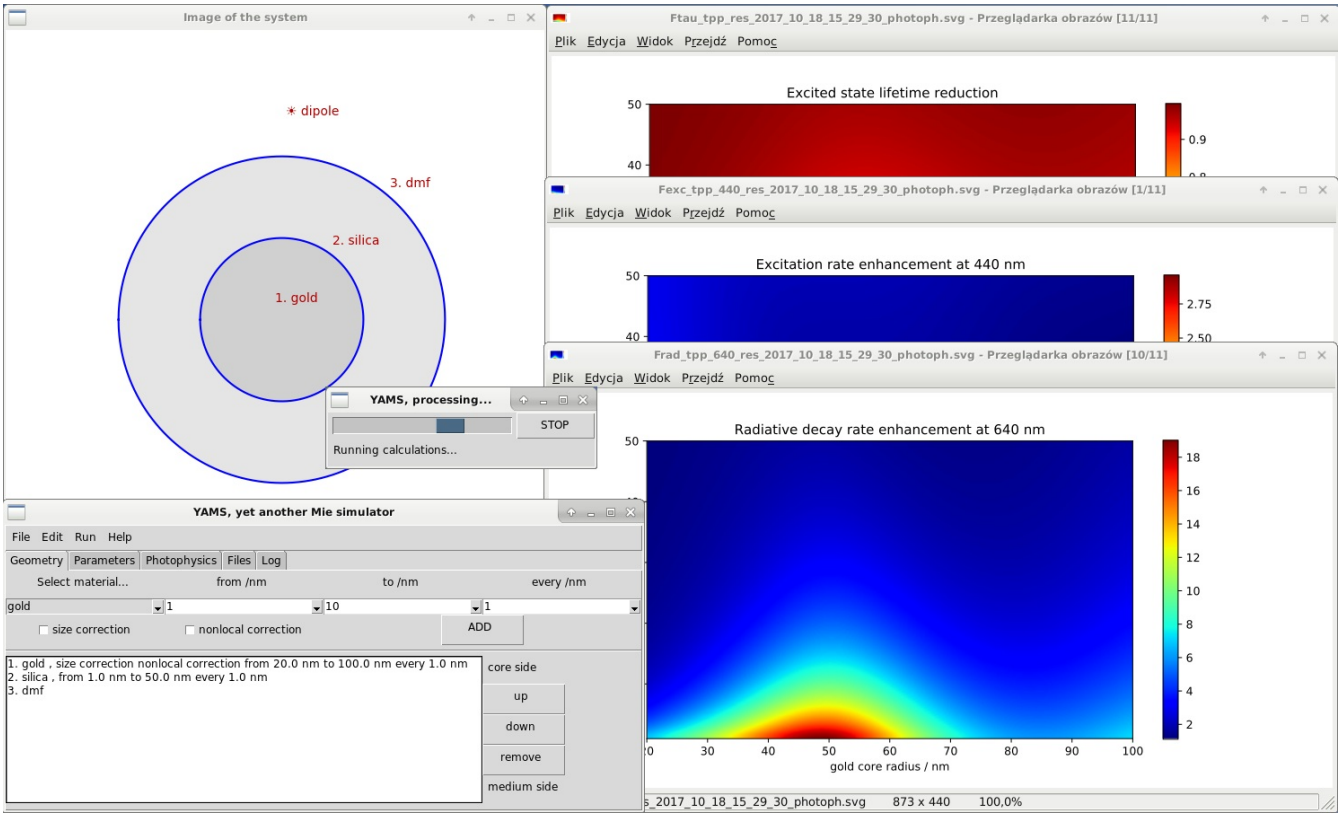
README.md

PDF version of this README is in docs/README.pdf

HTML version of this README is in docs/README.html

YAMS -- yet another Mie simulator

A Python program with GUI to calculate electromagnetic field enhancement factors for plain-wave and dipole radiation in spherical shell geometry. Additionally, it can calculate the subsequent changes of photophysics for given fluorophores.



How it works

Software calculates transfer matrices for electromagnetic field in multipolar expansion for nanoparticles composed of concentric spheres. The method for transfer matrix computation was taken from: Moroz2005.

