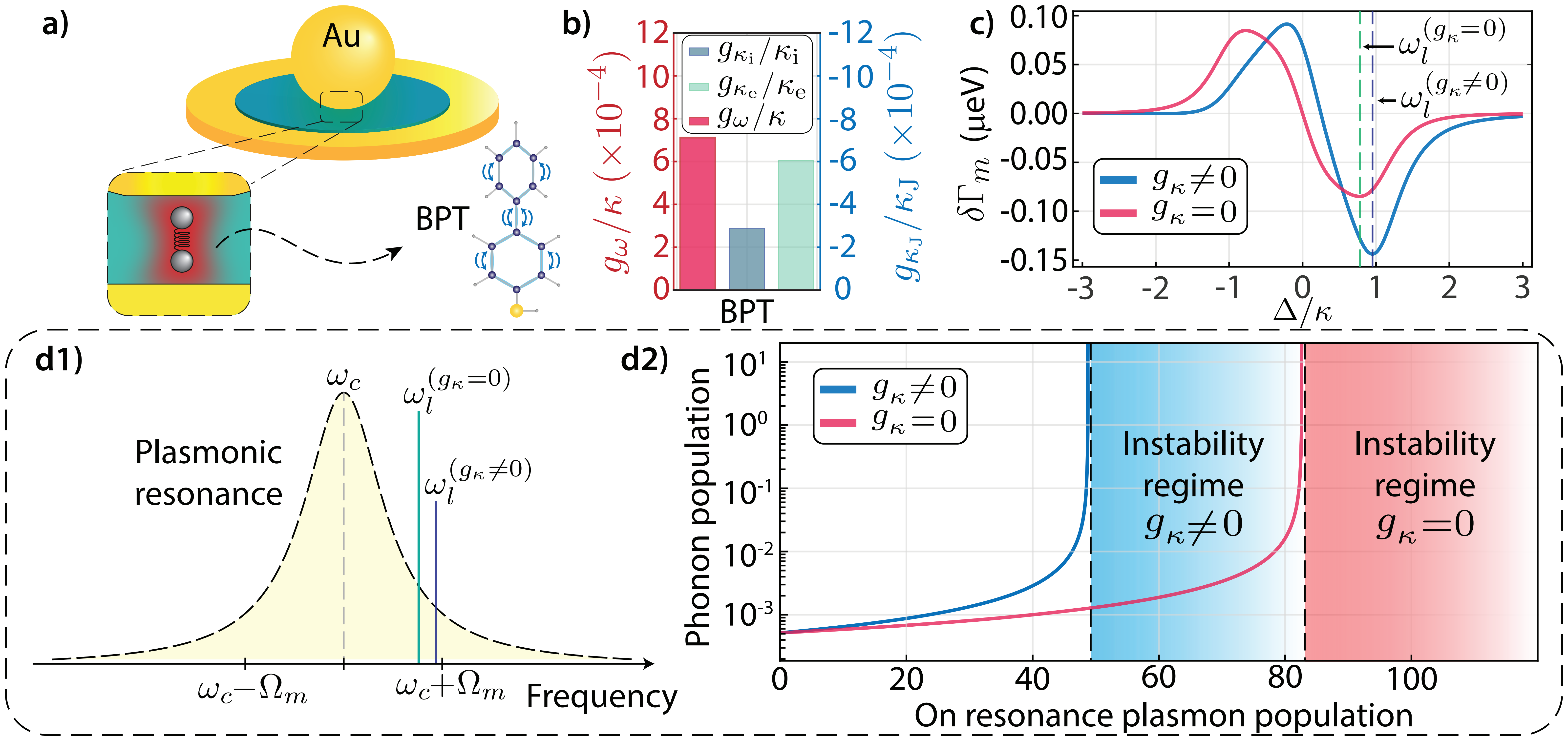
Quasinormal-mode perturbation theory for dissipative and dispersive optomechanics



André G. Primo*, Natália C. Carvalho, Cauê M. Kersul, Newton C. Frateschi, Gustavo S. Wiederhecker, Thiago P. M. Alegre**

