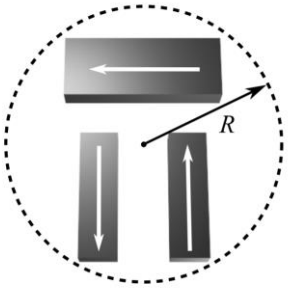
MAN
Modal Analysis
of Nanoresonators

QNMtoolbox_multipole: an openly available toolbox for computing the intrinsic multipole moments of QuasiNormal Modes

Tong Wu[1], Alexandre Baron[2], Philippe Lalanne[1], and Kevin Vynck[1]
[1]LP2N, Institut d'Optique d'Aquitaine, IOGS, Univ. Bordeaux, CNRS
[2]CRPP, Centre de Recherche Paul Pascal, Univ. Bordeaux, CNRS
wutong1121@sina.com
philippe.lalanne@institutoptique.fr
kevin.vynck@institutoptique.fr

Last revision: January, 2020



QNMtoolbox_multipole is copyright (c) 2019-2029, Institut d'Optique-CNRS.

QNMtoolbox_multipole is an openly available toolbox; you can redistribute it and/or modify it under the terms of the GNU General PublicLicense as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. It is composed of
- the present user guide document,
- 3 Matlab open-source scripts of the freeware package MAN (Modal Analysis of Nanoresonators) [1], built for extracting the normalized resonance modes (also called the quasinormal modes or QNMs) and computing their intrinsic multipole moments [2]. The script preferentially operates on the Matlab-COMSOL Livelink environment with the solver **QNMEig** [3]; however, the code would also work with QNMs computed with other software.
- **QNM_Dolmen_multipole.mph(.m) and QNM_Dolmen_multipole_sym.mph(.m)**, two COMSOL models provided under .mph or .m extensions for operation with the **QNMEig** solver. They are identical except that one is showing how to use planar symmetries.
- a repertory QNMtoolbox_multipole_functions composed of Matlab functions, to be included in the Matlab path. The functions should not be changed.

The toolbox 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 http://www.gnu.org/license/.

Table of contents

1. INTRODUCTION TO QNMTOOLBOX_MULTIPOLE

QNMtoolbox_multipole extracts the normalized resonance modes of plasmonic and photonic resonators computed by the **QNMEig** freeware [2] and then computes the intrinsic multipole moments. It serves as companion software to **QNMEig**.

1.1 Download & Installation

QNMtoolbox_multipole is a freeware that operates under the Matlab environment coupled to a COMSOL environment via Matlab-COMSOL Livelink in conjunction with COMSOL models featured or based on the QNMEig package. To install it, add the folder **QNMtoolbox_multipole** and its subfolders in your Matlab path.

1.2 How to acknowledge and cite

We kindly ask that you reference the **MAN** package from IOGS-CNRS and its authors in any publication/report for which you used it. The preferred citation for **QNMtoolbox_ multipole** is the following paper:

Tong Wu, Alexandre Baron, Philippe Lalanne, and Kevin Vynck, "Intrinsic multipolar contents of nanoresonators for tailored scattering", Phys. Rev. A(R)

1.3 Units and conventions of input/output data for QNMtoolbox_multipole

Unit. All the input information is required to be in the **SI unit**. Accordingly, the output information is given in the SI unit as well.

Convention. The time-dependent terms $\exp(-\mathrm{i}\omega t)$ are used for all the output results (e.g. the electric dipole moment, the magnetic dipole moment, …), which is in contrast to COMSOL.

1.4 Outline of the theory and related key issues

Similar to the real frequency case [4], the multipolar content of a QNM $\tilde{\mathbf{E}}$ can be determined quantitatively by expanding its field outside a sphere circumscribing the scatterer in vector spherical wave functions (VSWFs), as

$$\tilde{\mathbf{E}} = \tilde{k}^2 \sum_{n=1}^{\infty} \sum_{m=-n}^{n} E_{nm} \left[\tilde{a}_{nm} \tilde{\mathbf{N}}_{nm}^{(3)}(\mathbf{r}) + \tilde{b}_{nm} \tilde{\mathbf{M}}_{nm}^{(3)}(\mathbf{r})\right], \tag{1}$$

