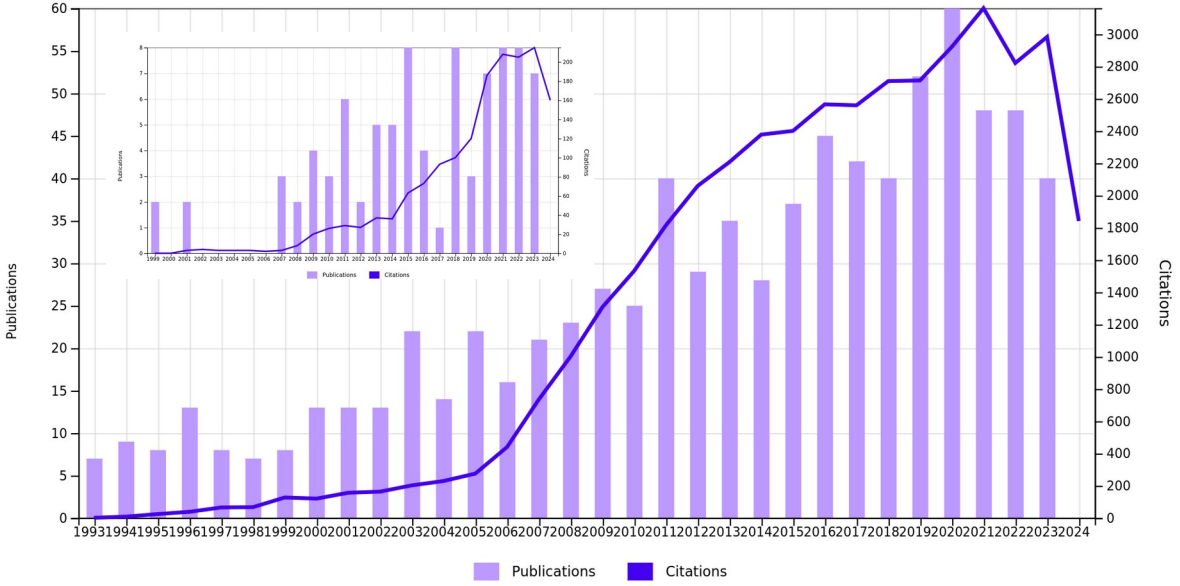
Atomic cascade computations for (nuclear) astrophysics

News from JAC

„tools for Just Atomic Computations“

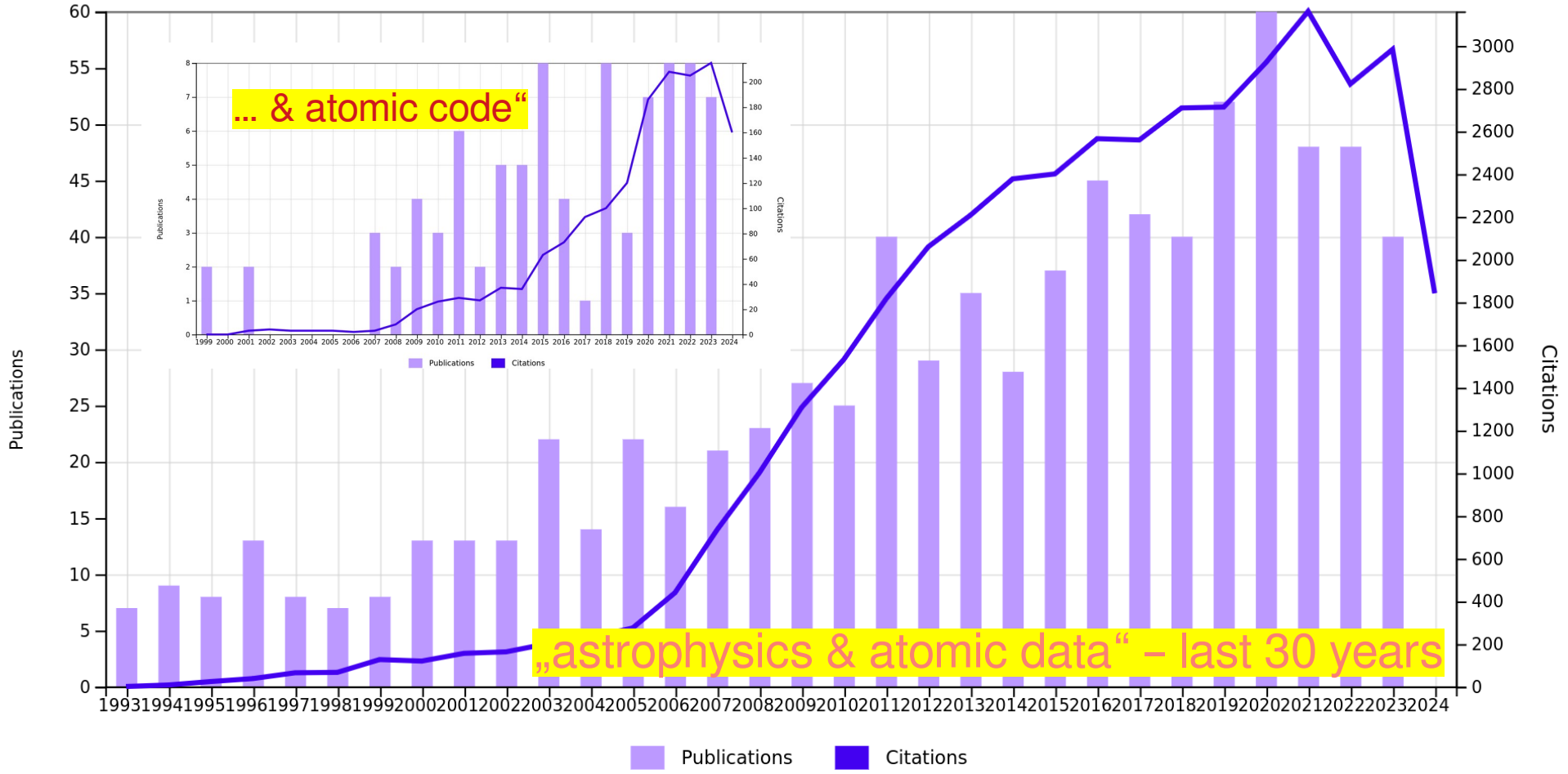
Stephan Fritzsche
Helmholtz-Institut Jena &
Theoretisch-Physikalisches Institut Jena
19th September 2024



Support & encouragement: H. Huang, A. Kumar, S. Schippers, Z.W. Wu,
AP@GSI, R. Beerwerth, B. Böning, G. Gaigalas, J. Gill, A. Harris,
Y. Hikosaka, N. Hosea, F. Koike,T. Mazza, M. Meyer, L. Sharma, S. Stock, A. Surzhykov, W. Wang, ...

Atomic cascade computations for (nuclear) astrophysics

		Total	950	1,073	921	1,180	740	360.97	11,190
⊖ 1	Atomic data for astrophysics .2. New analytic fits for photoionization cross sections of atoms and ions Verner, DA; Ferland, GJ; (...); Yakovlev, DG Jul 1 1996 | ASTROPHYSICAL JOURNAL 465 (1) , pp.487-498		109	116	113	152	75	53.1	1,540
⊖ 2	A major upgrade of the VALD database Ryabchikova, T; Piskunov, N; (...); Barklem, PS May 2015 | PHYSICA SCRIPTA 90 (5)		71	91	68	116	59	62.6	626