*agprimo@ifi.unicamp.br
**alegre@unicamp.br

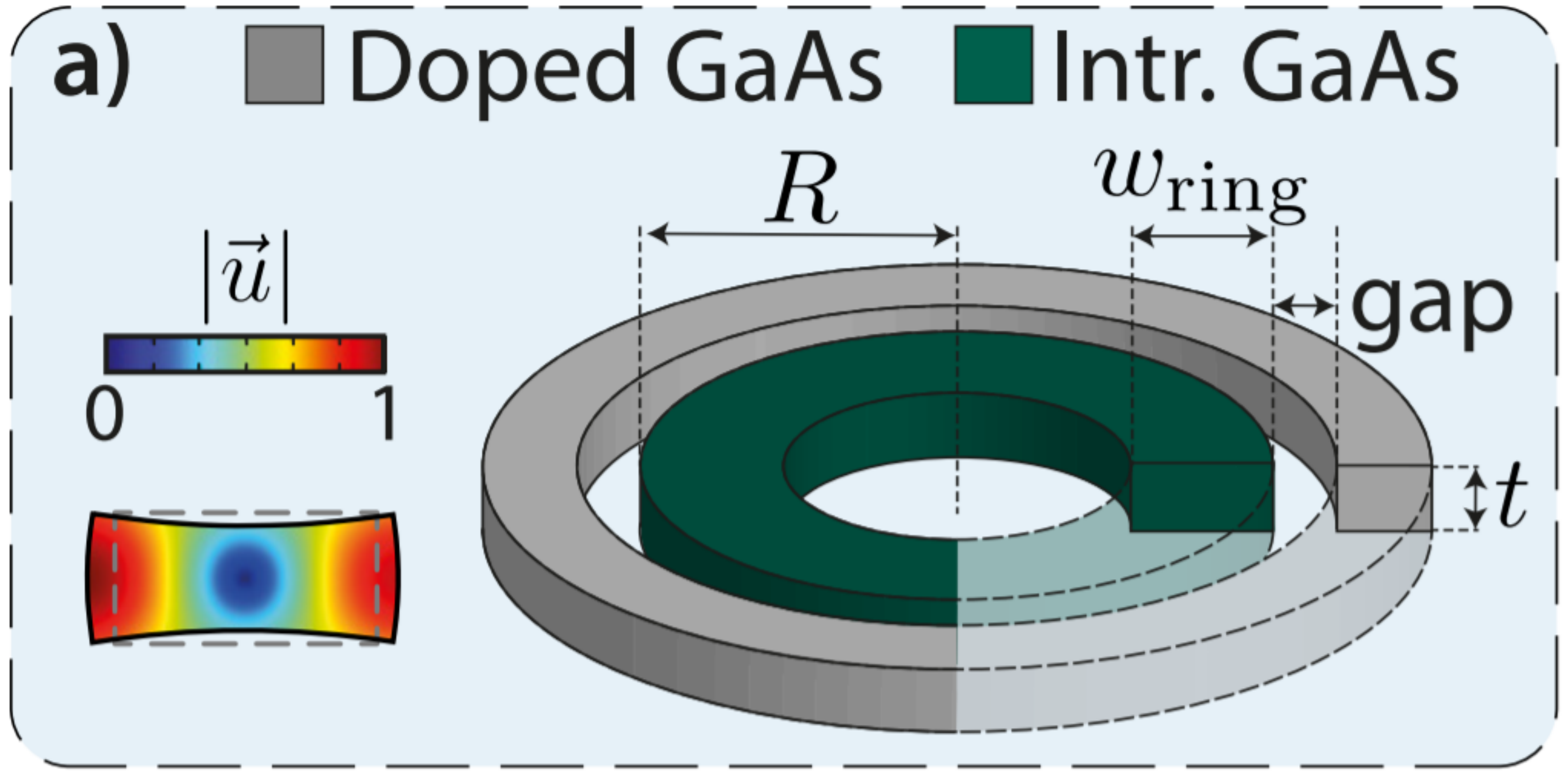
Available files

- Ring resonator coupled to absorber medium (Comsol, Python)
- Split-beam nanocavity (Comsol, Python)
- Nanoparticle-on-a-mirror (Comsol, Mathematica, Python)
- Infinite nanocylinder (Comsol, Mathematica, Python)
- Nanoparticle-on-a-mirror – Dielectric nanosphere (Comsol, Python)

Comsol version 5.4 is required. The accompanying simulation files were not tested on other versions. Details in the structure of the simulations can be found in the repository for: “Brillouin Optomechanics in Nanophotonic Structures” - DOI: 10.1063/1.5088169.

Ring resonator coupled to absorber medium

File “comsol_simulation_file_fig_2.mph” calculates optical and mechanical modes of a GaAs ring resonator coupled to a heavily doped GaAs ring. Implementations of the normal-mode and quasi-normal mode perturbation theory are found in the simulations.