where $\tilde{a}_{nm}$ and $\tilde{b}_{nm}$ are electric and magnetic multipole coefficients, $\tilde{\mathbf{N}}_{nm}^{(3)}(\mathbf{r})$ and $\tilde{\mathbf{M}}_{nm}^{(3)}(\mathbf{r})$ are the outgoing VSWFs and $\tilde{k} = \tilde{\omega} n_b / c$ the complex wavevector of QNM. $\tilde{\omega}$ is the complex eigenfrequency and n_b is the refractive index of the background medium. The specific definitions of all the functions and parameters in Eq. (1) can be found in the supplement material of [2]. The coefficients $\tilde{a}_{nm}$ and $\tilde{b}_{nm}$ are obtained by computing the inner product of the QNM field with the VSWFs on the circumscribing sphere surface (see the dashed circle in Fig. 1):

$$\tilde{a}_{nm} = \frac{\int \tilde{\mathbf{E}} \cdot \tilde{\mathbf{N}}_{nm}^{(3)}(R,\Omega) d\Omega}{\tilde{k}^2 E_{nm} \int \left|\tilde{\mathbf{N}}_{nm}^{(3)}(R,\Omega)\right|^2 d\Omega} \tag{2}$$

$$\tilde{b}_{nm} = \frac{\int \tilde{\mathbf{E}} \cdot \tilde{\mathbf{M}}_{nm}^{(3)}(R,\Omega) d\Omega}{\tilde{k}^2 E_{nm} \int \left|\tilde{\mathbf{M}}_{nm}^{(3)}(R,\Omega)\right|^2 d\Omega} \tag{3}$$

and the Cartesian multipole moments of the QNM, electric dipole $\tilde{\mathbf{p}}$, magnetic dipole $\tilde{\mathbf{m}}$, and electric quadrupole $\tilde{\mathbf{Q}}^e$, can then be retrieved by matching their far-field expressions with those of the VSWFs in spherical coordinates.

Alternatively, there is an approach to calculate the multipole moments which is based on the internal field. In brief, in our program, we adopt the formulas within long-wavelength approximation [5]. The exact expressions without any approximation can be found in [6]. The lowest order multipole moments, i.e., the electric dipole $\tilde{\mathbf{p}}$, magnetic dipole $\tilde{\mathbf{m}}$ and electric quadrupole $\tilde{\mathbf{Q}}^e$ are given by

$$\tilde{\mathbf{p}} \approx \int_V \varepsilon_0 [\varepsilon(\tilde{\omega}) - \varepsilon_b] \tilde{\mathbf{E}}(\boldsymbol{r})\, d\mathbf{r}, \tag{4}$$

$$\tilde{\mathbf{m}} \approx -\frac{i\tilde{\omega}}{2} \int_V \varepsilon_0 [\varepsilon(\tilde{\omega}) - \varepsilon_b]\, \boldsymbol{r} \times \tilde{\mathbf{E}}(\boldsymbol{r}) d\mathbf{r}, \tag{5}$$

and

$$\tilde{\mathbf{Q}}^e \approx 3 \int_V \varepsilon_0 [\varepsilon(\tilde{\omega}) - \varepsilon_b] \left[\boldsymbol{r}\tilde{\mathbf{E}}(\boldsymbol{r}) + \tilde{\mathbf{E}}(\boldsymbol{r})\boldsymbol{r} - \frac{2}{3}\boldsymbol{r} \cdot \tilde{\mathbf{E}}(\boldsymbol{r})\mathbf{I}\right] d\mathbf{r}, \tag{6}$$

where $\mathbf{ab}$ is the tensor product between $\mathbf{a}$ and $\mathbf{b}$, ε_b the permittivity of the background medium and $\mathbf{I}$ is the unit dyadic tensor. The integrations are performed over the resonator volume V.

To summarize, the QNM computation and normalization (the prerequisite to using **QNMtoolbox_multipole**) are performed with **QNMEig**, using COMSOL software. The **QNMtoolbox_multipole** then extracts the normalized QNM fields, performs the overlap integrals defined in Eqs. (2-3) to compute $\tilde{a}_{nm}$ and $\tilde{b}_{nm}$. $\tilde{\mathbf{p}}$, $\tilde{\mathbf{m}}$, and $\tilde{\mathbf{Q}}^e$ can then be retrieved. Alternatively, the toolbox also contains a program that allows for calculating the Cartesian multipole moments based on the internal field formulas Eqs. (4-6).