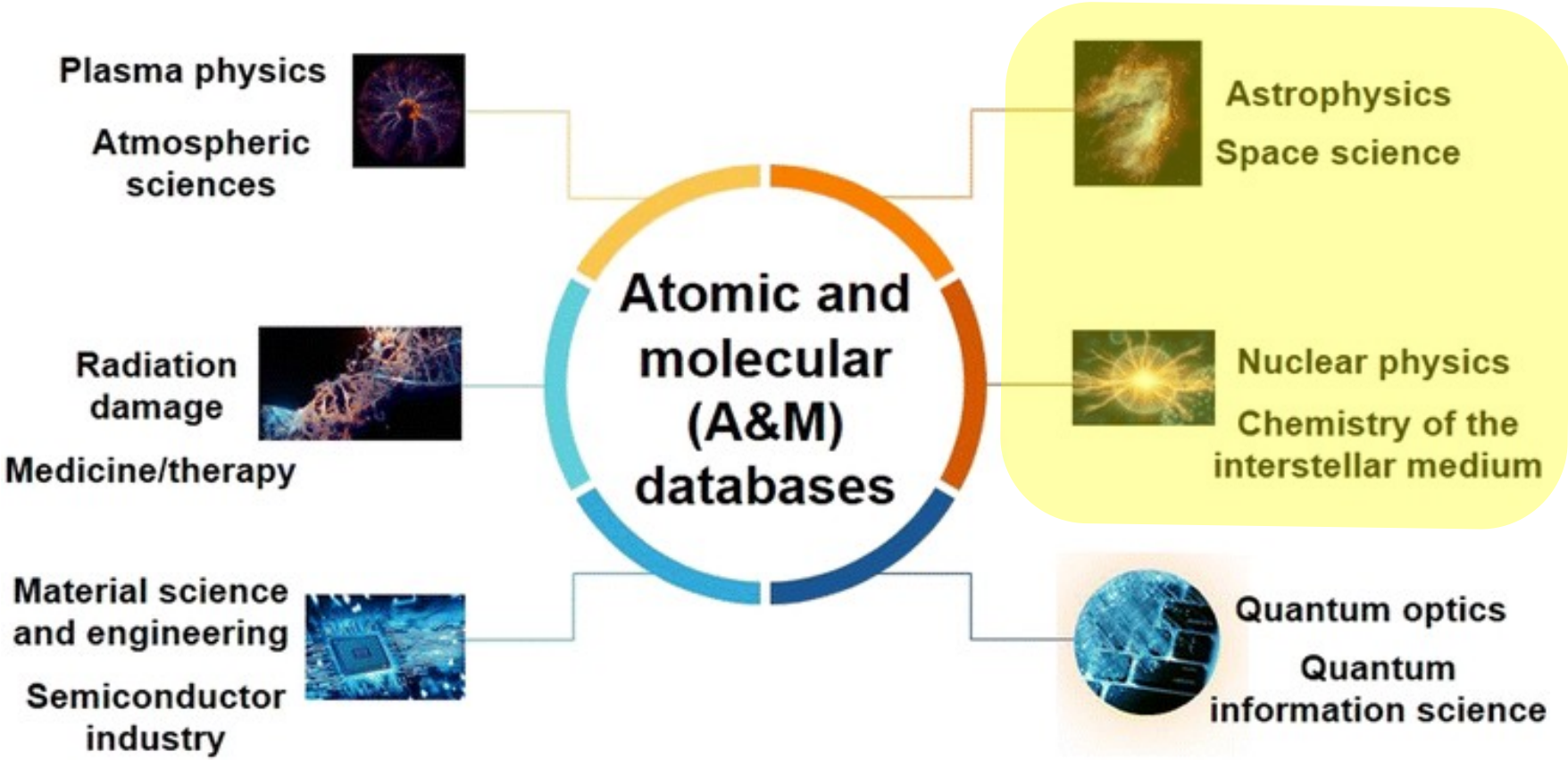


Atomic cascade computations for (nuclear) astrophysics

Role of atomic data & computations

X-ray astronomy (for example)

- Experiments / nuclear techiques
- Observations
- Modeling
- Atomic computations



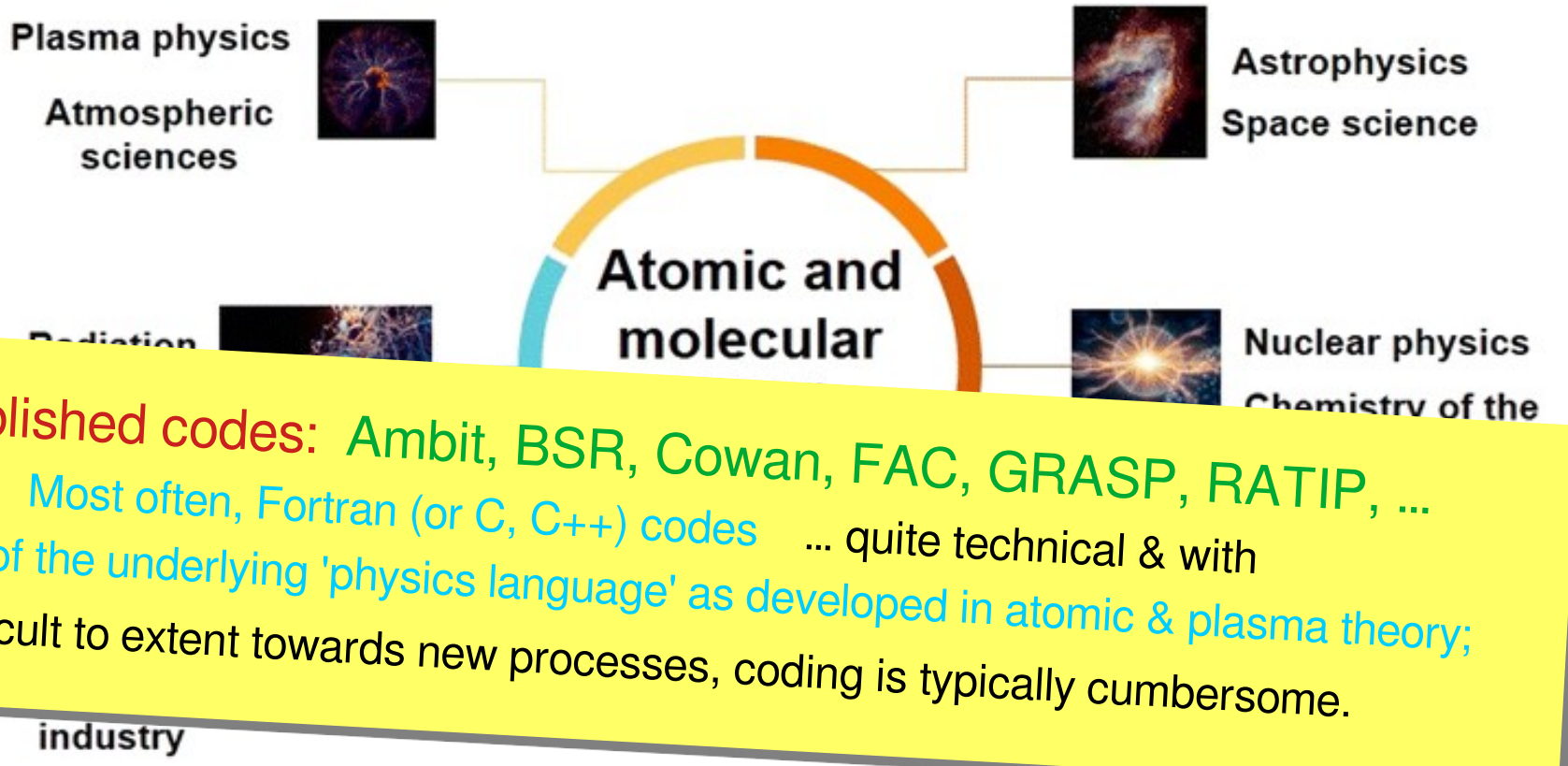
DOI: 10.1039/D3CP03752E (Review Article) Phys. Chem. Chem. Phys., 2023, 25, 26972-26985

Atomic cascade computations for (nuclear) astrophysics

Role of atomic data & computations

X-ray astronomy (for example)

- Experiments / nuclear techiques
- Observations
- Modeling
- Atomic computations