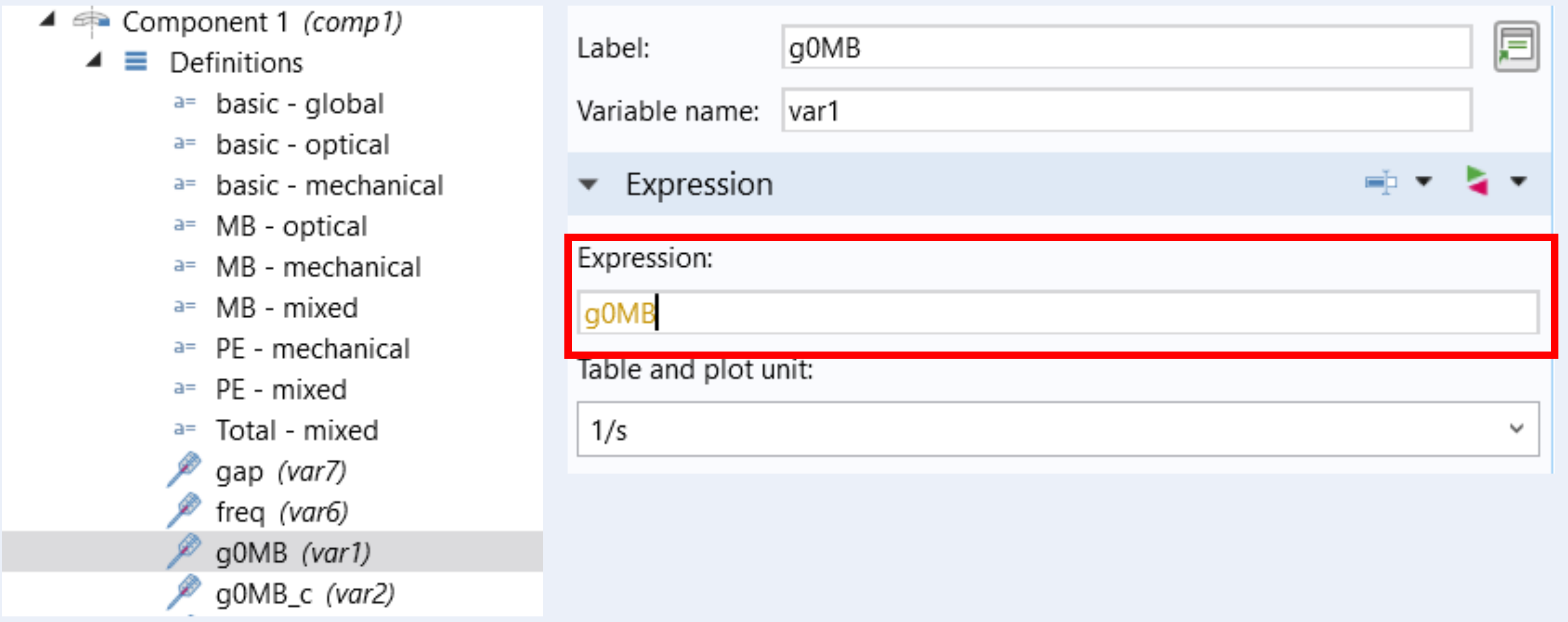


Python:

1. Open file “fig_2_plot_data.ipynb”. Load “Fig_2_MB_mod.txt” and “Fig_2_PE_mod.txt” to generate plots.

Generating data from Comsol:

1. Moving boundary or Photoelastic calculations are chosen by setting the variable *mmesh = 0* or *mmesh = 1* (*Global definitions -> Parameters 1*), respectively.
2. If *mmesh = 0*, probes g0MB (var1) and g0MB_c(var2) must be set to compute g0MB and g0MB_c, respectively.
3. If *mmesh = 1*, probes g0MB (var1) and g0MB_c(var2) must be set to compute g0PE and g0PE_c, respectively.