On the basis of that, field enhancement factor for excitation field (with respect to spatially averaged plain-wave) is calculated, the method was adapted from Le Ru and Etchegoin SPLAC Package. Enhancement of radiative emission rate was calculated as in Moroz2005. Enhancement of total emission rate was obtained by integration of Poynting vector below and under the dipole position.

The longitudinal modes in nonlocal metal dielectric response were treated as in Moroz2005 with metal hydrodynamic constant and longitudinal wave vector taken as in Raza2015 (omitting the size-correction part). The changes in electron mean-free path due to the surface scattering is obtained from: Moroz2008.

I recommend the following positions that explain the physics behind these calculations:

- LeRu2008
- Novotny2012

Similar software

There is quite a lot software on numerical field computation, both commercial and open source. Referring to Mie theory for field computation in spherical shell systems, one can find:

- Le Ru and Etchegoin SPLAC Package for MATLAB,
- Alexander Moroz Fortran codes (compiled, too)
-

Installation

Running with Python interpreter (recommended)

Prerequisites

- Python 3.5.3 (although any Python 3.x should do) On Linux Python 3.x should be available in your distribution repositories. Debian/Ubuntu:

```
sudo apt-get install python3
```

For Windows, check here: https://www.python.org/downloads/

- NumPy >= 1.13.3
- SciPy >= 0.18.1
- scikit-learn >= 0.19.0
- yaml >= 3.12
- matplotlib >= 2.1.0 (assuming you need plotting)
- psutil >= 5.0.1

Since Python 3.4, `pip` is included by default, so in order to install the above Python packages you need:

```
python3 -m pip numpy scipy sklearn yaml matplotlib psutil
```

For Python < 3.4, I guess you first need to get `pip` from https://pip.pypa.io/en/stable/installing/.

Running

Enter the folder `yams` and run:

```
cd yams
python3 ./yams.py
```

Utilizing package components

The following may turn useful for those who might need some scripting. 'The heart' of the package is `yams/fluoroph1layer2.py` to which you can pass input files, save name and chosen fluorophores directly. Check more in the comments of `yams/fluoroph1layer2.py`.

Running from binaries (Linux amd64)

Download tar.gz archive(~70 MB), and extract it:

```
tar -zxvf yams-amd64.tar.gz
```

Run it either double-clicking on the `dist/yams` file or from the command line:

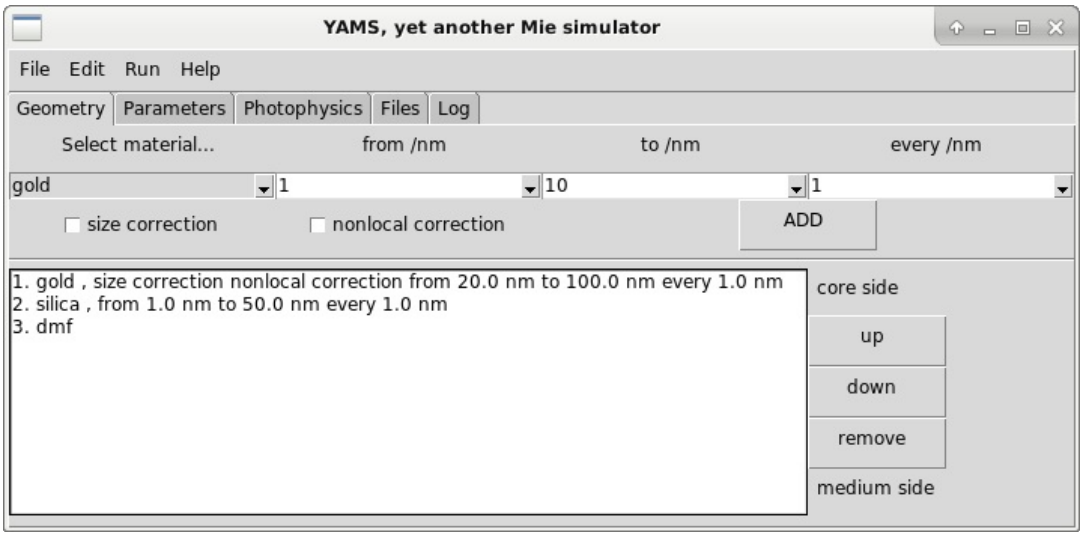
```
cd dist/yams
./yams
```

Tested on fresh install of Ubuntu 16.04.3 Desktop (Xenial Xerus). **Pros**: No need to install Python and extra packages. **Cons**: Archive is around 70MB because YAMS was written in pure Python and binaries were made with PyInstaller. That also explains performance.