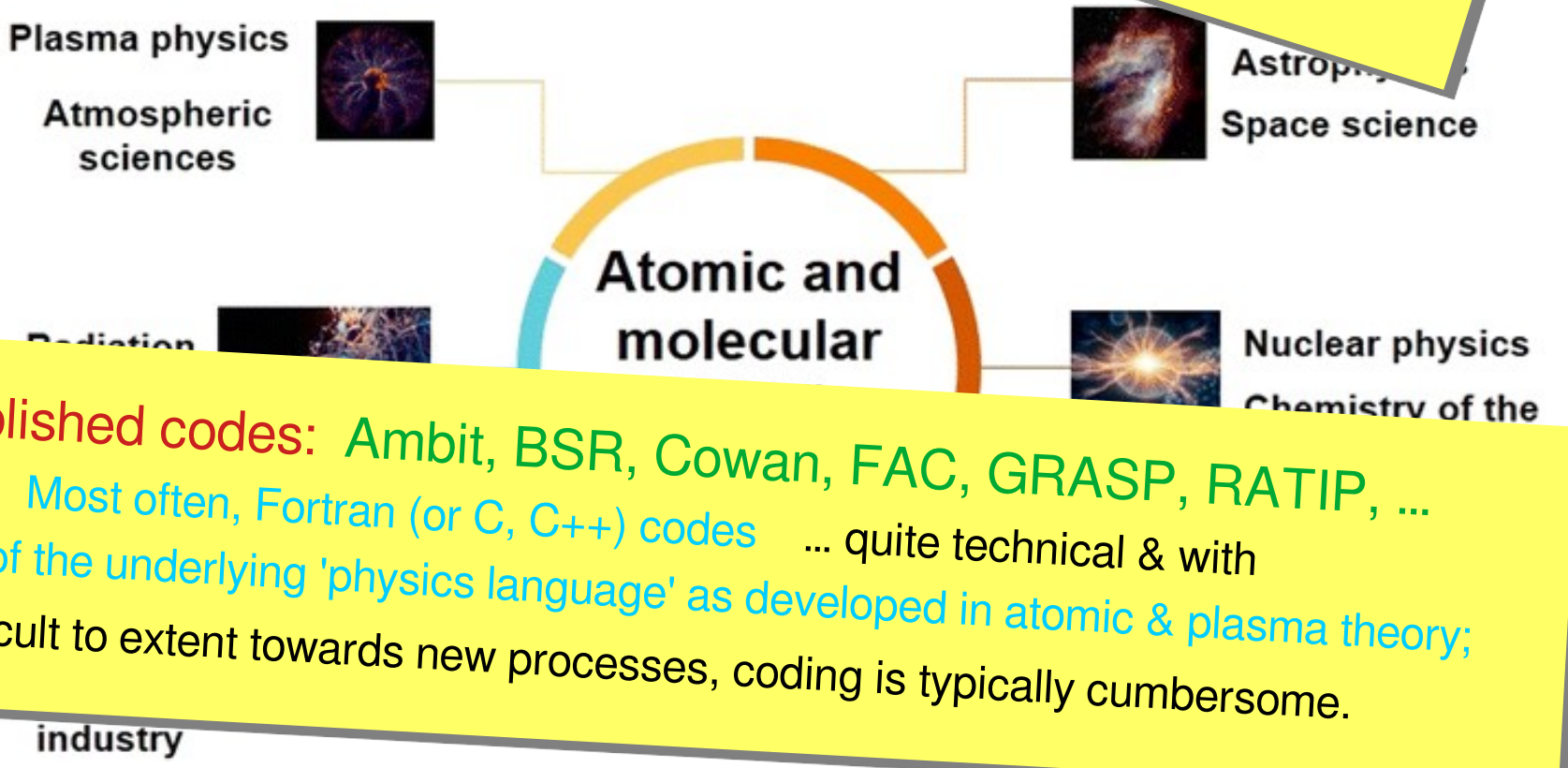
Established codes: Ambit, BSR, Cowan, FAC, GRASP, RATIP, ...

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

DOI: 10.1039/D3CP03752E (Review Article) Phys. Chem. Chem. Phys., 2023, 25, 26972-26985

astropy
A Community Python Library for Astronomy

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



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

DOI: 10.1039/D3CP03752E (Review Article) Phys. Chem. Chem. Phys., 2023, 25, 26972-26985

Jena Atomic Calculator (JAC)

(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 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
$e_s^- + A$	$\longrightarrow$	$A^* + e_s^{-\prime}$	... electron – impact excitation
$e_s^- + A$	$\longrightarrow$	$A^* + e_s^{-\prime} + e^-$	... electron – impact ionization
$A + n\,\hbar\omega$	$\longrightarrow$	A^*	... multi – photon excitation/decay

Quiz: Atomic processes in a nutshell

-- for “intermediates” in atomic and astro physics

$$A + n\,\hbar\omega \longrightarrow A^{+(*)} + e_p^- \quad \text{... multi-photon ionization}$$

$$A + n\,\hbar\omega \longrightarrow A^{+(*)} + (e_{p_1}^- + e_{p_2}^-) \quad \text{... multi-photon double ionization}$$

$$A^{q+} + e_s^- \longrightarrow A^{(q-1)+} + \hbar\omega \quad \text{... radiative recombination}$$

$$A^{q+} + e_s^- \longrightarrow A^{(q-1)+*} \longrightarrow A^{(q-1)+(*)} + \hbar\omega \quad \text{... dielectronic recombination}$$

$$A + \hbar\omega \longrightarrow A^{(*)} + \hbar\omega' \quad \text{... Rayleigh/Compton}$$

$$A^{q+*} \longrightarrow A^{(q+1)+(*)} + (e_a^- + \hbar\omega) \quad \text{... radiative Auger}$$

$$A^{q+*} \longrightarrow A^{(q+2)+(*)} + (e_{a_1}^- + e_{a_2}^-) \quad \text{... double Auger}$$

$$A + \hbar\omega \longrightarrow A^* \longrightarrow A^{(*)} + \hbar\omega' \quad \text{... photo-}$$

$$A + \hbar\omega \longrightarrow A^{+*}$$

- Indeed, these and many other processes (> 30) frequently occur in atomic spectroscopy, also in astro and plasma physics as well as at various places elsewhere.
- Often, these processes occur as „atomic cascades“ with increasing complexity.
- Which support should and can atomic theory provide ?
- Which tools are simple, suitable & readily available ?

Jena Atomic Calculator (JAC)

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

Features:

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

Is such a common „computational suite“ possible, desirable & feasible ?
How much help is needed by experiment & observations ?

Jena Atomic Calculator (JAC)

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

Features:

- **Necessary:** … for many modern applications in astro physics & elsewhere.
- **Useful:** … consistent data for different systems, processes & interactions.
- **Large:** … sizeable toolbox & code for a wide range of appliations.
- **Suitable:** … for spectros
- **Open software:** … new featur

