?