Label: g0MB
Variable name: var1
Expression: g0MB
Table and plot unit: 1/s

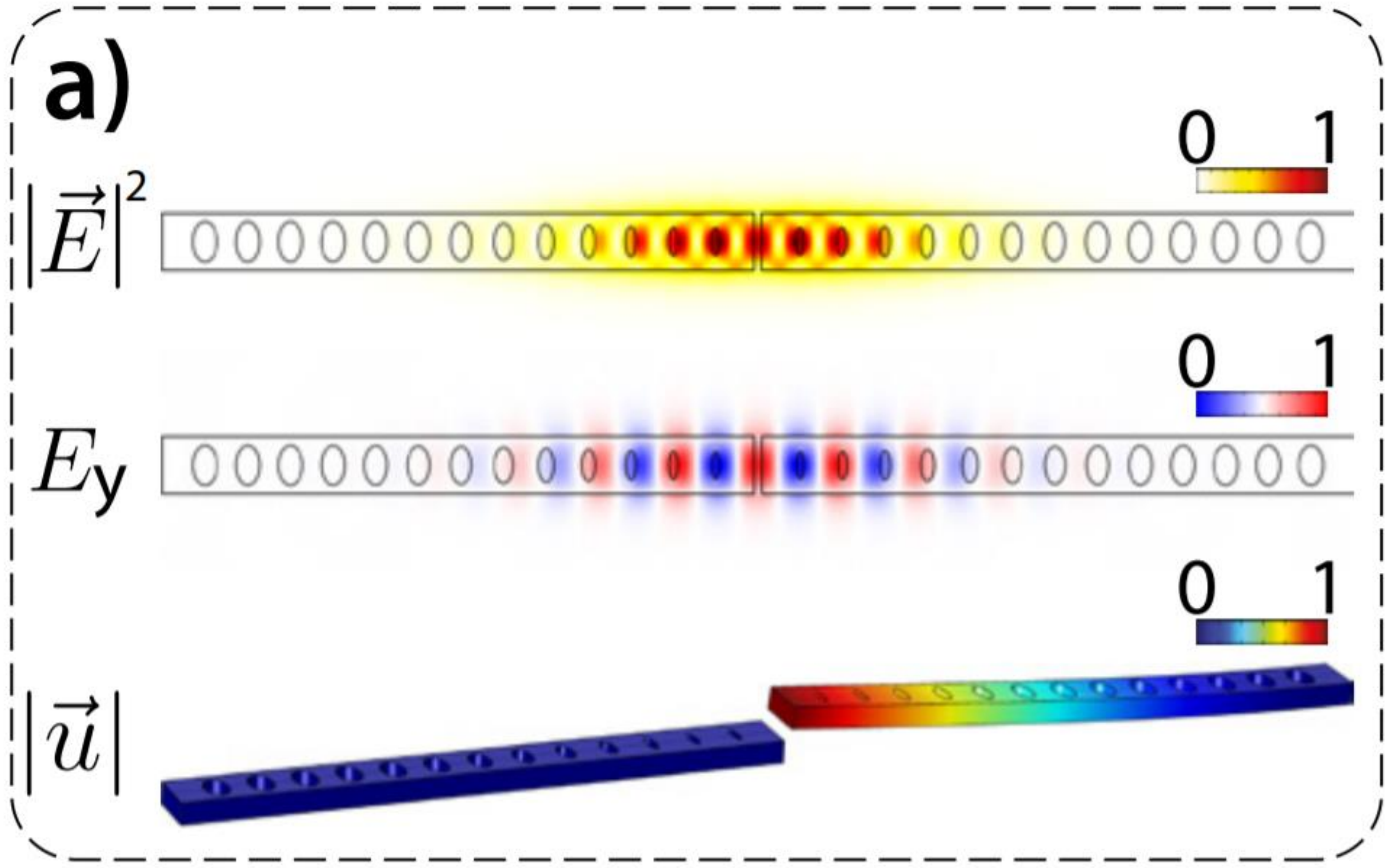
g0MB
g0PE
g0MB_c
g0PE_c

4. Run the simulation on the mechanics node. This one is set to solve for the mechanical breathing mode of the disk.
5. Run the simulation on the optics node. This one will sweep over the gap distance between the intrinsic and doped GaAs rings. For each gap, the optical modes for the deformed and undeformed geometries are computed. This follows from the usage of the Moving Mesh module. Data is generated automatically in a table.
6. Exported data can be found in the “fig_2_MB_mod.txt” and “fig_2_PE_mod.txt” files.

All variables with subscript “c” are defined to compute results using the normal-mode perturbation theory.