Is such a common „computati
How much help is need

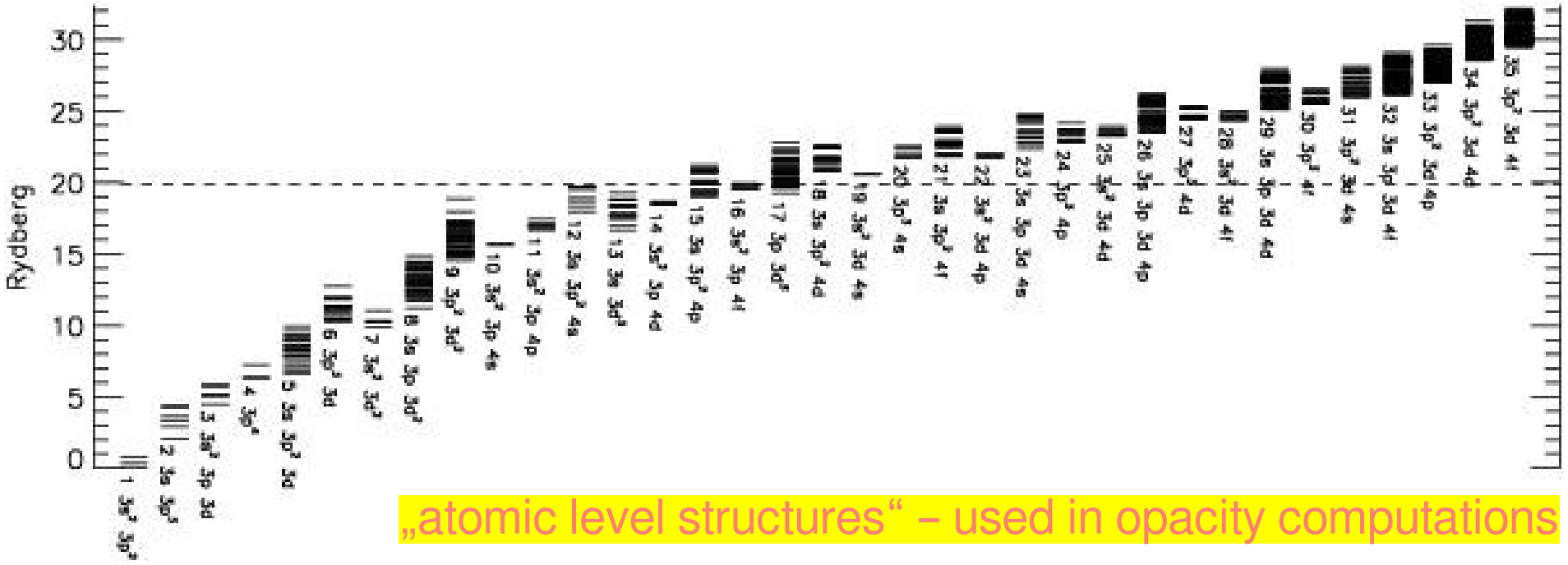
Plan ... or what is left for today

- Motivation: No. & variety of atomic processes
- JAC @ work: Getting started, RR + DR + ...
- Need & callenge of atomic cascades
- Customer's view: How to make use of JAC ?
- Summary

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

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

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



„atomic level structures“ – used in opacity computations

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

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

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

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

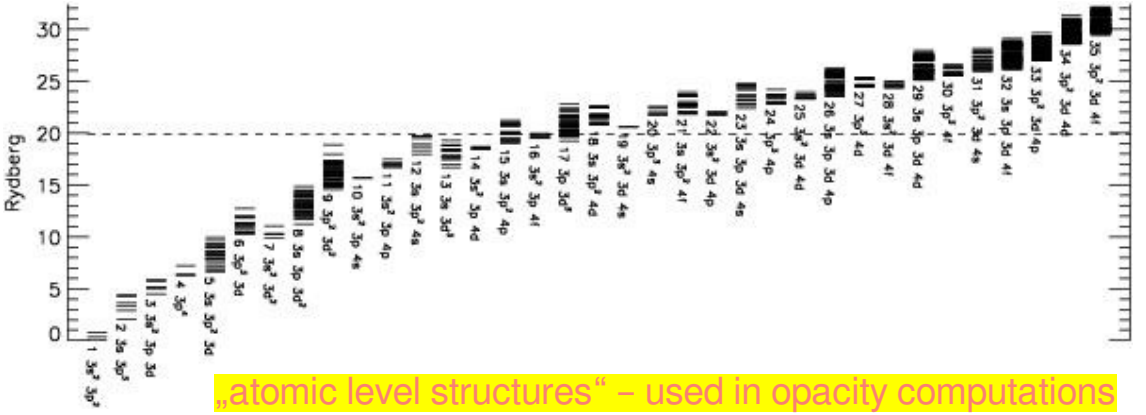
[Rn] (5f6d + 5f7s + 5f7d + 6d7p + 7s7p + ...)

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

```
> comp = 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] 7s7p"), ... ],
          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(comp)
```

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



„atomic level structures“ – used in opacity computations

Quiz: Atomic processes in a nutshell

-- for “intermediates” in atomic and astro physics

$$A + n\,\hbar\omega \longrightarrow A^{+(*)} + e_p^- \quad \text{... multi-photon ionization}$$

$$A + n\,\hbar\omega \longrightarrow A^{+(*)} + (e_{p_1}^- + e_{p_2}^-) \quad \text{... multi-photon double ionization}$$

$$A^{q+} + e_s^- \longrightarrow A^{(q-1)+} + \hbar\omega \quad \text{... radiative recombination}$$

$$A^{q+} + e_s^- \longrightarrow A^{(q-1)+*} \longrightarrow A^{(q-1)+(*)} + \hbar\omega \quad \text{... dielectronic recombination}$$

$$A + \hbar\omega \longrightarrow A^{(*)} + \hbar\omega' \quad \text{... Rayleigh/Compton}$$

$$A^{q+*} \longrightarrow A^{(q+1)+(*)} + (e_a^- + \hbar\omega) \quad \text{... radiative Auger}$$

$$A^{q+*} \longrightarrow A^{(q+2)+(*)} + (e_{a_1}^- + e_{a_2}^-) \quad \text{... double Auger}$$

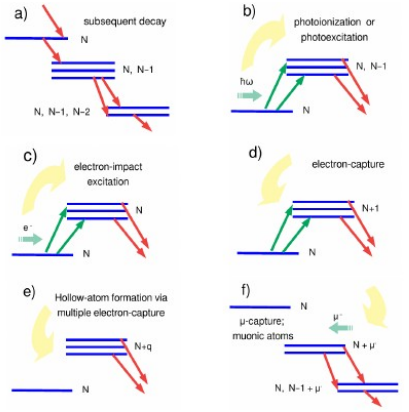
$$A + \hbar\omega \longrightarrow A^* \longrightarrow A^{(*)} + \hbar\omega' \quad \text{... photo-excitation \& fluores-nce}$$

$$A + \hbar\omega \longrightarrow A^{+,*} + e^- \longrightarrow \quad \text{...ce}$$

- Indeed, these and many other processes (> 30) occur in atomic spectroscopy, also in astro and plasma physics as well as at various places elsewhere.
- Often, these processes occur as „atomic cascades“ with increasing complexity.
- Which support should and can atomic theory provide ?
- Which tools are simple, suitable & readily available ?

(II) Radiative recombination (RR)

$$A^{q+} + e_s^- \longrightarrow A^{(q-1)+} + \hbar\omega$$

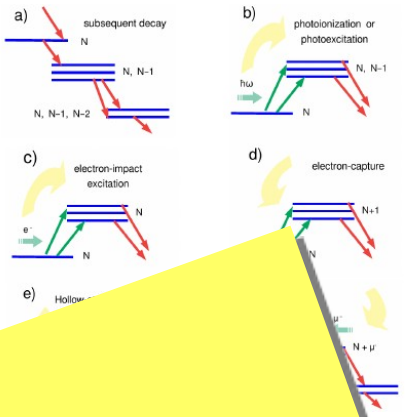


Example: Calculation of the Fe^{14+} $1s^2\ 2s^2\ 2p^6\ 3s^2$ RR cross sections

```
> setDefaults("unit: cross section", "Mbarn");    setDefaults("unit: energy", "eV")
> phSettings = PhotoRecombination.Settings([E1], [UseCoulomb], [1000.], [0.],
                                             false, true, false, false, true, 3, LineSelection() )

> 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 Fe^14+:  1s^2 2s^2 2p^6 3s^2  RR cross sections"

> comp = Atomic.Computation(Atomic.Computation(); name = name, grid = grid,
                nuclearModel = Nuclear.Model(26.), processSettings=phSettings,
                initialConfigs  = [Configuration("[Ne] 3s^2)“],
                finalConfigs    = [Configuration("[Ne] 3s^2 3p)“), Configuration("[Ne] 3s^2 3d)“)] )

 > results = perform(comp; output=true)
```

SF, CPC 240 (2019) 1-14

(II) Radiative recombination (RR)



$$A^{q+} + e_s^- \longrightarrow A^{(q-1)+} + \hbar\omega$$

Example: Calculation of the Fe^{14+} 1s 2s²

Great astrophysical interest

Radiative (photo-) recombination are crucial for rate equations and ionization balance in dynamical mergers.