Usage

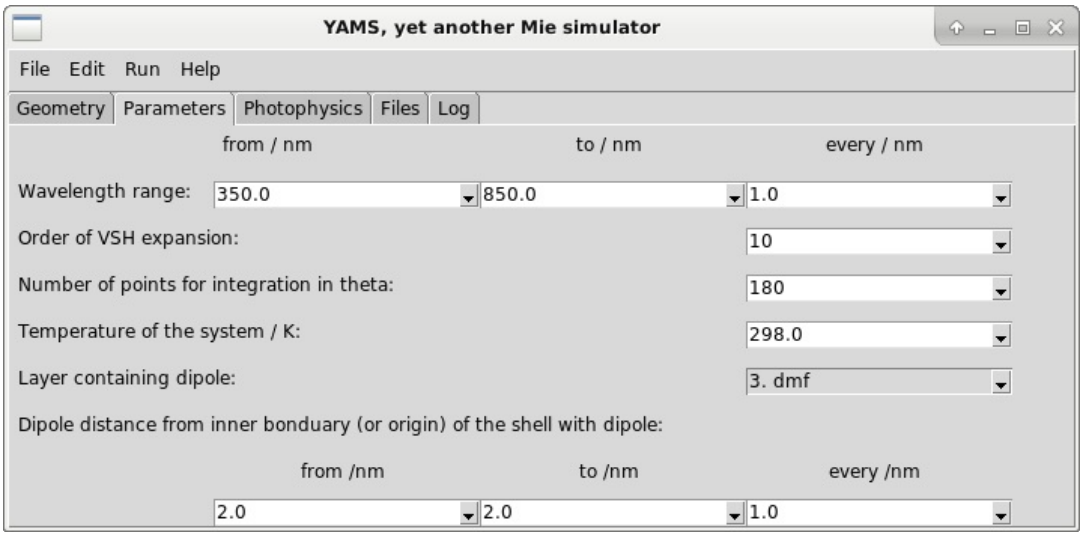
User interface

Geometry tab



Here you can design your core-shell nanoparticle by adding layers of desired size ranges. For noble metals, you can choose whether to apply size correction and nonlocal corrections in transfer-matrix calculations. Inside the list, you can change the layers order by *drag-and-drop* method. `Run > Preview` will show the image of the nanoparticle.

Parameters tab



Wavelength range should cover emission of the chosen fluorophores. Please mind that most of the refractive indices provided end at 850 nm.

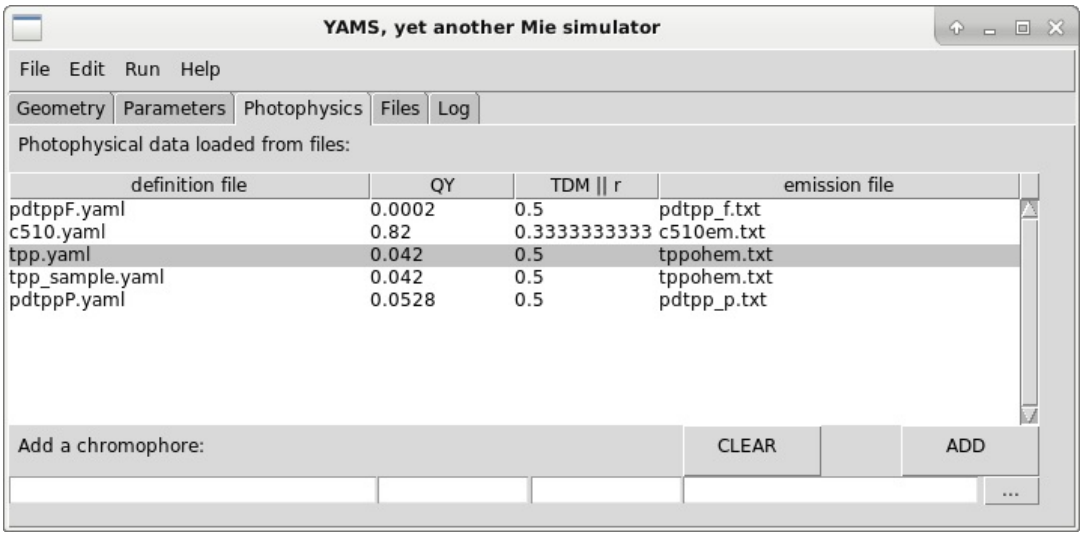
Vector spherical harmonics (VSH) expansion should be sufficient for the orders below 10. However, that might not be enough when dipole is very close to gold surface.

