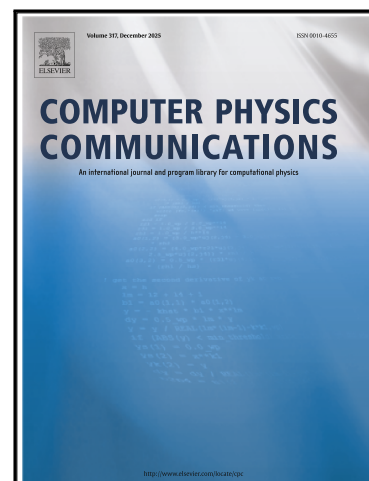


Journal Pre-proof

plasmonX: an Open-Source Code for Nanoplasmonics

Tommaso Giovannini, Pablo Grobas Illobre, Piero Lafiosca,
Luca Nicoli, Luca Bonatti, Stefano Corni, Chiara Cappelli

PII: S0010-4655(26)00017-2
DOI: <https://doi.org/10.1016/j.cpc.2026.110035>
Reference: COMPHY 110035



To appear in: *Computer Physics Communications*

Received date: 14 October 2025
Revised date: 5 January 2026
Accepted date: 12 January 2026

Please cite this article as: Tommaso Giovannini, Pablo Grobas Illobre, Piero Lafiosca, Luca Nicoli, Luca Bonatti, Stefano Corni, Chiara Cappelli, plasmonX: an Open-Source Code for Nanoplasmonics, *Computer Physics Communications* (2026), doi: <https://doi.org/10.1016/j.cpc.2026.110035>

This is a PDF of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability. This version will undergo additional copyediting, typesetting and review before it is published in its final form. As such, this version is no longer the Accepted Manuscript, but it is not yet the definitive Version of Record; we are providing this early version to give early visibility of the article. Please note that Elsevier's sharing policy for the Published Journal Article applies to this version, see: <https://www.elsevier.com/about/policies-and-standards/sharing#4-published-journal-article>. Please also note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Elsevier Logo

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Cover Thumbnail

plasmonX: an Open-Source Code for Nanoplasmonics

Tommaso Giovannini^{a,*}, Pablo Grobas Illobre^b, Piero Lafiosca^b, Luca Nicoli^b, Luca Bonatti^b, Stefano Corni^{c,d}, Chiara Cappelli^{b,**}^a Department of Physics and INFN, University of Rome Tor Vergata, Via della Ricerca Scientifica, 1, 00133, Rome, Italy^b Scuola Normale Superiore, Piazza dei Cavalieri 7, 56126, Pisa, Italy^c Department of Chemical Sciences, University of Padova, via Marzolo 1, Padova, Italy^d CNR Institute of Nanoscience, via Campi 213/A, Modena, Italy

ARTICLE INFO

Editor: Dr W Jong.

Keywords:
 plasmonics
 atomistic
 implicit
 nanoparticles
 graphene
 bimetallic

ABSTRACT

We present the first public release of `plasmonX`, a novel open-source code for simulating the plasmonic response of complex nanostructures. The code supports both fully atomistic and implicit descriptions of nanomaterials. In particular, it employs the frequency-dependent fluctuating charges (ω FQ) and dipoles (ω FQF μ) models to describe the response properties of atomistic structures, including simple and *d*-metals, graphene-based structures, and multi-metal nanostructures. For implicit representations, the Boundary Element Method is implemented in both the dielectric polarizable continuum model (DPCM) and integral equation formalism (IEF-PCM) variants. The distribution also includes a post-processing module that enables analysis of electric field-induced properties such as charge density and electric field patterns.

PROGRAM SUMMARY Program Title: plasmonXCPC Library link to program files: <https://doi.org/10.17632/zcd8fb4457.1>Developer's repository link: <https://github.com/plasmonX/plasmonX>

Licensing provisions: GPLv3

Programming language: Fortran 2008, Python

Nature of problem: Simulating the response properties of plasmonic metallic and graphene-based nanomaterials.

Solution method: Fully atomistic frequency-dependent fluctuating charges (ω FQ) [1,2] and dipoles (ω FQF μ) [3] models and implicit, non-atomistic Boundary Element Methods (BEM) [4]. The approaches are implemented within the quasistatic approximation.

Additional comments including restrictions and unusual features: The program has been mainly tested by using gfortran (versions 9-13) combined with the Math Kernel Library (MKL) provided by Intel.

CE: Please
 check
 instructions for
 correctness.

References

- [1] T. Giovannini, M. Rosa, S. Corni, C. Cappelli, A classical picture of subnanometer junctions: an atomistic Drude approach to nanoplasmonics, *Nanoscale* 11 (13) (2019) 6004–6015
- [2] T. Giovannini, L. Bonatti, M. Polini, C. Cappelli, Graphene plasmonics: Fully atomistic approach for realistic structures, *J. Phys. Chem. Lett.* 11 (18) (2020) 7595–7602.
- [3] T. Giovannini, L. Bonatti, P. Lafiosca, L. Nicoli, M. Castagnola, P. G. Illobre, S. Corni, C. Cappelli, Do we really need quantum mechanics to describe plasmonic properties of metal nanostructures?, *ACS Photonics* 9 (9) (2022) 3025–3034.
- [4] F. J. García de Abajo, A. Howie, Retarded field calculation of electron energy loss in inhomogeneous dielectrics, *Phys. Rev. B* 65 (11) (2002) 115418.