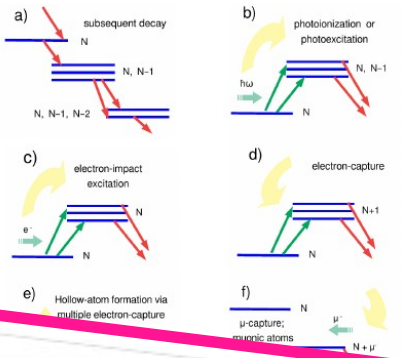
Saha equation & beyond;

tables of improved opacities and „line lists“.

```
> setDefaults("unit: cross sec
> phSettings = Phot
                                                    2, rbox = 10.0)
                                                    s sections"

                                    tion(); name = name, grid = grid,
                                    del(26.),  grid = grid, processSettings=phSettings,
                                    figuration("[Ne] 3s^2)“],
                                    = [Configuration("[Ne] 3s^2 3p)“), Configuration("[Ne] 3s^2 3d)“)] )

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

SF, CPC 240 (2019) 1-14

(II) Radiative recombination (RR)



$$A^{q+} + e_s^- \longrightarrow$$

Example: Calculation

```
> setDefaults("unit: cross
> phSettings = PhotoRe

> grid    = Radial.Grid(R
> name = "Computatio

> wa   = Atomic.Com
                NuclearM                                    s,
                InitialCo
                FinalCo                                     ')] )

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

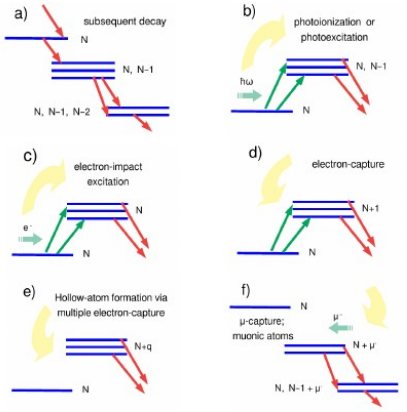
The code in this repository is distributed under the MIT licence. The associated User Guide, Compendium & Theoretical Background to JAC is distributed under the Creative Commons Attribution 4.0 International (CC BY 4.0) license.

For reference to (using) this code, please, use the Computer Physics Communications publication on JAC:

- S. Fritzsche: A fresh computational approach to atomic structures, processes and cascades Computer Physics Communications 240, 1 (2019)
- G. Gaigalas & S. Fritzsche: Angular coefficients for symmetry-adapted configuration states in jj-coupling. Comp. Phys. Commun. 267, 108086 (2021)
- S. Fritzsche, P. Palmeri & S. Schippers: Atomic cascade computations. Symmetry 13, 520 (2021)
- S. Fritzsche: Symbolic evaluation of expressions from Racah's algebra. Symmetry 13, 1558 (2021)
- S. Fritzsche & A. Surzhykov: Approximate atomic Green functions. Molecules 26, 2660 (2021)
- S. Fritzsche: Dielectronic recombination strengths and plasma rate coefficients of multiply-charged ions. A&A 656, A163 (2021)
- S. Fritzsche: Level structure and properties of open f-shell elements. Atoms 10, 7 (2022)
- S. Fritzsche: Photon emission from hollow ions near surfaces. Atoms 10, 37 (2022)
- S. Fritzsche, B. Böning: Strong-field ionization amplitudes for atomic many-electron targets. Atoms 10, 70 (2022)
- S. Fritzsche: Application of symmetry-adapted atomic amplitudes. Atoms 10, 127 (2022)
- S. Fritzsche, A.V. Maiorova & Z.W. Wu: Radiative recombination plasma rate coefficients of multiply-charged ions. Atoms 11, 50 (2023)
- S. Fritzsche, L.G. Jiao, Y.C. Wang & J.E. Sienkiewicz; Collision strengths of astrophysical interest for multiply charged ions. Atoms 11, 80 (2023)

See also `CITATION.bib` for the relevant references(s).

(II) Radiative recombination (RR)



$$A^{q+} + e_s^- \longrightarrow A^{(q-1)+} + \hbar\omega$$

Example: Calculation of the Fe^{14+} $1s\ 2s^2\ 2p^6\ 3s^2$ RR cross sections

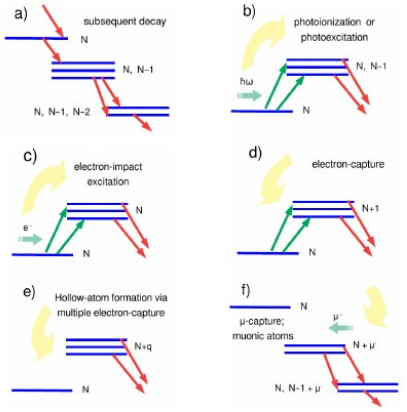
Desired features

- Intuitive (user) interface for quite different computations

  … similar to the user's research work + readily understandable output;
- Features for dealing with open-shell configurations and applications by just selecting suitable configurations & classes of virtual excitations (excitation & capture);
- Simple selection & control of physical units, both at input and output time;
- Access to different models and approximations.
- Default values, whenever feasible.
- ...

(II) Radiative recombination (RR)

$$A^{q+} + e_s^- \longrightarrow A^{(q-1)+} + \hbar\omega$$



Example: Calculatic

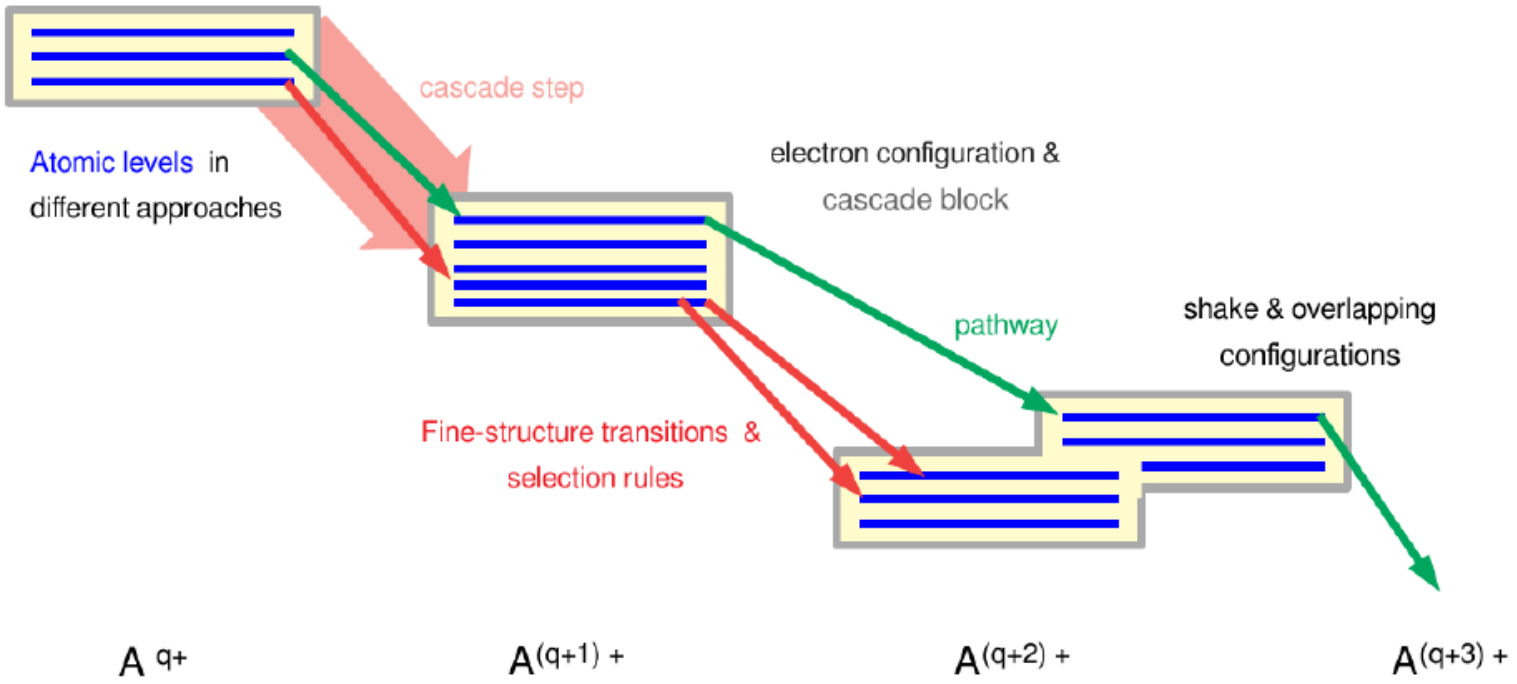
```
> setDefaults("unit: c
> phSettings = Phot


> grid    = Radial.Gri
> name = "Computat

> wa   = Atomic.Co
                Nuclear
                InitialCo
                FinalCo

 > wb   = perform(wa;
```

Why Julia ?