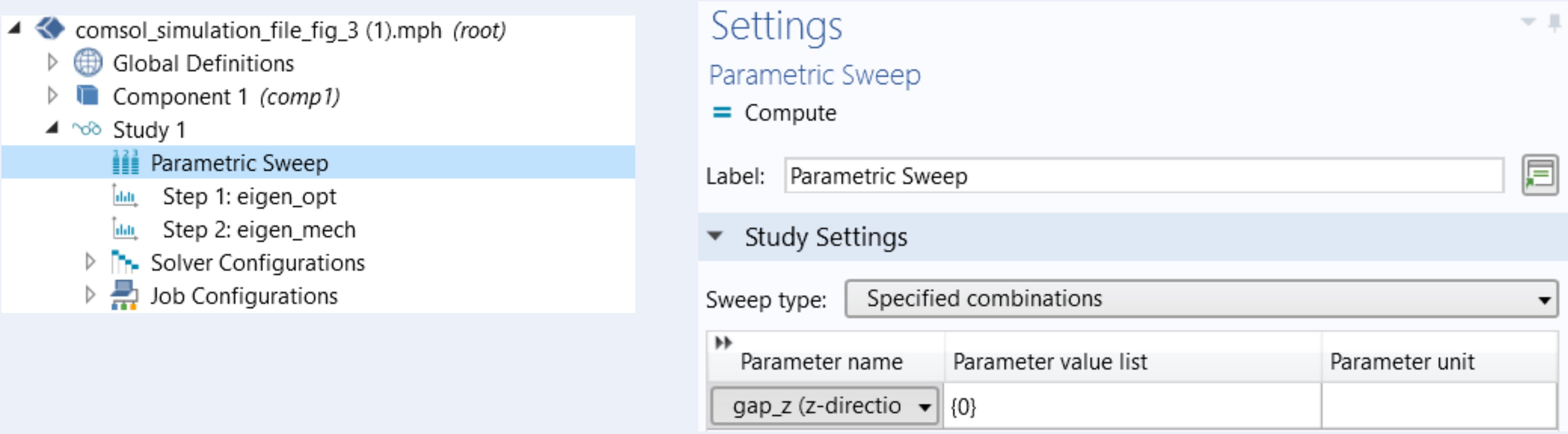
Split-beam nanocavity

File “comsol_simulation_file_fig_3.mph” calculates optical and mechanical modes of a split-beam nanocavity. Implementations of quasi-normal mode perturbation theory are found in the simulations.



Generating data from Comsol:

1. Set the desired z-gap in the *Parametric Sweep* node, under *Study 1:*



Label: Parametric Sweep

Sweep type: Specified combinations

Parameter name	Parameter value list	Parameter unit
gap_z (z-directio	{0}	

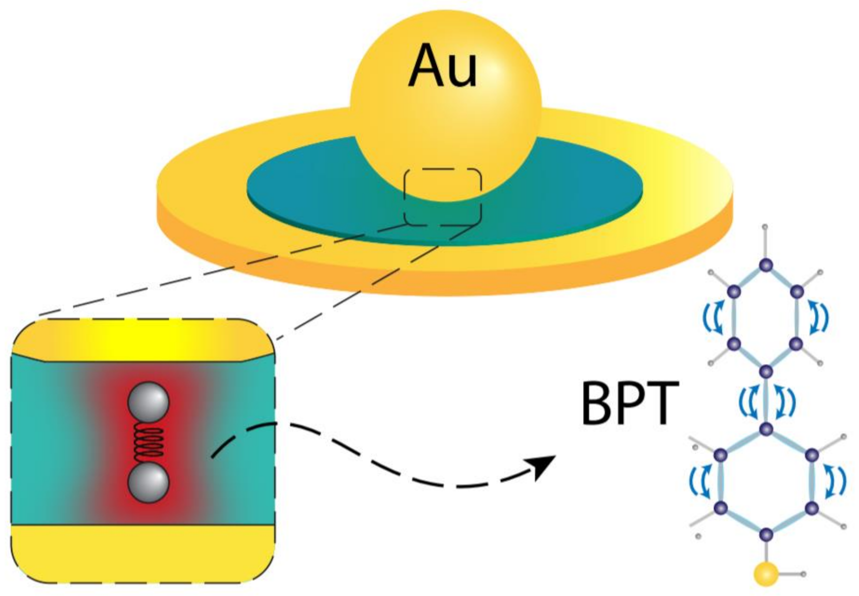
2. Run *Study 1.*
3. Data is automatically generated in a table.
4. Exported data can be found in “data_fig3.csv”.

Python:

1. Open file “fig_3_plot_data.ipynb”. Load “data_fig3.csv” to generate plots.

Nanoparticle-on-a-mirror

File “comsol_simulation_file_fig_4.mph” and “comsol_simulation_file_fig_4_no_PML.mph” calculate plasmonic modes of a gold nanoparticle placed on top of a gold mirror – in the presence and absence of PMLs, respectively. Gold is modeled through a Drude-Lorentz-type permittivity. Implementations of quasi-normal mode perturbation theory are found in the simulations, assuming the vibrational modes in question are of molecules treated as point-dipoles.

