# Instructions for analytical calculations of the Fano asymmetry parameter and intensity in light scattering problems

The main theoretical results were obtained in [1]. If these results and Python or MATLAB scripts were useful for your research, please make a corresponding reference for [1] in your article.

The technical implementation of the quasi-normal mode (QNM) theory is well developed and fully automated in the "MAN" (Modal Analysis of Nanoresonators) package (https://zenodo.org/records/7400937), which is written in MATLAB. The scripts below are simple Python adaptations of the "MAN" scripts. We strongly recommend using MATLAB with "MAN" due to its broader functionality, but for those more familiar with Python, these scripts can be a good introduction to QNM and Fano resonances.

Below, we briefly describe the procedure for *ab initio* calculation for the Fano asymmetry parameter $q_m$ and intensity $\sigma_m$ of a single m$^{\text{th}}$ QNM in light scattering. This instruction applies to three examples:

- **Example 1**: Split-ring resonator in air.
- **Example 2**: The same split-ring resonator on a substrate.
- **Example 3**: Plasmonic dolmen in air.

Procedures for all examples are very similar with some specific differences, which are described below.

For Python the calculations are semi-manual and consist of three steps: (I) Calculations in COMSOL; (II) Data export and (III) Python calculations.

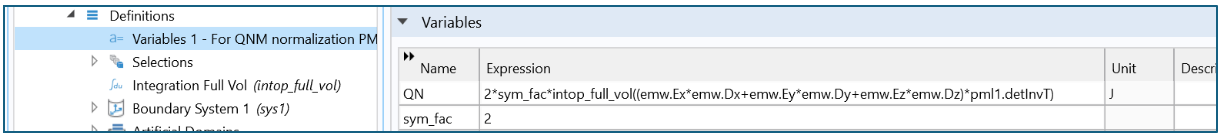
---

**Example 1**. Split-ring resonator in air;

In this example, we studied light scattering from dielectric split-ring resonator embedded in air (see the Supplementary information, Section III.1 in [1]).

**(I) Calculation in COMSOL.**

Create a standard model for eigenfrequency calculation (see "Example 1. QNMs Split-ring resonator.mph"). The only difference comparing to a standard calculation is that we need to normalize eigenmode fields. Here PML-norm is used (see details in [2]). In the model, PML-norm is calculated as a variable "QN", which is defined in next way:



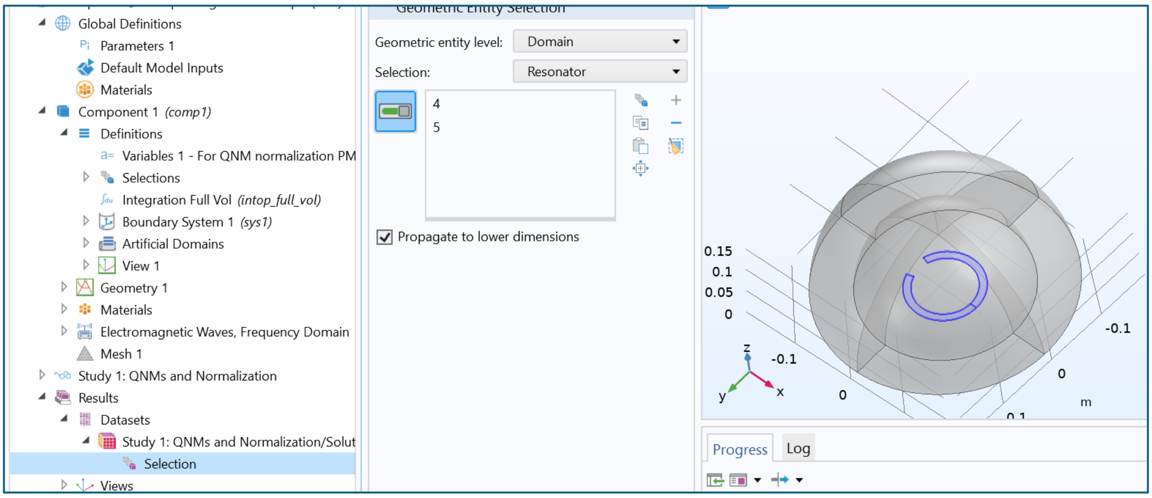

were "intop_full_vol" is volume integration over all domains (including PML).

Perform eigenfrequency calculation.