- (Very) fast, high-level language (from MIT, since ~ 2012).
- Combines productivity „and“ performance (not „either“).
- 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, ...

(III) Atomic casades

– key to many astrophysical observations

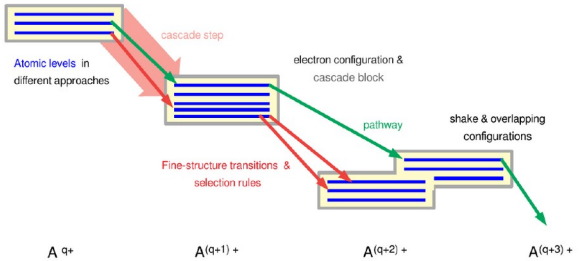


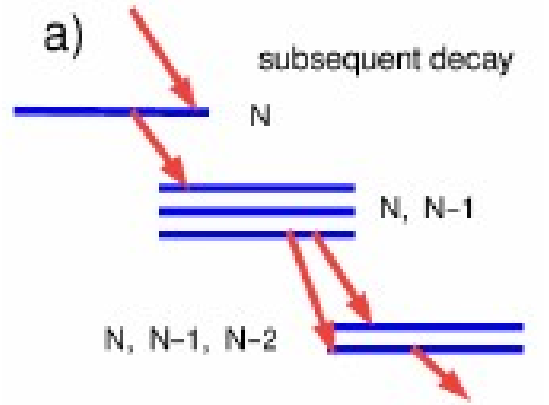
Atomic spectroscopy

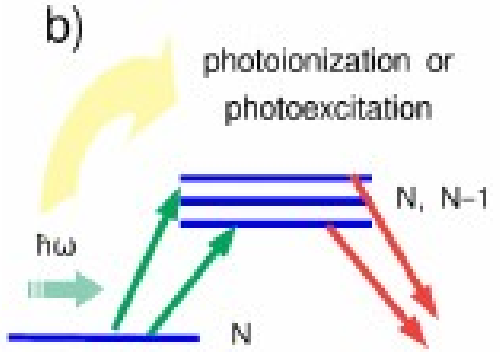
- Ion distributions
- X-ray & photon spectra
- Electron spectra (for different energy regions)
- Coincidence spectra (?)

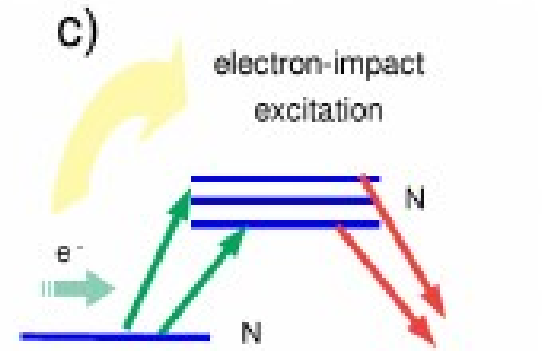
(III) Atomic casades

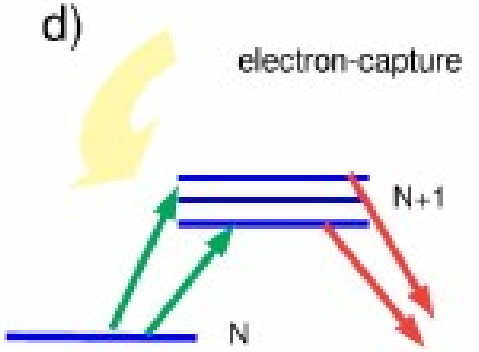
– key to many astrophysical observations

