Modeling nanoparticle scattering with QNM, based on COMSOL—Matlab simulations.

Steps:

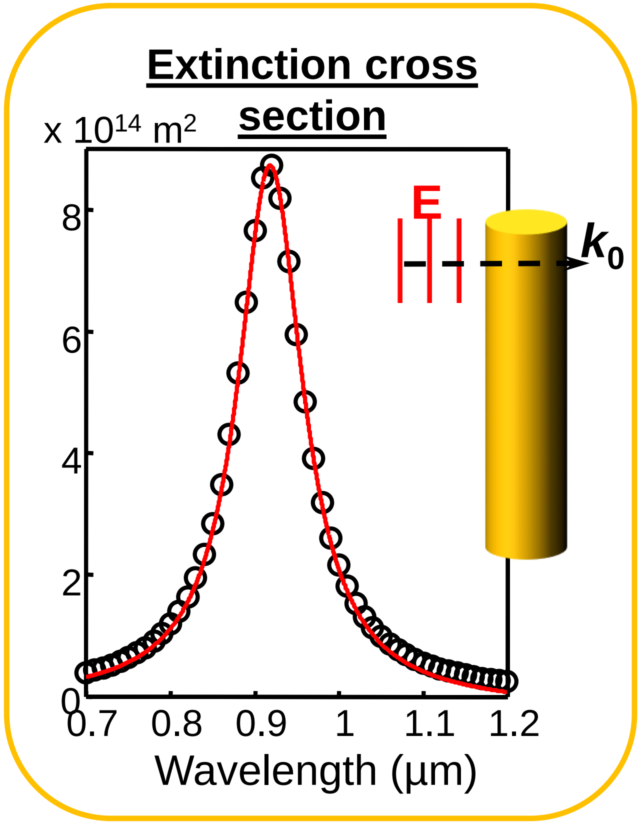
I. Set up the COMSOL model sheet

II. Run COMSOL with Matlab via COMSOL Livelink module

III. Computation and normalization of QNM field (*ScriptQNM_web.m*)

IV. Calculate cross-sections using QNM data (*Script_cross_sections_QNM.m*)

V. Calculate cross-sections using fully-vectorial calculations with COMSOL (*Script_cross_sections_FV.m*)



Reference

Q. Bai. *et al*, Efficient and intuitive method for the analysis of light scattering by a resonant nanostructure. Opt. Express **21**, 27371-27382 (2013).

I. Set up the COMSOL model sheet

If you want to use your own COMSOL model sheets with the QNM programs

- Your model sheet **must be completely defined and runnable directly in COMSOL**

- The QNM program will **perform on its own the modifications in the model sheet required for it to be run with complex frequencies** *(cf Reference).*

- If you define frequency-dependent materials' permeability and permittivity, **the frequency variable name you used to defined your materials** (ex; epsilon(omega)=…) **will be used in the programs**, so this variable **must be the same for all materials**.

- The **programs assume a simple model sheet** with only one "frequency domain" study *"std1"* (it will create a 2nd one ; *"std2"* when required), one geometry *"geom1",* one mesh *"mesh1",* one electric point dipole *"epd1",…* . **If your model does not respect these conditions, please modify the corresponding variable names in the QNM programs** (at the beginning of Initialization sections) accordingly to your model sheet.

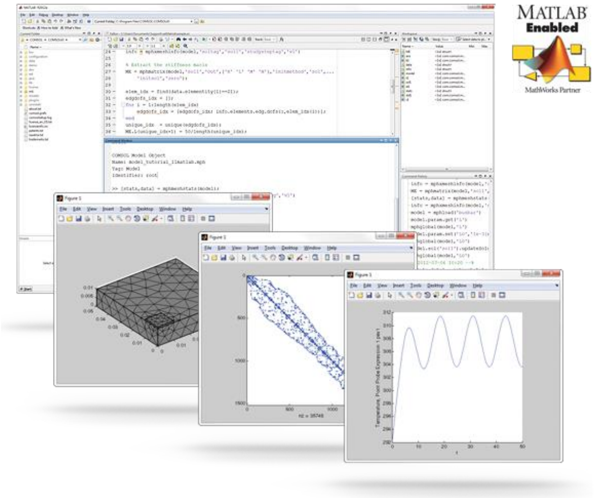
- **You can exploit planar (anti-)symmetries** of your simulated system to reduce the computation time. This will be taken into account in the QNM program **via a variable 'sym_factor'** ('1'=no symmetries, '2' = 1 symmetry, '4' = 2 symmetries, and '8' = 3 symmetries).

II. Run COMSOL with Matlab via COMSOL Livelink module

- To use the QNM programs, **you must have COMSOL with Matlab Livelink module, and Matlab installed on the same machine**.