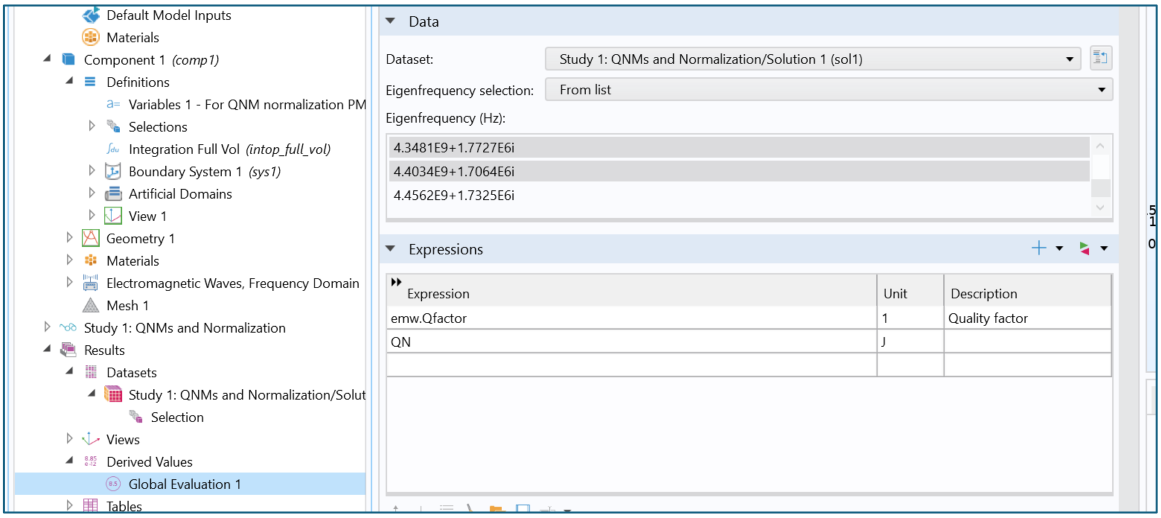
## (II) Data export

The next step is appropriate data export, which is necessary to perform the integration in Python using the Gaussian quadrature. For that purpose, perform the next steps:

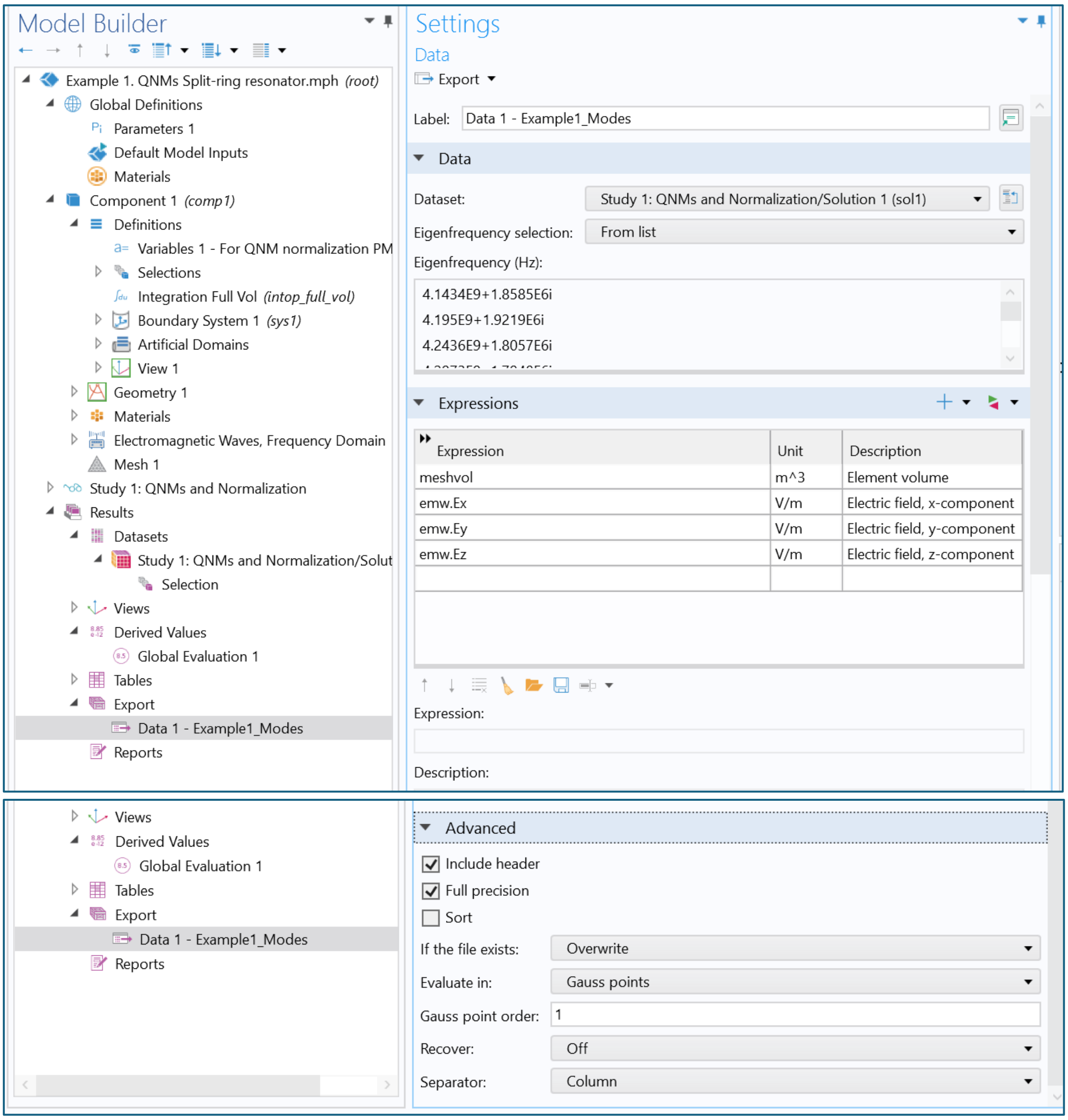
1. In “Datasets” choose data and make a selection, for the selection choose only resonator domains;