Number of points for integration in theta refers to averaging of excitation field enhancement over θ angle (0 - 180 °) around the nanoparticle, which has to be done numerically.

Temperature of the system -- the default temperature is 298 K. Differences due to temperature change are applied through:

- Modified dielectric function of metal (provided that the metal properties are set in `pkg_resources/mat_sizecor.yaml` file),
- Modified density (and hence dielectric function) of the medium (provided that solvent relative density for this temperature is in appropriate file in `pkg_resources/rho/` and there is the entry for the solvent in `pkg_resources/mat_tempcor.yaml`.

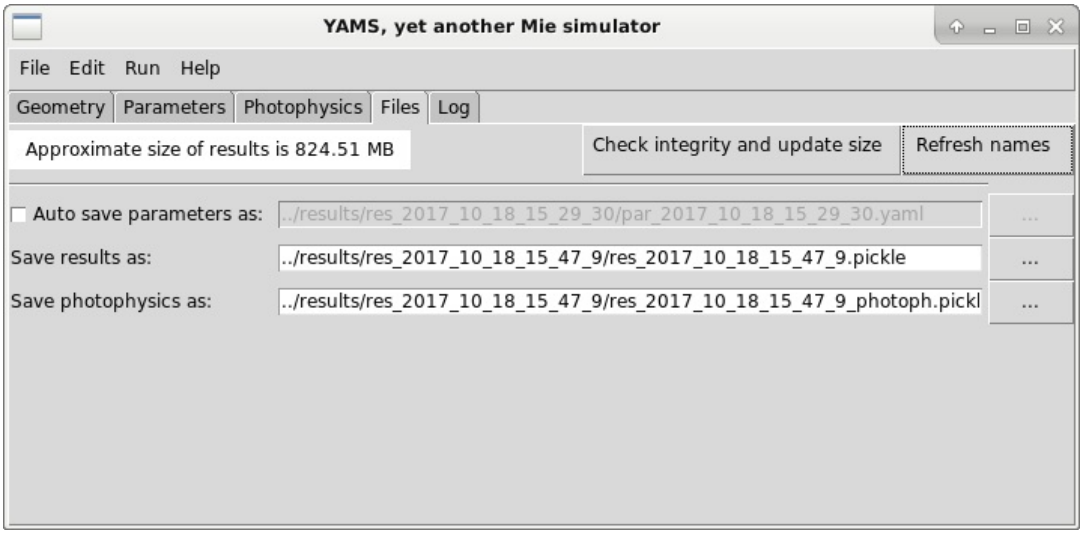
Photophysics tab



Here you can choose one or more fluorophores for computation on how their their photophysics in influence by the presence of the nanoparticles. Fluorophores listed there are found by the software on launch in the directory `pkg_resources/photophysics/`.

You can add new fluorophore using entry field in the bottom. New file in `pkg_resources/photophysics/` will be then created and emission spectrum will be copied there.