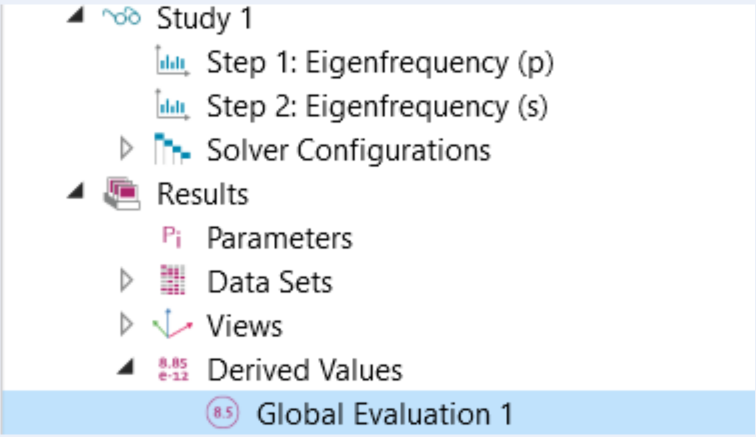


Python:

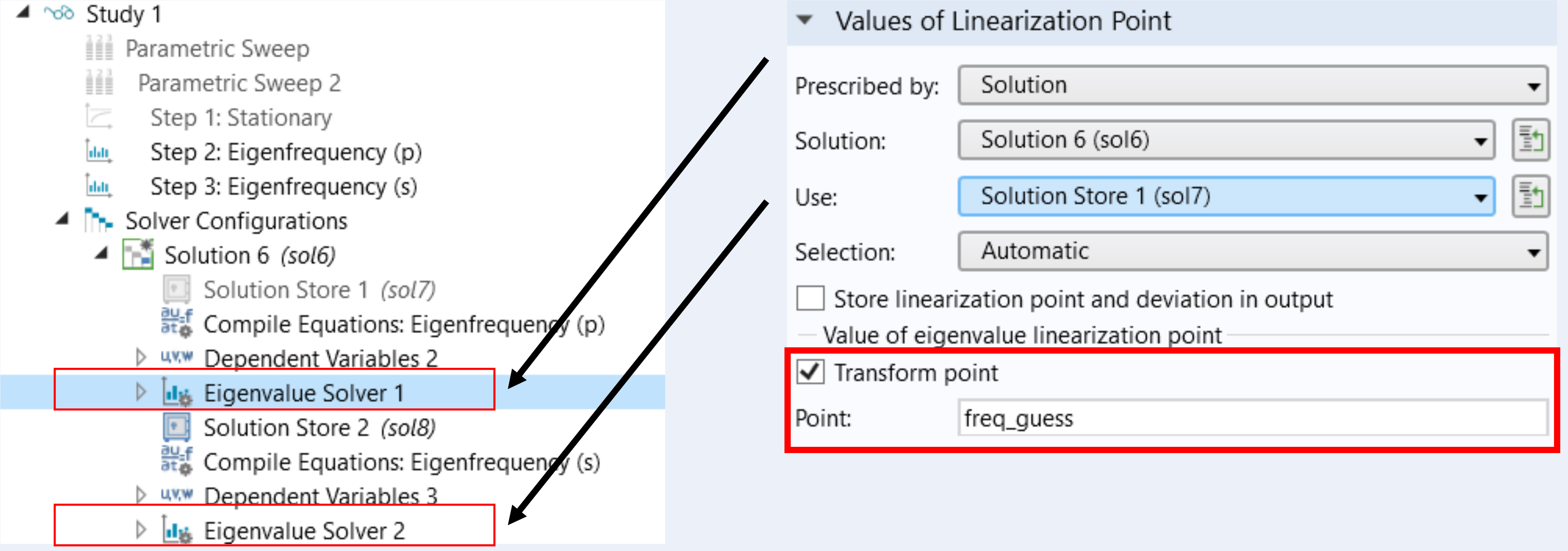
1. Figure S2 c) and d) are generated from the same Comsol file, although the radius of the gold nanosphere is different than the one used on the main text. The single difference being that a sweep over the gap between nanoparticle and PML is performed. This is found in file “comsol_simulation_file_S2_c_d.mph”.
2. Output data (“fig_S2_c_d_data.txt”) is loaded in “fig_S2_plot_data.ipynb”

Generating data from Comsol:

1. Run *Study 1.*
2. Under the *Results -> Derived values* node, compute *Global Evaluation 1*. The output is the value for the generalized optomechanical coupling and the Q-factor of the cavity. Those are used as input for the Mathematica file used to generate Figs. 4 c) and d2).