2. Perform “Global evaluation” to calculate Q factors and PML-norms (emw.Qfactor, NN), evaluate these in new table and save it in txt-file called “Example1_Norms.txt”;




3. In “Export”, chose corresponding data set with the selection. In “Advanced settings” choose Evaluate in “Gaussian points”. In data to evaluate, choose “meshvol, emw.Ex, emw.Ey, emw.Ez”. Here “meshvol” is the Gaussian weight in Gaussian Quadrature and all quantities will be calculated in Gaussian point. Export this data to txt-file “Example1_Modes.txt”.




4. Note that COMSOL uses “j” for imaginary unit, but Python uses symbol “i”. Change all “j” to “i” in both txt-files.
5. Note that all data have to be in the SI units.

As a result, there will be two data-files “Example1_Norms.txt” and “Example1_Modes.txt”. Note that file “Example1_Modes.txt” is a table with data, which is ordered in next way

| X | Y | Z | meshvol | emw.Ex (mode №1) | emw.Ey (mode №1) | emw.Ez (mode №1) | meshvol | emw.Ex (mode №2) | emw.Ey (mode №2) | … |
|---|---|---|---|---|---|---|---|---|---|---|
| … | … | … | … | … | … | … | … | … | … | … |

Here X, Y, Z – are Gauss points coordinates, which are used to define excitation field $\mathbf{E}_b$ analytically in Python.

**(III) Python calculations**

Python script is used to perform overlap integral calculations (*via* Gaussian Quadrature) and create figures. For the example 1, corresponding Python script is “Example 1. FanoScript - SplitRingResonator.ipynb” (Jupiter Notebook programming environment was used).

The script is simply a reproduction of the theoretical steps derived in the Supplementary Materials, Section I in [1]. Variables correspond to notations from [1] and can be simply recognized. All necessary comments are given in the script. As a result, it calculates the Fano asymmetry parameter $q_m$ and intensity $\sigma_m$ of a single m$^{\text{th}}$ QNM.

Note that the script is general for non-dispersive materials. If one wants to perform calculations for some other particle, the only changes will be geometry and permittivity changes in COMSOL model.

---

**Comment about Example 2.** The same split-ring resonator on a substrate.

For the second example the calculation is very similar to one described above. The only changes are:

- In COMSOL model "Example 2. QNMs Split-ring resonator on substrate.mph" the non-dispersive substrate with refractive index n = 1.75 were introduced.
- In the Python script "Example 2. FanoScript - SplitRingResonatorOnSubstrate.ipynb" the definition of the background field Eb is changed, it accounts incident plane wave and reflected wave in the upper half-plane.

This example reproduced results from the Supplementary information, Section III.3 in [1].

---

**Comment about Example 3.** Plasmonic dolmen in air.

The calculation for this case is more complex, it is necessary to use the Auxiliary field formulation of the QNM theory [3] and COMSOL model need to be modified in an appropriate way (the details can be found in the "MAN" repository by this link: https://zenodo.org/records/7400937). The COMSOL model "Example 3. QNM Plasmonic silver dolmen.mph" for that case was taken from the "MAN" repository [4]. Additionally, Python script "Example 3. FanoScript - Plasmonic dolmen.ipynb" is modified to account frequency dependence of the dielectric permittivity. All other steps remained the same.

This example reproduces results from the Supplementary information, Section II in [1].

---

## References

[1] M. Bochkarev, N. Solodovchenko, K. Samusev, M. Limonov, T. Wu, and P. Lalanne, "Quasinormal modes as a foundational framework for all electromagnetic Fano resonances", [Will be updated], DOI: [Will be updated].

[2] C. Sauvan, T. Wu, R. Zarouf, E. A. Muljarov, and P. Lalanne, "Normalization, orthogonality, and completeness of quasinormal modes of open systems: the case of electromagnetism," Opt. Express 30, 6846-85 (2022).

[3] W. Yan, R. Faggiani, and P. Lalanne, "Rigorous modal analysis of plasmonic nanoresonators," Phys. Rev. B **97**, 205422 (2018).

[4] T. Wu, D. Arrivault, W. Yan, and P. Lalanne, "Modal analysis of electromagnetic resonators: User guide for the MAN program," Comput. Phys. Commun. **284**, 108627 (2023).