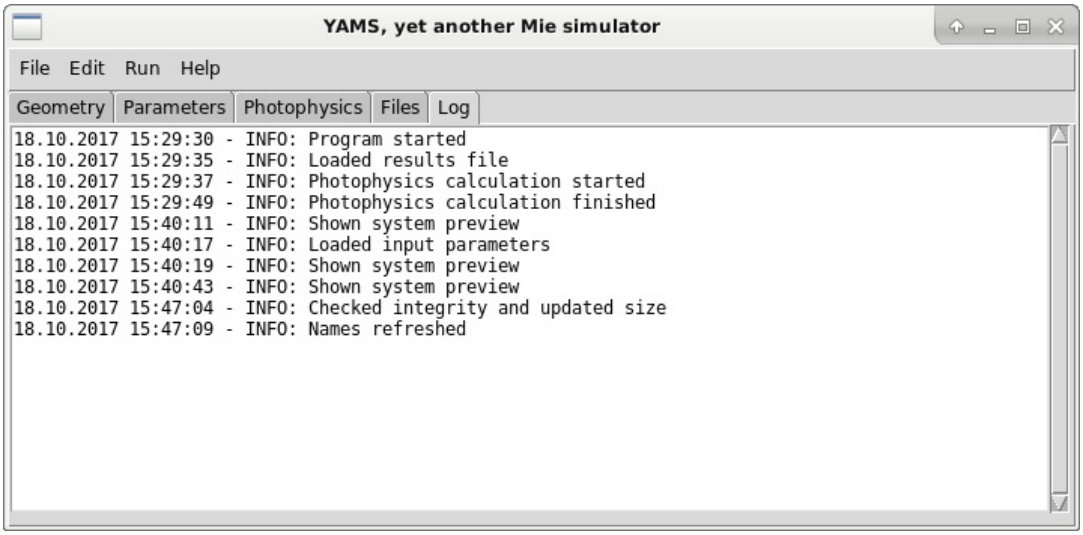
Files tab



Here you can:

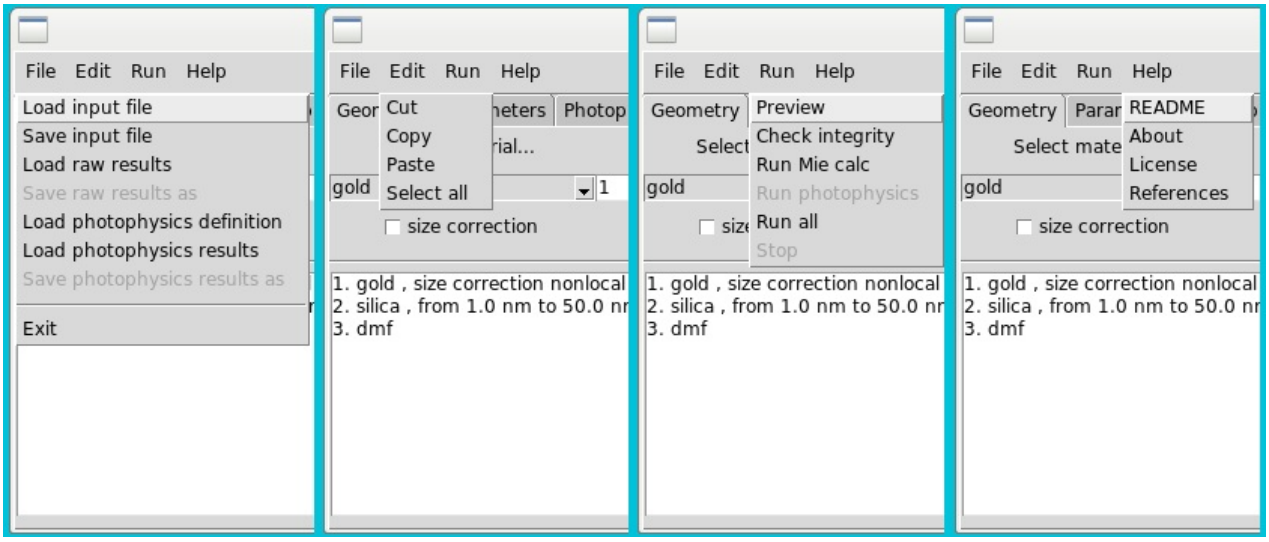
- Check if the computation parameters are correct and what will be the size of the results: button `Check parameters and update size`,
- Refresh the filenames generated by program with correct date/time,
- Set the custom filenames and location of result files.

Log tab



Log tab provides the information on program operations during current session. Unfortunately, it does not collect system warning and error messages.

Menus



File menu allows to:

- load predefined geometry input file,
- save current geometry as input file,
- load *raw* results from Mie calculation (in either .mat or .pickle format),
- save previously loaded (or just calculated) *raw* results (in either .mat or .pickle format),
- load photophysics definition file (which will copy the file and emission spectrum to `/pkg_resources/photophysics/`),
- load photophysics results (in either .mat or .pickle format),
- save previously loaded (or just calculated) photophysics results (in either .mat or .pickle format).

Run menu allows to:

- show current geometry as an image,
- check the correctness of the input parameters,
- run Mie calculation,
- run photophysics calculation (assuming you have loaded or just calculated Mie calculation results),
- run both, that is Mie and photophysics,
- stop running calculation.