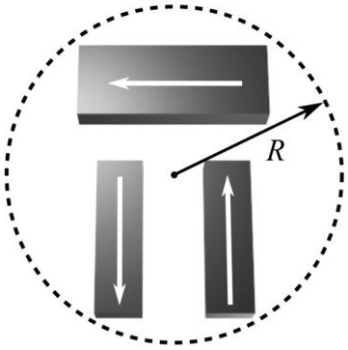


Fig. 1. Illustration diagram of the investigated resonator. The multipole coefficients $\tilde{a}_{nm}$ and $\tilde{b}_{nm}$ are obtained by computing the inner product of the QNM field with the VSWFs on the circumscribing sphere surface, as denoted by the dashed circle.

1.5 Primary functions

Function name	Description	Subfunction[a]
main_field_multipole.m	Main Matlab script for extracting the QNMs and calculating the multipole components of the QNM. All the inputs are modified in the "user defined parameters" section of this program.	N
main_field_multipole_nosym.m	Same with 'main_field_multipole.m' but for an example where no symmetry is used.	N
modelPost_obj_multi2	Read the input parameters assigned by the user.	Y
extract_sol	Extract the material parameters, eigenfrequencies, modal normalizers from the COMSOL file.	Y
extract_point_power	Extract the modal fields inside the resonator from the COMSOL file.	Y
extract_point_sym	Obtain the uncalculated modal fields using symmetry.	Y
anm_bnm_eigen_QN	Use Eq. (2) and Eq. (3) to calculate the $\tilde{a}_{nm}$ and $\tilde{b}_{nm}$, and then further calculate the Cartesian multipole moments, $\tilde{\mathbf{p}}$, $\tilde{\mathbf{m}}$, and $\tilde{\mathbf{Q}}^e$.	Y
multipole_LZX	Compute the $\tilde{\mathbf{p}}$, $\tilde{\mathbf{m}}$, and $\tilde{\mathbf{Q}}^e$ using Eqs. (4)-(6).	Y
idmaterial.m	Identify the material type for each domain of the resonator.	N
Drude_epsln_multi.m	Compute the permittivity using the Drude model.	N
Nnm_fixed.m	Compute the $\mathbf{e}_\mathrm{r}$, $\mathbf{e}_\theta$, and $\mathbf{e}_\phi$ components of the VSWFs $\tilde{\mathbf{N}}_{nm}^{(3)}$.	N

Mnm_fixed.m	Compute the $\mathbf{e}_{\mathrm{r}}$, $\mathbf{e}_{\theta}$, and $\mathbf{e}_{\phi}$ components of the VSWFs $\widetilde{\mathbf{M}}_{nm}^{(3)}$.	N
dipole_test2.m	Compute the $\tilde{\mathbf{p}}$ and $\tilde{\mathbf{m}}$ using $\tilde{a}_{nm}$ and $\tilde{b}_{nm}$.	N
quadrupole_test.m	Compute the $\tilde{\mathbf{Q}}^{e}$ using $\tilde{a}_{nm}$.	N
…[b,c]		

[a] When the subfunction term is 'Y' meaning that the function is a subfunction of the module '**modelPost_obj_multi2.m**'

[b]There exist some functions in the Module '**modelPost_obj_multi2.m**' which are unlisted for concision. These functions are used for data and file management.

[c] Some programs in the folder 'functions' used for calculating special mathematical funtions and sampling points on the integral surface are also unlisted. Among these programs, the 'retgauss_precision.m', 'retgauss.m', 'retelimine.m', 'retcolonne.m' are extracted from **RETICOLO** [1], another freeware available at the group website.

1.6 Input parameters assigned by the user

Input	Description	COMSOL[a]
model_path	Path of the COMSOL file.	N
model_name	Name of the COMSOL file.	N
PEC	Perfect electric conductor planes. e.g. PEC=[0 1 0] denotes there is a PEC symmetry plane in the Y-direction. If no PEC symmetry plane is used in the COMSOL file, set it as [0 0 0].	Y
PMC	Perfect magnetic conductor planes. e.g. PEM=[1 0 0] denotes there is a PMC symmetry plane in the X-direction. If no PMC symmetry plane is used in the COMSOL file, set it as [0 0 0].	Y
scatter	Domains of the resonator.	Y
matvar	Property name of the Variable in the COMSOL file where material parameters of the resonator are provided.	Y
nmax	Truncated order n for the VSWFs.	N
r	The radius of the circumscribing sphere, the R of Fig. 1.	N
Ndes_th1	Variable to control the number of points generated on the circumscribing sphere for integration. Normally, set all of them as 10 is sufficient to calculate Eqs. (2-3) accurately.	N
Ndes_th2		
Ndes_ph1		
Ndes_ph2		

[a] When the COMSOL is 'Y' meaning the input parameter should be given based on the information in the COMSOL file.