3. Be sure that *transform point* is on and set to *freq._guess. Transform point* can be found in *Solver configurations -> Eigenvalue Solver 1 (or 2).*



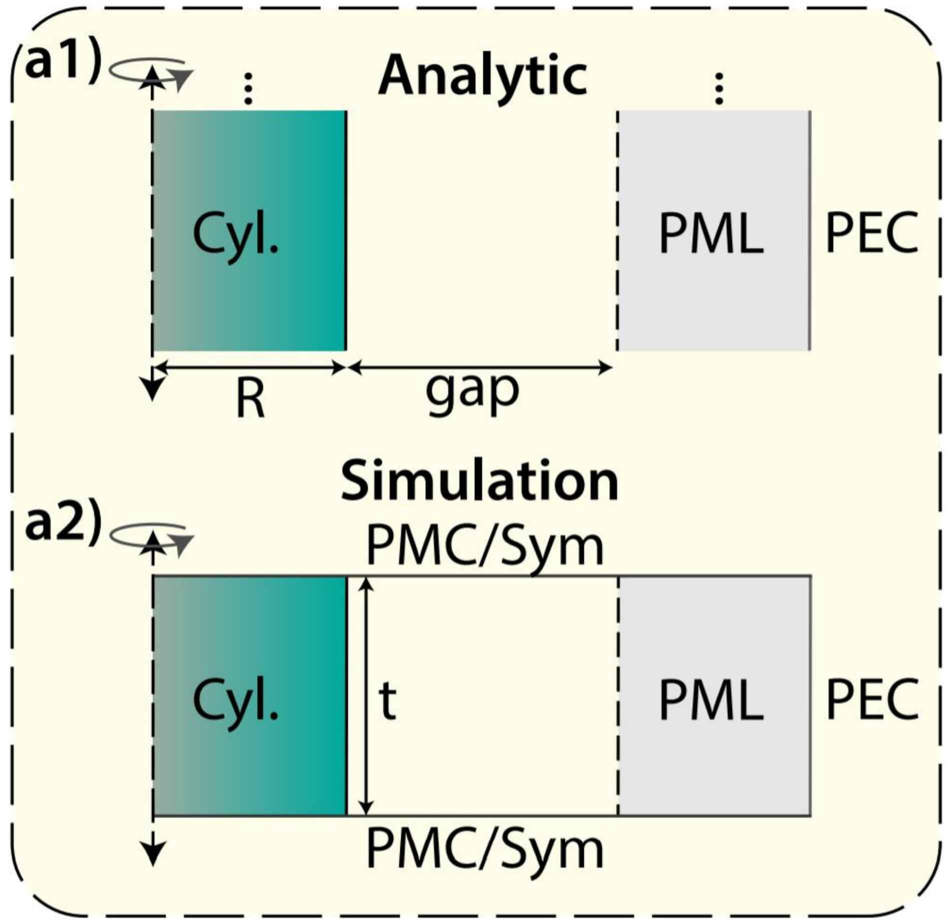
Prescribed by: Solution
Solution: Solution 6 (sol6)
Use: Solution Store 1 (sol7)
Selection: Automatic
☐ Store linearization point and deviation in output
Value of eigenvalue linearization point
☑ Transform point
Point: freq_guess

Mathematica:

1. Results from the Comsol simulation are inputs for variables *f0*, $g\kappa$ and $g\omega$, kabs, and $g\kappa$abs. The force spectrum and damping rate modifications are calculated from these results.

Infinite nanocylinder

This model supports both analytic and numerical solutions. The analytic solution is found in “fig_S2_a_b_mathematica.nb” or “fig_S1_b_mathematica.nb”, whereas the numerical solution is found in “comsol_simulation_file_S2_a_b.mph”. Both compute optical and mechanical modes of the structure. In this exemple, only moving boundary contributions are considered in the optomechanical coupling.



Generating data from Comsol:

1. Run the *Optomechanics* node. This is set to calculate optical and mechanical modes of the structure.
2. A sweep over the gap distances between cylinder and PML will be performed. The output is automatically generated in a table.

Mathematica:

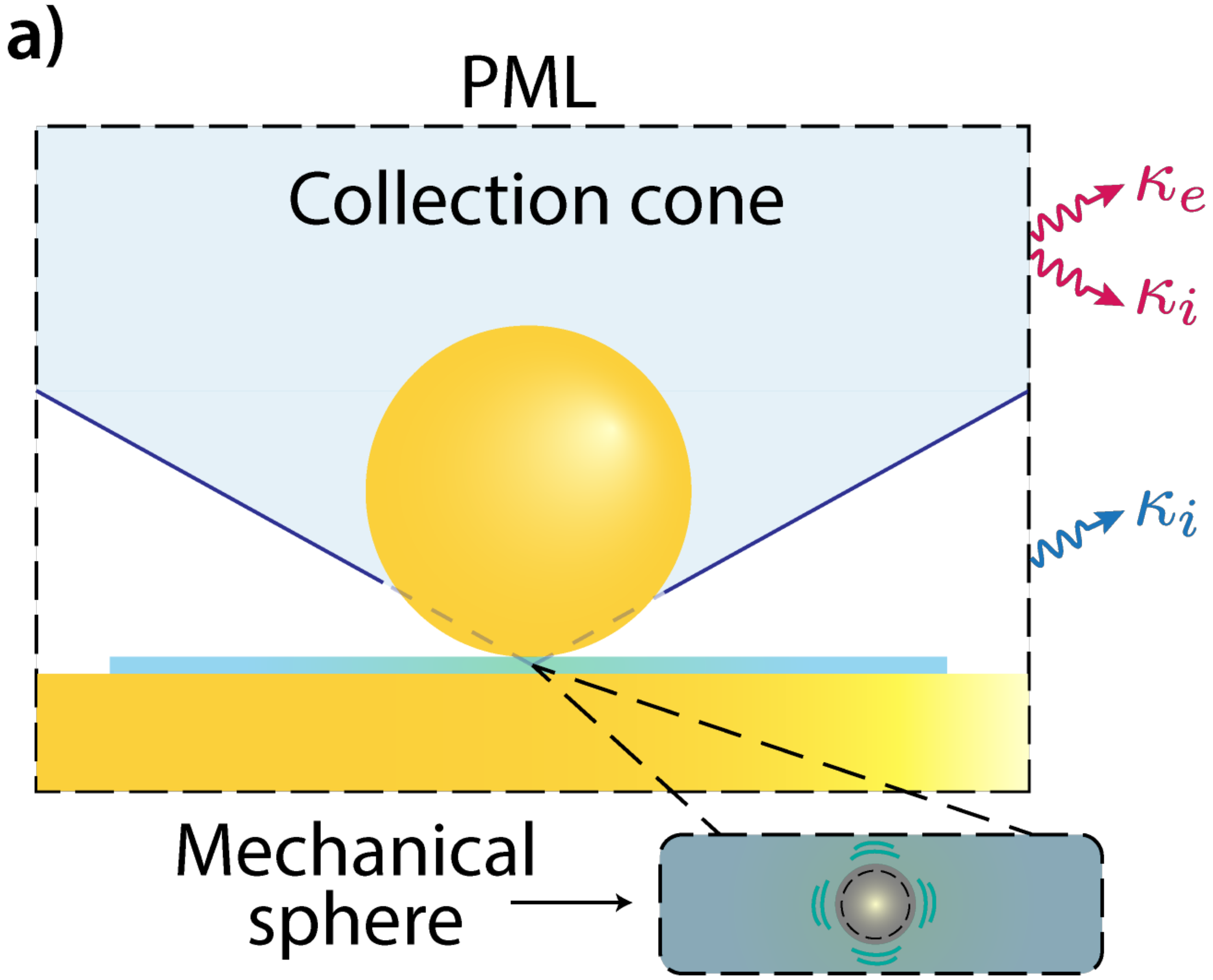
1. Analytic results for the optical and mechanical modes are available. In “fig_S2_a_b_mathematica.nb”, the exact and perturbation theory moving boundary optomechanical couplings are evaluated. This data is then used as benchmark for our Comsol simulations.
2. In “fig_S1_b_mathematica.nb” we use the same calculations to analyse the convergence of the perturbation series. For that purpose, we gradually increase the boundary deformations on the cylinder and compare exact and perturbation theory predictions.

Python:

1. Using “fig_S2_plot_data.ipynb” we import the Comsol data found in “fig_S2_a_b_data.txt”. Results from the analytic calculations were already incorporated in the .ipynb file.
2. Run the code to generate plots.

NPoM – Dielectric nanosphere

File “comsol_simulation_file_figS3_sphere” and “comsol_simulation_file_figS3_sphere_no_PML.mph” follow the same implementantions used in Fig. 4. A dielectric sub-nanometer sphere is placed at the center of the gap between the sphere and mirror. Its radial mechanical breathing mode is computed and used as input for Moving-Mesh calculations. A collection cone is defined in the geometry builder and is used to compute radiative decay rates in its inside and outside.



Generating data from Comsol:

1. Under Global Definitions -> Parameters 1 -> NA: set the desired numerical aperture desired. Note that for the purposes of the simulation absent of PML implementation, the numerical aperture plays no role and thus this step may be ignored.
2. Run *Study 2.* This will compute the mechanical mode used as input for the Moving-Mesh module.
3. Run *Study 1.* A table will be automatically generated and should be exported.
4. Exported data can be found in “sphere_FigS3_NA_XX.txt” or “sphere_FigS3_noPML.txt”, depending on which of the Comsol files were run. Here, XX stands for the chosen numerical aperture.
5. Be sure that *transform point* is on and set to *freq._guess. Transform point* can be found in *Solver configurations -> Eigenvalue Solver 1 (or 2).*

Python:

1. Files generated in Comsol are loaded on file “fig_S3_plot_data.ipynb” and plots are generated.