Help menu allows to:

- view *.pdf* version of README,
- display information about the software,

- display licence,
- view all references in *.pdf*.

Examples

Input files

All input files, as well as settings file are give in `yaml` format. Current specifications of the format are given here. In fact, on the basis of existing files one can create their own without checking `yaml` specifications.

Geometry input files

Sample geometry files are contained in `input_files` folder and it is advised to save the newly created files also there. The files contain all the information needed to compute transfer matrices and enhancement factors. Sample geometry file with comments is in `input_files/c510test_sample.yaml`. In doubt, you can always save input file generated in GUI by choosing `file > save input as` or load the file by `file >load input file`.

Photophysics definition file

Photophysics definition files consist of:

- quantum yield,
- orientation of transition dipole moment with respect to the radial direction,
- a name of the file with emission spectrum.

For detailed specifications please check `pkg_resources/photophysics/tpp_sample.yaml`. If you want to add new fluorophore (not using the GUI), you just create a new file. There is no need to add information about photophysics in any other configurations files.

Output files

Software typically returns output in two formats: `.pickle` and `.mat`. I found `.txt` not suitable for large n-dimensional matrices.

Mie calculation

Output from Mie calculation consist of:

- *param* -- variable with contents of the input,
- *dip_range* -- vector of distances between the dipole and closest inner layer,
- *rho_rel* -- relative density of solvent at provided temperature,
- *results* -- list/cell array of results. Each position in list is a different set of layers sizes from the provided input.

Each result entry consist of:

- *Ca_dict* -- positions of the layers,
- *dip_range* -- absolute positions of the dipole,
- *QabsM*, *QscaM*, *QextM* -- absorption, scattering and extinction efficiencies of the nanoparticle, calculated from backward transfer matrices, as a function of wavelength (in nm),
- *QabsT*, *QscaT*, *QextT* -- absorption, scattering and extinction efficiencies of the nanoparticle, calculated from forward transfer matrices, should in principle be the same as *QabsM*, *QscaM*, *QextM*,
- *Fexcperp*, *Fexcpara* -- enhancement of the excitation (plane-wave) field for orientations along the radial direction and parallel to nanoparticle surface, respectively. Enhancement factors are averaged over nanoparticle surface.
- *MRadPerp*, *MRadPara* -- radiative part of the enhancement of dipole emission,
- *MTotPerp*, *MTotPara* -- total enhancement of dipole emission, non-radiative energy transfer to the nanoparticle,
- *MNRPerp*, *MNRPara* -- non-radiative part of the enhancement of dipole emission, calculated as *MTot* - *MRad*.

The shape of enhancement factors matrices is [number of dipole positions] x [number of wavelengths]. The enhancement factors are given with respect to mean incoming plain-wave/dipole field intensity.

Photophysics

Output of photophysics calculation consists of:

- *param*, *dip_range*, *rho_rel* as in **Mie calculation**,
- *QabsM*, *QabsT*, *QextM*, *QextT*, *QscaM*, *QscaT* -- efficiencies matrices reshaped, to `[layer0 size] x [layer1 size] x ... x [layerN size] x [wavelength]` and multiplied by *rho_rel* to express the observed changes in the spectra,
- *Fexc_fluorophore* -- excitation enhancement for orientation of given fluorophore, reshaped to `[layer0 size] x ... x [layerN size] x [dipole positions] x [wavelength]`. This should give the change of exciation spectrum.
- *Frad_fluorophore* -- radiative decay rate enhancement as a function of emission wavelength, for given fluorophore orientation, the same size as *Fexc_fluorophore*. This should give the change of emission spectrum shape.
- *FkRad_fluorophore* -- radiative decay rate enhancement integrated over fluorophore emission spectrum, the shape is: `[layer0 size] x ... x [layerN size] x [dipole positions]`,
- *FkTot _fluorophore* -- total decay rate enhancement integrated over fluorophore emission spectrum, the shape is as in *FkRad_fluorophore*,
- *Ftau_fluorophore* -- reduction of fluorophore excited state lifetime, taking into account *FkTot _fluorophore* and its intrinsic non-radiative decay rate,
- *FQY_fluorophore* -- enhancement of fluorophore quantum yield, *FkRad* * *Ftau*,
- *FGamma_fluorophore* -- enhancement of the observed fluorophore emission, *Fexc* * *FQY*.