1. Introduction

Plasmonic nanostructures, ranging from metal nanoparticles to graphene-based materials, have attracted significant interest due to their unique ability

to confine light at the nanoscale, enabling applications ranging from enhanced spectroscopies and sensing to quantum optics and nanoelectronics [1–9]. The accurate simulation of their optical response, particularly in regimes where

* Corresponding author.

** Corresponding author.

quantum effects such as tunneling and nonlocality become relevant [10–12, 13, 13–22], remains a major theoretical and computational challenge.

Several numerical approaches have been developed to describe the response of plasmonic nanosystems [23–30]. The most accurate methods rely on *ab initio* quantum mechanical (QM) techniques, generally defined within the framework of time-dependent density functional theory (TDDFT) and its approximation (TD density functional tight binding – TDDFTB) [31–40]. While these approaches allow for a rigorous treatment of the QM effects underlying the plasmonic response, their huge computational cost drastically limits the nanostructure sizes that can be simulated, which are generally composed of less than 1000 atoms [39], and are thus far from realistic, experimentally studied nanosystems. An attempt to overcome this limitation is offered by semiclassical and “non-classical” electromagnetic models, which have been developed to incorporate key quantum corrections at a reduced computational cost [41–44]. These include nonlocal, quantum hydrodynamic descriptions (and related extensions), quantum-corrected models that effectively account for tunneling in narrow gaps, and surface-response formulations based on Feibelman parameters [45–52].

As an alternative, purely classical, non-atomistic approaches have been developed, such as the Discrete Dipole Approximation (DDA) [53], the Finite-Difference Time-Domain (FDTD) [54], and the Boundary Element Method (BEM) [55–60], which model the nanostructure as a continuum dielectric characterized by its complex permittivity function. Such approaches are rooted in electrodynamics and solve Maxwell’s equations by imposing the proper boundary conditions. Because they are based on classical physics, they can indeed afford large nanostructures; however, they lack any description of atomistic features and QM effects, and therefore, might fail to accurately describe regimes where quantum and mesoscopic effects become relevant [61–63], including electron tunneling, nonlocal screening, Landau damping, quantum confinement, and atomic-scale defects, such as picocavities [64–66].

To bridge the gap between quantum accuracy and classical scalability, we have recently proposed novel atomistic, yet classical approaches for nanoplasmonics, named frequency-dependent fluctuating charges (ω FQ)[67,68] and fluctuating dipoles (ω FQF μ) [69], which are specifically designed to account for the physical mechanisms underlying the optical response of plasmonic nanostructures. These methods offer a remarkable accuracy that is comparable to state-of-the-art *ab initio* TDDFT even for structures below the quantum size limit [58,69], and are also able to account for non locality and quantum effects thanks to an effective description of quantum tunneling [67,69], challenging the assumption that a full QM treatment is required to accurately model the plasmonic response. Also, being based on classical electrodynamics, ω FQF μ retains the positive aspects of classical approaches, i.e. it can be applied to nanostructure sizes that are untreatable at the *ab initio* level [66,70].

In this paper, we introduce `plasmonX`, a novel open-source code for simulating the plasmonic response of nanostructures, which features the first robust implementation of ω FQ and ω FQF μ models. In addition, `plasmonX` provides a solid implementation of continuum dielectric methods (BEM), in a unified and efficient framework. Specifically, the dielectric polarizable continuum model (DPCM) [71] and Integral Equation Formalism PCM (IEFPCM) [26,72–74] formalisms are implemented, supporting arbitrary geometries via mesh discretization. Differently from most simulation tools used in nanoplasmonics, which address plasmonic response either through continuum electrodynamics (e.g. MNPBEM [75]), or through general-purpose quantum-mechanical methods, `plasmonX` is designed to bridge atomistic classical models and continuum approaches within a single, scalable software platform. In particular, `plasmonX` is the first open-source software that allows atomistic classical simulations of nanoplasmonics. The code is written in modern Fortran 2008, with a Python-based interface for input parsing, and a CMake build system that ensures cross-platform compatibility. It supports multiple numerical solvers, including matrix inversion, Krylov subspace iterative methods [76,77], and a fully memory-efficient “on-the-fly” Generalized Minimum RESidual algorithm (GMRES) implementation [78], making it suitable for both small-scale and high-performance computing (HPC) applications for nanostructures of more than 1 million atoms [70].

The manuscript is structured as follows: we briefly recall the theoretical foundations of the methods implemented in `plasmonX`, its modular architecture, and input specification, followed by performance benchmarks and illustrative

simulations, including the optical properties of noble metal nanoalloys and field-enhancement in gold dimers. Finally, we introduce the `plasmonX` dedicated post-processing suite, which is distributed along with the main code, enabling visualization and analysis of induced densities and fields.