1.7 Output

Output	Description	Structure
Sca.in	Input information checklists.	Struct
Sca.material		
Sca.inpart		
Sca.in_sym		
Sca.Sol	Information about the solutions.	Struct including solnum, tb_freq, QN.
Sca.Sol.solnum	Number of QNMs.	Integer

Sca.Sol.tb_freq	Eigenfrequencies for the QNMs.	[Sca.Sol.solnum×1] array
Sca.VSHe	Multipole moments and coefficients with the $\exp(-i\omega t)$ convention computed with Eqs. (2-3).	Struct including anm_qnm, bnm_qnm, pk_qnm, mk_qnm, EQ_qnm, with k standing for X, Y, and Z.
Sca.VSHe.anm_qnm	Multipole coefficients $\tilde{a}_{nm}$ and $\tilde{b}_{nm}$ computed with Eqs. (2-3).	[nmax×(2nmax+1)×Sca.Sol.solnum] array. The first index stands for n, the second for $m + \mathrm{nmax} + 1$, and the third for the number of the solution. E.g. anm_qnm(4, $3 + \mathrm{nmax} + 1$, 7) stands for the $\tilde{a}_{4,3}$ of the 7th QMM.
Sca.VSHe.bnm_qnm		
Sca.VSHe.px_qnm	X, Y, and Z components of the electric dipole moments computed with $\tilde{a}_{nm}$ and $\tilde{b}_{nm}$.	[1× Sca.Sol.solnum] array E.g. px_qnm(2) stands for the $\tilde{p}_x$ of the 2nd QMM.
Sca.VSHe.py_qnm		
Sca.VSHe.pz_qnm		
Sca.VSHe.mx_qnm	X, Y, and Z components of the magnetic dipole moments computed with $\tilde{a}_{nm}$ and $\tilde{b}_{nm}$.	[1× Sca.Sol.solnum] array E.g. mz_qnm(3) stands for the $\tilde{p}_x$ of the 3rd QMM.
Sca.VSHe.my_qnm		
Sca.VSHe.mz_qnm		
Sca.VSHe. EQ_qnm	Electric quadrupole moments computed with $\tilde{a}_{nm}$ and $\tilde{b}_{nm}$.	[3×3×Sca.Sol.solnum] array E.g. EQ_qnm(1,2,4) stands for the $\tilde{Q}^e_{xy}$ of the 4th QMM.
Sca.CTSe	Multipole moments in Cartesian coordinate with the $\exp(-i\omega t)$ convention computed with Eqs. (4-6) and those in [6].	Struct including pxyz_qnm_Lzx, mxyz_qnm_Lzx, txyz_qnm_Lzx, EQ_qnm, MQ_qnm, and EO_qnm.
Sca.CTSe. pxyz_qnm_Lzx	X, Y, and Z components of the electric dipole moments computed with Eq. (4).	[3× Sca.Sol.solnum] array E.g. pxyz_qnm_Lzx (2, 9) stands for the $\tilde{p}_y$ of the 9th QMM.
Sca.CTSe. mxyz_qnm_Lzx	X, Y, and Z components of the magnetic dipole moments computed with Eq. (5).	[3× Sca.Sol.solnum] array E.g. mxyz_qnm_Lzx (1, 9) stands for the $\tilde{p}_x$ of the 9th QMM.
Sca.CTSe. EQ_qnm	Electric quadrupole moments computed with Eq. (6).	[3×3×Sca.Sol.solnum] array E.g. EQ_qnm(1,2,4) stands for the $\tilde{Q}^e_{xy}$ of the 4th QMM

2. TUTORIAL: THE MULTIPOLE MOMENTS OF THE QNMS OF AN AG DOLMEN

In this part, we provide details on how to compute the multipole moments of a QNM with **QNMtoolbox_multipole**. A simple example, silver dolmen in air, is provided. The COMSOL model file is "**dolmen_PRA2020_multipole_sym.mph**".

Figure 1 sketches the plasmonic dolmen, made of silver and composed of an upper rod 128×50×20 nm^3 separated by a gap of 30-nm-wide gap from two lower rods 30×100×20 nm^3 separated by 30 nm.