Additionally, plots for chosen wavelengths for *Fexc*, *Frad*, *FGamma* and for *Ftau* and *FQY* are generated. They are meant to provide quick insight into the results rather than serve as publication-ready figures. The plotting can be switched off in `pkg_resources/settings.yaml`, where you can also change the chosen wavelengths.

Advanced

Settings

Advanced settings are in `pkg_resources/setting.yaml`. These include multiprocessing, vector spherical harmonics (VSH) series truncation in case on non-convergence and plotting options. Check comments of `pkg_resources/setting.yaml`.

Adding refractive index

1. Prepare .txt file with complex refractive index as a function of wavelength (in nm). Put the file in `pkg_resources/ref_ind` directory.
2. Add entry to `pkg_resources/materials.yaml`:

```yaml
material:
    file: file_without_txt_extension    # file with refractive index
                                        # location: resources/ref_ind/file_without_txt_extension.txt
                                        # first column in wavelength in nm
                                        # second column is complex refractive index
                                        # the field cannot be empty
    ref: key_of_reference               # reference to paper with published refractive index
                                        # key in pkg_resources/references/RI.bib
                                        # the field can be left empty
```

3. (Optional) Add reference entry to `pkg_resources/references/RI.bib`.

Adding corrections for metal

Corrections for metal include:

- correction for surface scattering of electrons (limited mean-free path),

- correction for temperature to dielectric function.
- correction for lonlocality of dielectric function.

Corrections can be included by adding appropriate entry in `pkg_resources/mat_sizecor.yaml`. Entry consist of material constants in the right units, check comments in `pkg_resources/mat_sizecor.yaml`.

Adding temperature correction for medium

Solvents increase their density and refractive index with decreasing temperature. The changes in refractive index are calculated basing on density changes using Eyckmann's formula, check eq. (2) in Mantulin1973. To include temperature changes on refractive index one needs the relative (to 298 K) densities of the solvents for each temperature. The file consist of a column with temperatures (in Kelvin) and a column with relative densities. Location of the files in `pkg_resources/rho`. New entry for the solvent should be made in `pkg_resources/mat_tempcor.yaml` (just like in `pkg_resources/materials.yaml`).

Credits

Contact: annamariakelm@gmail.com, GitHub: https://github.com/akelm

How to cite

Kelm, A. (2017). YAMS -- yet another Mie simulator [Computer software]. Available from: https://github.com/akelm/YAMS BibTeX file

References

[1] Konrad, A.; Wackenhut, F.; Hussels, M.; Meixner, A. J.; Brecht, M. *J. Phys. Chem. C* **2013**, 117, 21476–21482. [2] Le Ru, E.; Etchegoin, P. *Principles of Surface-Enhanced Raman Spectroscopy and Related Plasmonic Effects*; Elsevier Science, **2008**. [3] Liu, M.; Pelton, M.; Guyot-Sionnest, P. *Phys. Rev. B* **2009**, 79, 035418. [4] Moroz, A. *J. Phys. Chem. C* **2008**, 112, 10641–10652. [5] Moroz, A. *Ann. Phys.* **2005**, 315, 352–418. [6] Novotny, L.; Hecht, B. *Principles of Nano-Optics; Cambridge University Press*, **2012**. [7] Raza, S.; Bozhevolnyi, S. I.; Wubs, M.; Mortensen, N. A. *J. Phys.: Condens. Matter* **2015**, 27, 183204.

You can also find the following `.bib` reference files:

- pkg_resources/references/general.bib for references on which program is bases,
- pkg_resources/references/RI.bib for the sources of refractive indices,
- pkg_resources/references/rho.bib for the sources of temperature-dependent densities.

All the references are in file: pkg_resources/references/references.pdf

License

Copyright (C) 2017 Anna Kelm

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see http://www.gnu.org/licenses/.

License file

To do

- ☐ Total energy extracted from dipole from Green functions in section 9. in Moroz2005
- ☒ x-axis label in plotting when one variable is dipole position
- ☒ change base path explicitly to the one of yams/yams.py, regardless of from which directory it is opened
- ☒ correct size of the results
- ☒ key bindings for starting calculations
- ☒ add icon