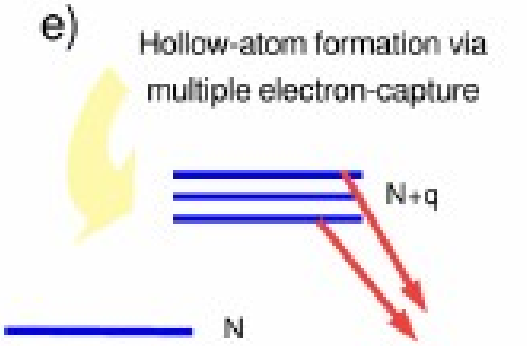


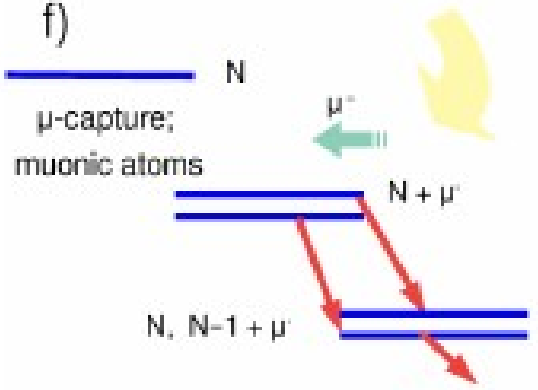






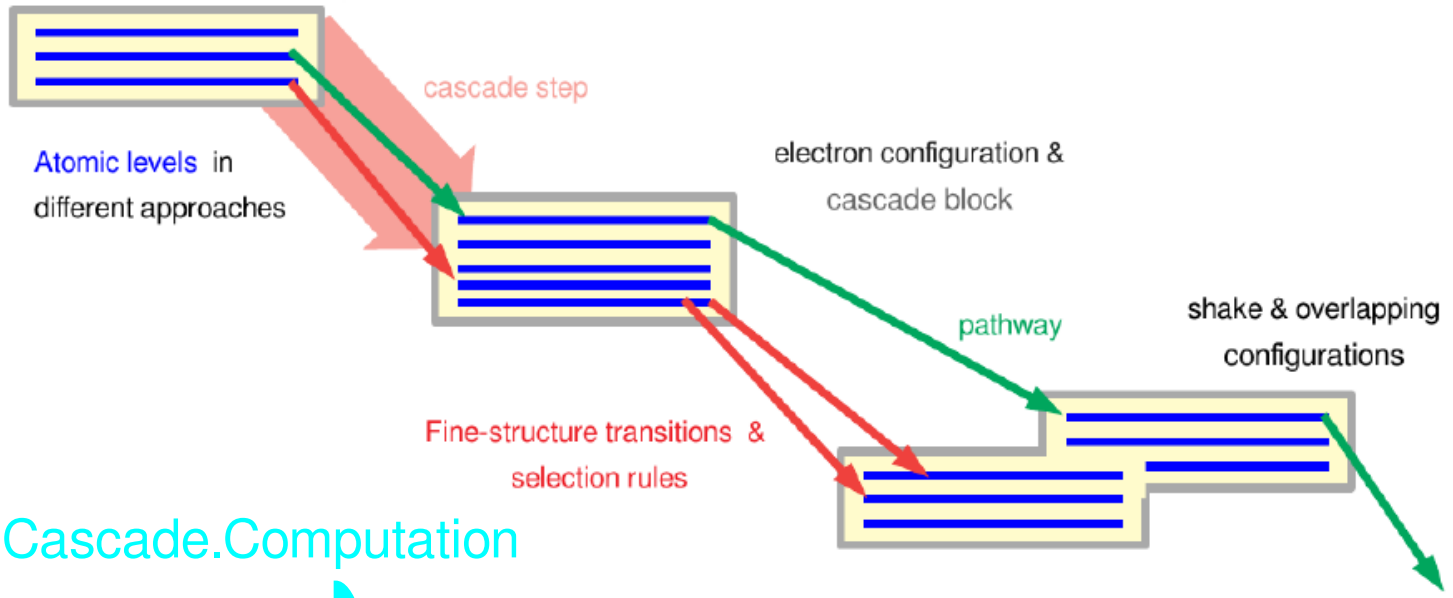






plasma processes & rate coefficients

(III) Atomic cascade computations



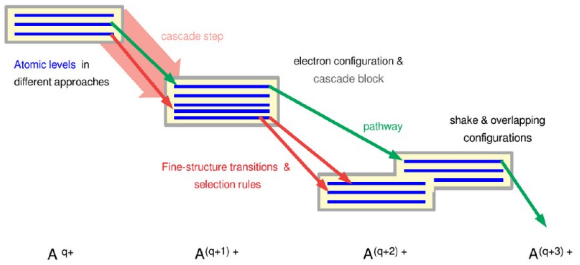
> ? Cascade.Computation

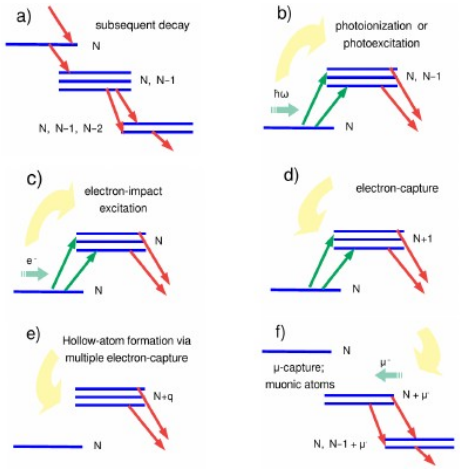
Physics/
stabilization

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.
```

(III) Atomic cascade computations



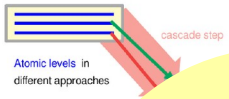


Example: Calculation of the Mg 1s $2s^2$ $2p^6$ $3s^2$ decay cascade after 1s ionization

```
> 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"

