

# 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, as well as the provided Python scripts, were useful for your research, please include an appropriate citation of [1] in your article.

The technical implementation of the quasinormal mode (QNM) theory is well developed and fully automated in the MATLAB-based package **MAN** (Modal Analysis of Nanoresonators), available at <https://zenodo.org/records/7400937>

The scripts provided below are simple Python adaptations of the MAN scripts. We strongly recommend using MATLAB with MAN [2] because of its broader functionality; however, for those more familiar with Python, these scripts can serve as a helpful introduction to QNM theory and Fano resonances.

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

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

The procedures for all examples are very similar, with specific differences described in the corresponding sections.

In Python, the calculations are semi-manual and consist of three main steps:

- (I) COMSOL calculations;
- (II) Data export;
- (III) Python-based processing.

---

---

---

## 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.2 in [1]).

### (I) Calculation in COMSOL.

Create a standard model for the eigenfrequency calculation (see *Example 1. QNMs Split-ring resonator.mph*). The only difference from a standard calculation is that we need to normalize the eigenmode fields. In this case, we use the PML normalization (see details in [3]). In the model, the PML norm is computed as the variable “QN”, which is defined as follows:

Definitions			
<i>a=</i> Variables 1 - For QNM normalization PM			
▷ Selections			
<i>fsl</i> Integration Full Vol ( <i>intop_full_vol</i> )			
▷ Boundary System 1 ( <i>sys1</i> )			
▷ Artificial Domains			
Variables			
Name	Expression	Unit	Description
QN	$2 * \text{sym\_fac} * \text{intop\_full\_vol}((\text{emw.E}x * \text{emw.D}x + \text{emw.E}y * \text{emw.D}y + \text{emw.E}z * \text{emw.D}z) * \text{pml1.detInvT})$	J	
sym_fac	2		

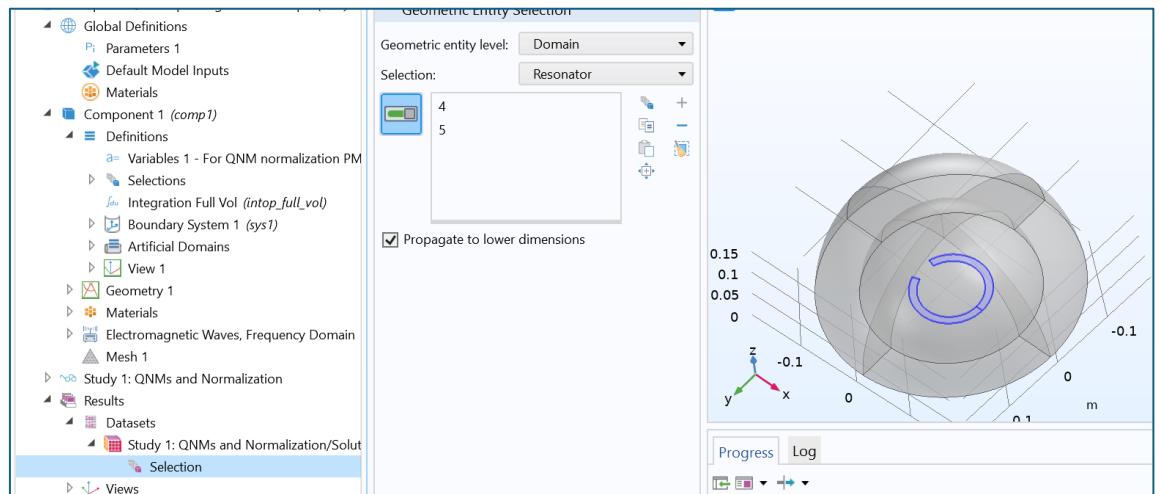
were “*intop\_full\_vol*” is the volume integration over all domains (including PML).

Perform eigenfrequency calculation.

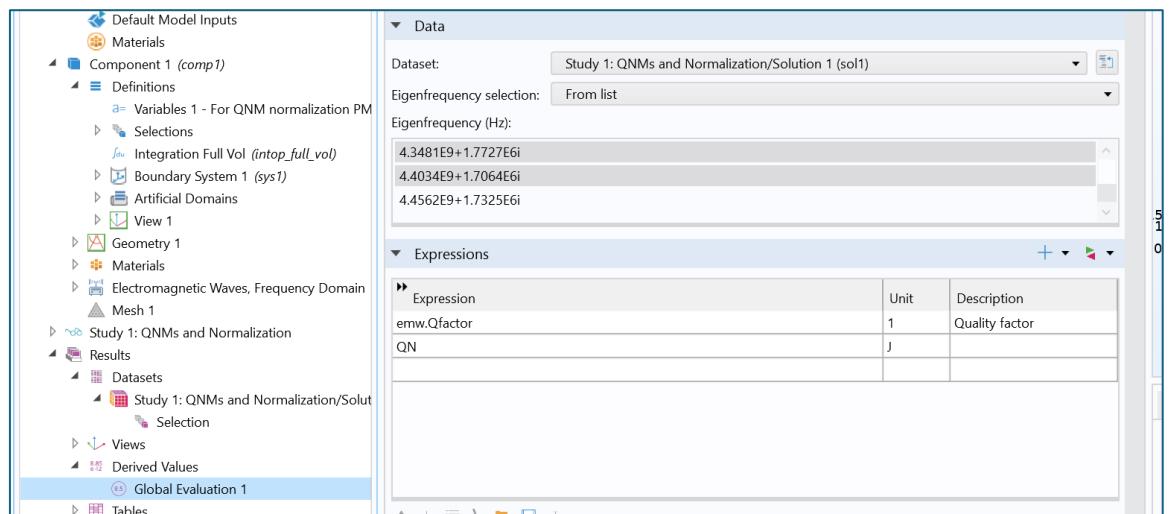
## (II) Data export

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:

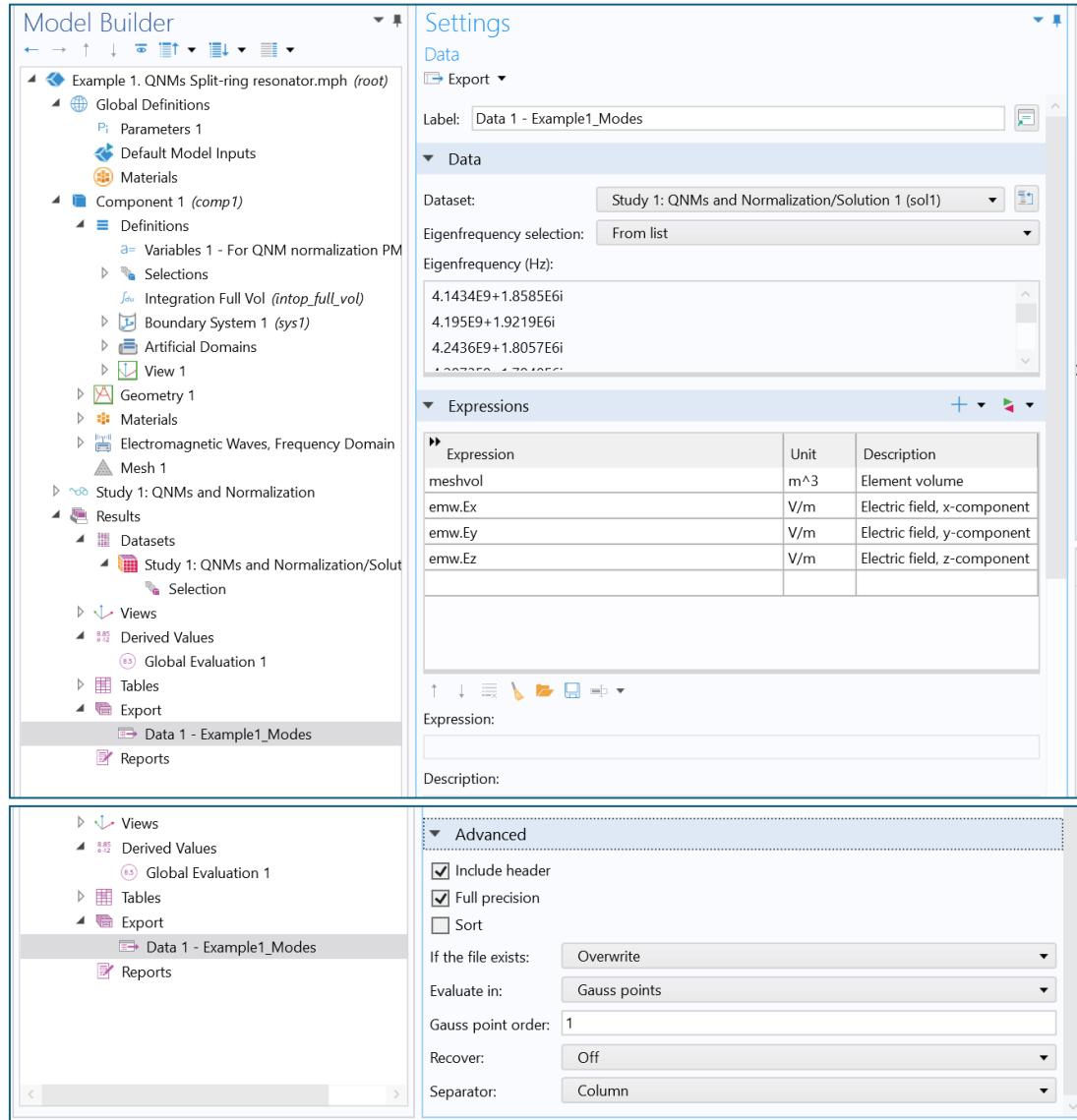
1. In “Datasets” choose data and make a selection, for the selection choose only resonator domains;



2. Perform “Global evaluation” to calculate Q factors and PML-norms (*emw.Qfactor*, *QN*), 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 points. Export this data to txt-file “*Example1\_Modes.txt*”.



4. Note that all data have to be in the SI units [meters, Hz].

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 organized in the 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 with results. For the example №1, corresponding Python script is “*Example 1. FanoScript - SplitRingResonator.ipynb*” (Jupiter Notebook environment was used).

The script is a simple reproduction of the theoretical derivation 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 and in absence of substrate. If one wants to perform calculations for a particle with another geometry, 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  $E_b$  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.5 in [1].

---

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

The calculation for this case is a little more complex, it is necessary to use the Auxiliary field formulation of the QNM theory [4]. The COMSOL model need to be modified in an appropriate way. The model “*Example 3. QNM Plasmonic silver dolmen.mph*” for that case was taken from the “MAN” repository [2] (<https://zenodo.org/records/7400937>). Additionally, the 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, “Electromagnetic Quasinormal Modes: a foundational framework for describing Fano resonances of extinction spectra”, *Nanophotonics*, 2026.
- [2] 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).
- [3] 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). <https://zenodo.org/records/7400937>
- [4] W. Yan, R. Faggiani, and P. Lalanne, “Rigorous modal analysis of plasmonic nanoresonators,” *Phys. Rev. B* **97**, 205422 (2018).