MATLAB

COMSOL MULTIPHYSICS®



```
info = mphxmeshinfo(model,'soltag','sol1','studysteptag','v1');

% Extract the stiffness matrix
MK = mphmatrix(model,'sol1','Out',{'K' 'L' 'M' 'N'},'initmethod','sol',...
    'initsol','zero');

elem_idx = find(data.elementity{1}==2);
edgdofs_idx = [];
for i = 1:length(elem_idx)
    edgdofs_idx = [edgdofs_idx; info.elements.edg.dofs(:,elem_idx(i))];
end
unique_idx = unique(edgdofs_idx);
MK.L(unique_idx+1) = 50/length(unique_idx);
```

```
COMSOL Model Object
Name: model_tutorial_llmatlab.mph
Tag: Model
Identifier: root

>> [stats,data] = mphmeshstats(model);
```

- Be aware that the QNM programs **were written using COMSOL 4.4 and Matlab 2013a, and might not be usable with older versions of either software**.

III. Computation and normalization of QNM field

- Open the Matlab script **“ScriptQNM_web.m”,** and make sure that the Matlab functions **“setCOMSOL_ComplexFreq.m”** (adapt the model sheet to complex frequency) and **“omega_generation.m”** (generate the pole evaluation freauency triplet) **are in your Matlab working path.**

- Check parameters;
  - ***Physical parameters;*** the initial guess of your complex QNM pole frequency
  - ***Numerical critical parameters;*** filenames to save data (we save parameters in a *.mat* file, and field data in a new *.mph* model sheet), initial frequency triplet shift (f0=initial guess, f1=initial guess*(1-delta), f2=initial guess*(1+delta)), and symmetry factor.
  - ***COMSOL parameters***; model filename, variable names (frequency, and background material), simulation stop condition (if COMSOL solver takes too many iterations without converging, the program stops and assumes it found the pole frequency)
  - ***Numerical parameters;*** stop condition for the iterative process (max number of pole frequency estimations allowed), field evaluation parameters (position of evaluation point, field component evaluated), field xOz slice image parameters (position of the window, and resolution). If you want another plane for the slice, modify the program in the Initialization section .

- **Run the program**; it **iteratively approaches the pole frequency** where the simulated field theoretically diverges (there is a plot showing this divergence as the iterative process converges towards the pole).

- For **each iteration**, the program **calculates the total** and **background fields,** to calculate the scattered field, necessary to normalize the QNM.

- When the **iterative process converges towards a pole frequency** (COMSOL cannot calculate the field for this frequency), the program **displays the convergences of the estimated pole frequency** and **the normalized field, the divergence of the calculated field,** and **the final normalized field on the xOz plane.**

- **Progression information** is displayed **in Matlab command window,** and **via COMSOL progress window.**

- If you want to **stop the loop**, **cancel the ongoing simulation via COMSOL progress window.** This way, if you stop the simulation because it takes too long to reach your stop condition (too loose COMSOL-solver stop condition), the Matlab program can still continue and normalize the field.

IV. Calculate cross-sections using QNM data

- **Open the Matlab script “Script_cross_sections_QNM.m”**

- You can **only use this program** if you previously **calculated and normalized a QNM using *“ScriptQNM_web.m”*** because of the specific data format.

- This program calculates the **scattering and absorption cross-sections for a z-polarized plane wave, propagating along x-axis** (cf Reference). Beware of symmetries (cf Script)

- It **can only be used if the resonator is homogenous**, but can be modified to work in a more general case where the resonator is composed of 2 or more materials.

- Check parameters;
  - ***Numerical parameters;*** filename to save data (save parameters and cross-sections in a *.mat* file), filename of the *.mat,* normalization file (we retrieve a lot of parameters for the new calculations directly from this file), name of the material composing the resonator (tag in COMSOL model sheet), and wavelengths for which you want cross-sections to be calculated.

- **Run the program**; it first **initializes COMSOL numerical methods** to calculate volume integrals and other quantities, then **calculates overlap integrals in loop on the wavelength**, and finally **calculates cross-sections and plot them**.

IV. Calculate cross-sections using fully-vectorial calculations

- Open the Matlab script **“Script_cross_sections_FV.m”**

- You can **use this program independently of the 2 other programs.**

- This program calculates the **scattering and absorption cross-sections for a z-polarized plane wave, propagating along x-axis**.

- **In the model sheet** used by this program, you need to **define a closed surface around the resonator (not overlapping with the PMLs), with a mesh dense enough** to calculate accurately the Poynting vector flux (then the scattering cross-section) and **a geometrical selection attached to this surface**. We advise to define this surface as a sphere, and the selection as an “explicit selection ” of all the boundaries of this sphere.

- Here again, the **programs assumes a simple model sheet (variable name),** but there must be **no active source in it**. Indeed, the program will **automatically set the study in a “scattered field formulation ”** and **define the plane wave** in the model sheet.

- **You cannot exploit any symmetries of your system in this program,** unless you modify the code in this purpose.

- This program **can only be used if the resonator is homogenous**, but can easily be modified to work in the case where the resonator is composed of 2 or more materials.

- Check parameters;
  - ***Numerical parameters;*** filename to save data (save parameters and cross-sections in a *.mat* file), and wavelengths for which you want cross-sections to be calculated.
  - ***COMSOL parameters;*** model filename, variable names (frequency, background material, resonator material, and integration surface selection)

- **Run the program**; it first **initializes COMSOL**, then **runs field simulations and calculates cross-sections in loop on the wavelength,** and finally **plots the results**