The Ag dolmen has the permittivity

$$\varepsilon_{\mathrm{Ag}} = \varepsilon_{\infty,\mathrm{Ag}}\left(1 - \frac{\omega_{\mathrm{p,Ag}}^2}{\omega^2 - \omega_{0,\mathrm{Ag}}^2 + i\omega\gamma_{\mathrm{Ag}}}\right), \tag{7}$$

with $\varepsilon_{\infty,\mathrm{Ag}} = 1$, $\omega_{\mathrm{p,Ag}} = 1.3659\mathrm{e}16\ \mathrm{rad/s}$ corresponding to $\lambda_{\mathrm{p,Ag}} = 138\ \mathrm{nm}$ in vacuum, $\gamma_{\mathrm{Ag}} = 0.0023\omega_{\mathrm{p,Ag}}$, and $\omega_{0,\mathrm{Ag}} = 0$. The dolmen is embedded in air with $\varepsilon_{\mathrm{b}} = 1$.

2.1 Modeling Instructions

The steps for building the model are basically same as those of the "**QNMEig_Sphere.mph**" provided in the repertory of the **QNMEig**. Thus, here we only focus on the differences between the two. We recommend that the user starts from the **QNMEig_Sphere.mph** and follows the instructions below to build the model.

1/ In the **Global Definitions** part, substitute the geometric parameters for the sphere with those of the Dolmen.

- On the **Model builder** toolbar, click **Parameters 1**.
- In the **Settings** window for the **Parameters 1**, locate the **Parameters** section. In the table, enter the following settings:

Name	Expression	Description
Symmetry parameter for the structure		
sym_factor	4	symmetry factor=2^(Number of symmetry planes)
Geometrical parameters		
width_doublet	30[nm]	width of doublet
length_doublet	100[nm]	length of doublet
width_gros	128[nm]	width of top rod
length_gros	50[nm]	length of top rod
epaisseur	20[nm]	thickness
gap_gros	30[nm]	the separation between the top rod and the doublet
gap_doublet	30[nm]	gap of the doublet

2/ Rebuild the **Geometry 1** for the studied system.

The geometry consists of a dolmen in air background, surrounded by a PML. Thanks to mirror symmetry, the whole structure is reduced to one quarter for simulations. Specific mirror symmetries with respect to electromagnetic fields are imposed either by **Perfect Electric Conductor** or **Perfect Magnetic Conductor** boundary conditions.

3/ Because we use the symmetry condition in this model, the normalizer for the QNM should be redefined.

- On the **Definitions** toolbar, click **QNM normalization**.
- In the **Settings** window for the **QNM normalization**, locate the **Variables** section. In the table, replace the **Expression** for QN as:

Name	Expression
QN	2*sym_factor*intAll((emw.Ex*emw.Dx+emw.Ey*emw.Dy+emw.Ez*emw.Dz)*pml1.detInvT)+sym_factor*intMetal((emw.Ex*emw.Dx+emw.Ey*emw.Dy+emw.Ez*emw.Dz)*pml1.detInvT)*fdisp

4/ To let the **QNMtoolbox_multipole** automatically identify the material parameters, we defined the following **Variables**.

- On the **Definitions** toolbar, click **Variables**.
- In the Label window for **Variables**, enter 'Lorentz-Drude parameters for silver dolmen'.
- In the **Settings** window for Variables, locate the **Variables** section. In the table, enter the following settings:

Name	Expression
omegap1	omegap_Ag
gamma1	gamma_Ag
omega01	omega0_Ag
epsiloninf	epsiloninf_Ag

5/ Build a proper mesh for the studied system.

6/ Chose a proper frequency for the eigensolver to search around and launch the computation

7/ Save the model.

2.2 Computation the multipole moments

1/ Open the Matlab script **main_field_multipole.m**, and make sure that the Matlab files in the folder **QNMtoolbox_multipole** and its subfolder are in your Matlab working path.

2/ In **main_field_multipole.m**, check and modify the input parameters listed in section 1.6 if necessary.

3/ Run the program.

4/ Check the values listed in section 1.7.