> comp = 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(comp; output=true)
```

(III) Atomic casades

– key to many spectroscopic observations



Recent applications

- 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.
- DR strength & plasma rate coefficients: Fritzsche, A&A 656 (2021) A163.
- Near K-edge photoionization of Fe^{q+} ions: Schippers *et al.*, ApJ 908 (2021) 52.
- Multiple photodetachment of oxygen anions: Schippers *et al.*, PRA 106 (2022) 013114.
- Super Coster-Kronig transition of Xe $4s^{-1}$: Hikosaka and SF, PCCP 24 (2022) 17535.
- Strong CI contributions to the Xe $4p^{-1}$ decay: Kosugi *et al.*, PRA 107 (2023) 022814.
- Double core Ar $2p^{-2}$ hole states: Röhrig *et al.*, in preparation (2024).
- Double core Xe $4d^{-2}$ hole states: Hikosaka and SF, just started (2024).
- Multiple photoionization of La^{+} ions: Schippers and coworkers (2024), onging.
- Cascades for astrophysics ?

(III) More atomic casades: Recent work

– classification of different cascade schemes

- Dielectronic capture (scheme) ... for just the (dielectronic) capture & the formation of doubly-excited levels.
- Dielectronic recombination ... for the (dielectronic) recombination of electrons.
- Electron-excitation ... for modeling the direct (EIE) & resonant impact excitation (re-autoionization).
- Electron-ionization ... the same but for ionization (EII + REDA + EIE/autoionization).
- Expansion-opacity ... expansion opacity of an ions in their ground levels.
- Hollow-ion ... decay of hollow ions.
- Impact-excitation ... (direct) electron-impact excitation via collision strength.
- Impact-ionization ... (direct) electron-impact ionization, obtained by empirical models.
- Photoabsorption ... for synthetic photoabsortion spectra (direct + resonant).
- Photoexcitation ... initial photoexcitation, based on inner-shell excitations.
- Photoionization ... for direct photoionization.
- Radiative recombination ... for the radiative recombination (REC).
- Stepwise decay ... standard decay via numerous radiative and Auger transitions.

More atomic casades: Recent work

– classification of different cascade schemes

- Dielectronic capture
- Dielectronic recomb
- Electron-excitation
- Electron-ionization
- Expansion-opacity
- Hollow-ion
- Impact-excitation
- Impact-ionization
- Photoabsorption
- Photoexcitation
- Photoionization

Table 1. Cascade schemes as (partly) implemented and supported by the JAC toolbox. These schemes are internally distinguished by different (concrete) data types `<:` `Cascade.AbstractDataScheme`.

Cascade scheme & brief explanation.
Dielectronic capture (scheme): to model the formation of doubly-excited levels due to the resonant capture of one additional electron, and up top a maximum excitation energy with regard to the ground level of the original ion: $A^{q+} + e^- \longrightarrow A^{(q-1)+*}$; cf. `DielectronicCaptureScheme()`.
Dielectronic recombination: to model both, the formation of and radiative stabilization due to the resonant capture of one additional electron. All doubly-excited levels are taken into account for a given list of subshells and up to a maximum excitation energy: $A^{q+} + e^- \longrightarrow A^{(q-1)+*} \longrightarrow A^{(q-1)+(*)} + \hbar\omega$; cf. `DielectronicRecombinationScheme()`.
Electron-excitation: to model electron excitation spectra including both, the direct EIE and resonant contributions due to the dielectronic capture of an electron with subsequent re-autoionization $A^{q+} + e_i^- \longrightarrow A^{q+*} + e_f^-$ & $A^{q+} + e_i^- \longrightarrow A^{(q-1)+(*)} \longrightarrow A^{q+*} + e_f^-$; cf. `ElectronExcitationScheme()`.
Electron-ionization: to model electron ionization spectra including the direct (EII) and resonant contributions. Resonant contributions may arise from the EIE with subsequent autoionization and from dielectronic capture of an electron with subsequent double-autoionization: $A^{q+} + e_i^- \longrightarrow A^{(q+1)+*} + e_r^- + e_f^-$ & $A^{q+} + e_i^- \longrightarrow A^{q+(*)} + e_f^- \longrightarrow A^{(q-1)+*} + e_a^- + e_f^-$ & $A^{q+} + e^- \longrightarrow A^{(q-1)+*} + e_r^- + e_f^-$; cf. `ElectronIonizationScheme()`.
Expansion-opacity: to model the expansion opacity … excited level, based on a given reference … cf. `ExpansionOpacityScheme()`
Hollow-i…
…ImpactExcitationScheme(). $A^{q+} + e_i^- \longrightarrow A^{q+*} + e_f^-$
…ndard decay via numerical…

Important for astrophysics; very different complexity; systematic treatment of cascades, ...

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

README.md

"just very briefly"

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.

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

Kinds of computations

„just very briefly“

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,

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.
                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.

JAC: Jena Atomic Calculator

— User Guide, Compendium & Theoretical Background —

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

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

~ 850 pages

Community platform for atomic computations

Interactive High-Level Language

JAC

Jena Atomic Calculator

A Julia implementation for atomic computations.

Open-source applications in physics, science and technology.

level energies

playground for electronic structure theory

transition probabilities

Rayleigh & Compt

ionization

(multi-) photon

Rydbe

internal conversion

Electron-impact processes

isotope shifts for nuclear moments

alpha-variations

double (auto-) ionization

Zeeman splittings & Landé factors

electric dipole moments

electron affinities

electron capture

polarizibilities

plasma diagnostics

au & cascades

m factors

energy shifts i

plasma rates

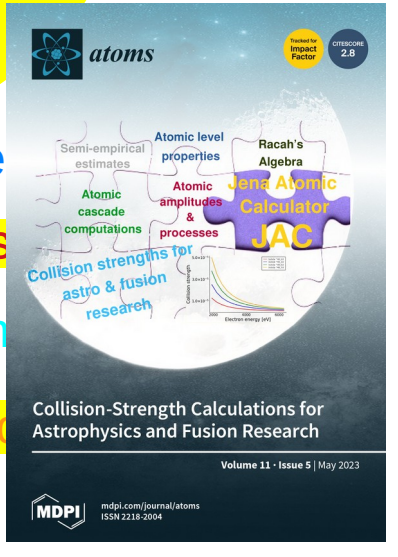
br

dispe

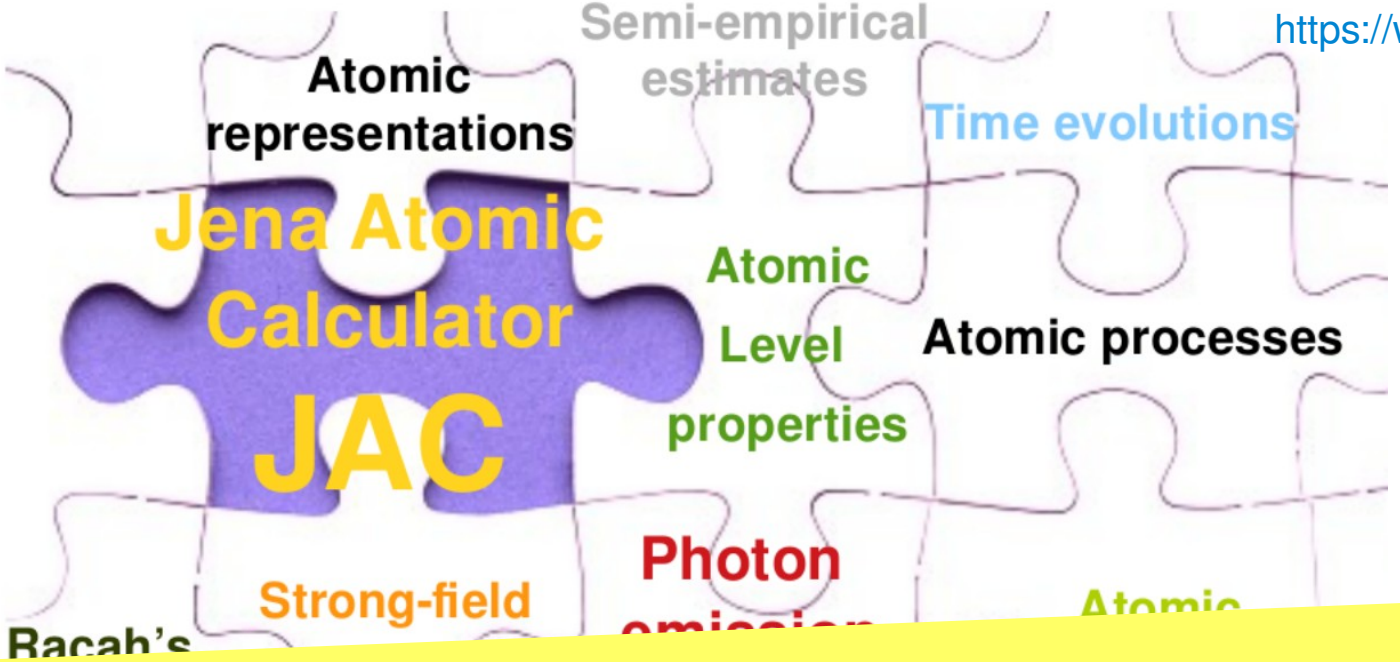
hyperfine s

parity n

Coulomb excitati



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



„stimulus“

Features:

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

Is such a common „computational suite“ possible, desirable & feasible ?

Sure, just try and enjoy !!

What do we need in atomic structure and collision theory ?

– a descriptive language for doing atomic computations ...

Requirements:

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

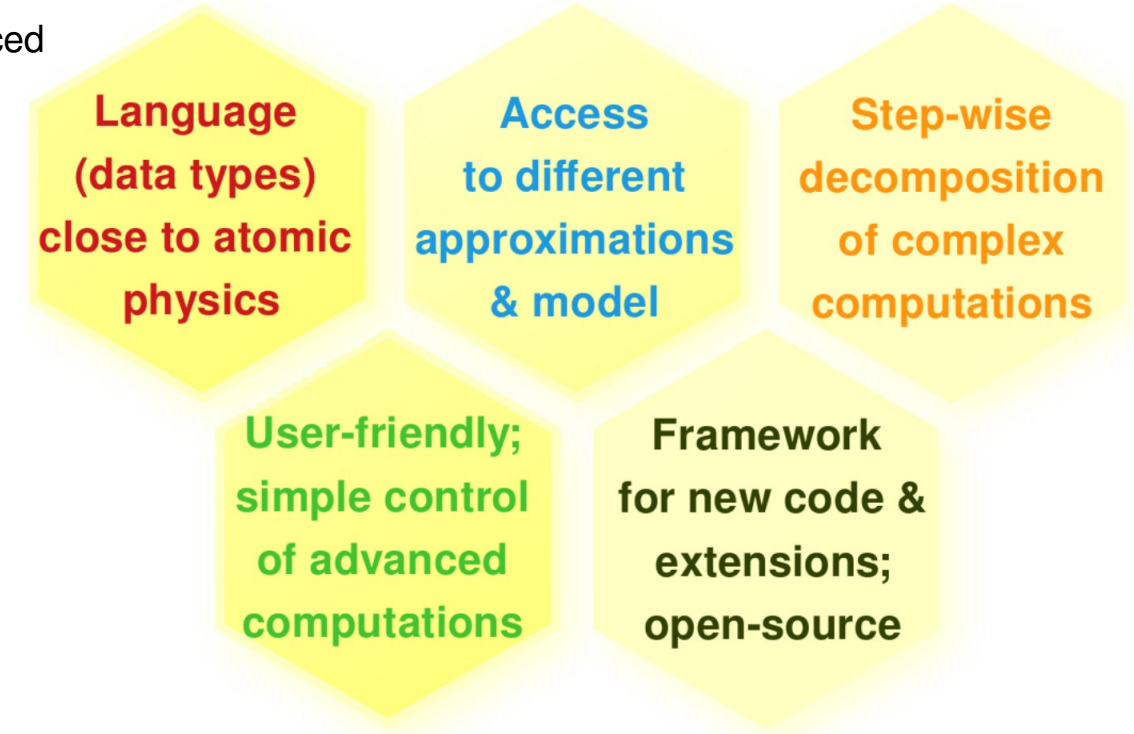
  Shell, Subshell, Configuration, Orbital, Basis, Level, Multiplet, Cascade, Pulse, ...

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

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

- Simple to learn and apply.

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



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:

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

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

- Implementation and c

- Support a coarse-g

  A pseudo-code desc

- Simple to learn and

  With a simplified co
  computations, comp
  atomic cascades; ...

Why Julia ?

- (Very) fast, high-level language (from MIT, since ~ 2012).
- Combines productivity „and“ performance (not „either“).
- 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, ...

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

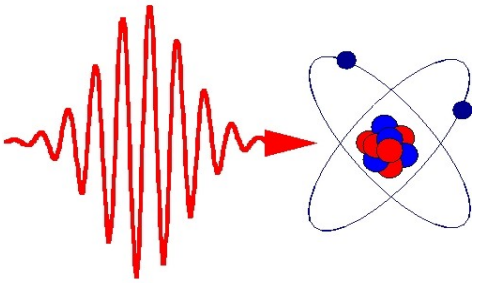
Interactive High-Level Language

JAC

Jena Atomic Calculator

A Julia implementation for atomic computations.

Open-source applications in physics, science and technology.



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.

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