2.3 Retrieving and interpreting the results

1/ Checking all the inputs
Typing 'Sca.in' in the command window of the Matlab, then all the input variables are listed.
In our example, 'Sca. in' is:
struct with fields:
model_path: 'X:\XXX \COMSOL model licensed'
model_name: ' dolmen_PRA2020_multipole_sym .mph'
r: 1.2000e-07 % Radius of the circumscribing sphere. The value has to be smaller than the radius of the air domain.
nmax: 2 % Maximum number of the n.
Ndes_th1: 10
Ndes_th2: 10
Ndes_ph1: 10
Ndes_ph2: 10 % Variable to control the number of points generated on the circumscribing sphere
epsln_b: 1 % Maximum number of the n.
namb: 1 % Refractive index of the background medium.
Nmaterial: 1 % Total number of the materials.
matvar: {'var12'} %Tag of the Variables in the COMSOL model where 'omegap1', 'gamma1', …, are defined.
scatter: [4 5] % Domains of the scatter.
Nsca: 2 % Number of domains of the scatter.
scamat: [1 1] % The material type for the domains of the scatter. In this case, the value [1 1] denotes that the material types for the domains 4 and 5 of the scatter are both 1. Information about the k-th material is recorded in 'Sca.material(k)'. If the material type is 0, it means the domain has a constant permittivity and the value is recorded in 'Sca.in.scaeps'.
scaeps: [1 1]
sub: '' % Irrelevant with this toolbox.

Nsub: 0 % Irrelevant with this toolbox.
submat: [1×0 double] % Irrelevant with this toolbox.
subeps: [1×0 double] % Irrelevant with this toolbox.

2/ Checking the symmetry
Typing 'Sca.in_sym' in the command window of Matlab, then the symmetry planes used in the COMSOL model are shown.
In our example, 'Sca. in_sym' is given as:
struct with fields:
PEC: {[1 0 0]}
PMC: {[0 0 1]} % In the present case, we have one PEC plane in the X-direction and one PMC plane in the Z-direction.

3/ Obtaining the values of the multipole moments
The values of the multipole moments and coefficients can be obtained by typing the variable names listed in section 1.7. Here, we list parts of the output values of our example:

Physical parameters	Value for QNM1	Value for QNM2	Command
$\tilde{p}_x$(Calculated with the surface integral)	-0.6465e-16 - 0.0953ie-16	-0.51050e-16 +0.1047e-16i	Sca.VSHe.px_qnm
$\tilde{p}_y$(Calculated with the surface integral)	0.0847e-31 0.2310ie-31	0.1966e-31 + 0.1081e-31i	Sca.VSHe.py_qnm
$\tilde{m}_z$ (Calculated with the surface integral)	-2.6782e-11 -8.9046e-09i	-1.8870e-09 -2.0839e-09i	Sca.VSHe.mz_qnm
$\tilde{Q}^e_{xy}$(Calculated with the surface integral)	-7.3236e-24 -2.7222e-24i	-1.6873e-23 +1.4715e-24i	Sca.VSHe.EQ_qnm(1,2,:)
$\tilde{a}_{11}$	0.9189 -7.5547i	-1.4873 -6.5222i	Sca.VSHe.anm_qnm(1,nmax+1+1,:)
$\tilde{p}_x$(Calculated with Eq. (4))	-0. 6939e-16 -0.1023e-16	-0.5751e-16 0.1166e-16i	Sca.CTSe.pxyz_qnm_Lzx(1,:)
$\tilde{m}_z$(Calculated with Eq. (5))	-1.900e-11 -9.190e-09i	-1.988e-09 -2.294e-09i	Sca.CTSe.mxyz_qnm_Lzx(3,:)
$\tilde{Q}^e_{xy}$(Calculated with Eq. (6))	-7.9032e-24 -2.7953e-24i	-1.7672e-23 +1.6431e-24i	Sca.CTSe.EQ_qnm(1,2,:)

REFERENCES

[1] https://www.lp2n.institutoptique.fr/light-complex-nanostructures
[2] T. Wu, A. Baron, P. Lalanne, and K. Vynck, "Intrinsic multipolar contents of nanoresonators for tailored scattering", Phys. Rev. A(R) ??????????
"Intrinsic multipolar contents of nanoresonators for tailored scattering"
[3] W. Yan, R. Faggiani, P. Lalanne, Phys. Rev. B **97**, 205422 (2018).
"Rigorous modal analysis of plasmonic nanoresonators"
[4] S. Mühlig, C. Menzel, C. Rockstuhl, and F. Lederer, Metamaterials **5**, 64 (2011).
"Multipole analysis of meta-atoms"
[5] P. D. Terekhov, K. V. Baryshnikova, Y. A. Artemyev, A. Karabchevsky, A. S. Shalin, and A. B. Evlyukhin, Phys. Rev. B **96**, 035443 (2017).
"Multipolar response of nonspherical silicon nanoparticles in the visible and near-infrared spectral ranges"
[6] R. Alaee, C. Rockstuhl, and I. Fernandez-Corbaton, Opt. Commun. **407**, 17 (2018).
"An electromagnetic multipole expansion beyond the long-wavelength approximation"