2. Theoretical Models

In this section, we provide a brief overview of the theoretical methods implemented in `plasmonX` (ω FQ, ω FQF μ , and BEM), highlighting their common theoretical formalism, which allows for a modular implementation.

2.1. ω FQ

ω FQ [58,67,68,79], endows each atom of the nanostructure with a complex charge that is a function of the frequency ω of the external perturbation. The charge exchange between the atoms is governed by the Drude conduction model, which describes the intraband contribution to plasmon decay. ω FQ also features an effective description of quantum tunneling by means of a Fermi-like damping function, which introduces the typical exponential decay as a function of the atom-atom distance. We refer the interested reader to Ref. [67] for the complete derivation of ω FQ equations.

The complex ω FQ charges q are obtained by solving the following complex linear system, written in a matrix form [70]:

$$[\bar{\mathbf{K}}\mathbf{T}^{qq} - z_q(\omega)\mathbf{I}_N]\mathbf{q}(\omega) = -\bar{\mathbf{K}}\mathbf{V}(\omega) \quad (1)$$

where $\bar{\mathbf{K}}$ is a $N \times N$ matrix whose elements read:

$$\bar{K}_{ij} = K_{ij} - \sum_k K_{ik}\delta_{ij} \quad (2)$$

where δ_{ij} is the Kronecker delta. The elements of the $\bar{\mathbf{K}}$ matrix take the following form:

$$K_{ij} = \begin{cases} \frac{[1-f(r_{ij})]A_i}{r_{ij}} & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases} \quad (3)$$

A_i is the effective area of atom i through which the charge flux passes, r_{ij} is the distance between the i -th and j -th atoms, while $f(r_{ij})$ is the Fermi-like damping function introduced to mimic quantum tunnelling:

$$f(r_{ij}) = \frac{1}{1 + \exp\left[-d\left(\frac{r_{ij}}{s \cdot r_{ij}^0} - 1\right)\right]} \quad (4)$$

where r_{ij}^0 is the nearest neighbor distance, a fixed quantity specific to each lattice. In Eq. 1, $\bar{\mathbf{K}}$ matrix is multiplied by \mathbf{T}^{qq} matrix that takes into account the charge-charge interactions [67,80]. In ω FQ, each charge q_i , placed at \mathbf{r}_i , is associated with a Gaussian-type distribution of width equal to R_{q_i} :

$$\rho_{q_i}(\mathbf{r}) = \frac{q_i}{\pi^{3/2}R_{q_i}^3} \exp\left(-\frac{|\mathbf{r} - \mathbf{r}_i|^2}{R_{q_i}^2}\right) \quad (5)$$

In this framework, the charge-charge interaction kernel \mathbf{T}^{qq} reads [80]:

$$T_{ij}^{qq} = \frac{1}{|\mathbf{r}_{ij}|} \operatorname{erf}\left(\frac{|\mathbf{r}_{ij}|}{R_{q_i - q_j}}\right) \quad (6)$$

where $R_{q_i - q_j} = \sqrt{R_{q_i}^2 + R_{q_j}^2}$. Note that the diagonal matrix of \mathbf{T}^{qq} is the chemical hardness. In Eq. 1, $z_q(\omega)$ is a diagonal complex shift (\mathbf{I}^N is the $N \times N$ identity matrix) that accounts for the Drude mechanisms:

$$z_q(\omega) = -\frac{\omega}{2n\tau}(\omega\tau + i) \quad (7)$$

where n and τ are the static density and scattering time, respectively. n can be expressed as:

$$n = \frac{\sigma_0 m^*}{\tau} = \frac{\omega_p^2 m^*}{4\pi} \quad (8)$$

where σ_0 is the static conductance of the nanomaterial, ω_p is the plasma frequency, and m^* is its effective electron mass. The latter can be approximated to

1 a.u. for metal nanoparticles [81], while for doped graphene-based substrates, it can be expressed as:

$$m^* = \frac{\sqrt{\pi n_{2D}}}{v_F} \quad (9)$$

where n_{2D} is the 2D electron density, and v_F is the Fermi velocity [82], which is related to the Fermi energy E_F through the expression $v_F = \sqrt{\frac{2E_F}{m_0}}$ (m_0 is the electron rest mass [82]). The 3D electron density for graphene can be written in terms of n_{2D} , which in turn depends on the value of E_F , which can be recovered directly from experimental conditions [83]. Finally, in Eq. 1, the right-hand side is made up of the product of the $\bar{\mathbf{K}}$ matrix and the external potential $\mathbf{V}(\omega)$ associated with the external field $\mathbf{E}(\omega)$.

2.2. ω FQF μ

ω FQ is suitable for materials whose electronic properties can be well described by a Drude mechanism, such as simple metals (e.g., sodium) and graphene-based nanostructures. To describe d -metals (Ag, Au), the contribution of interband transitions must be included [84]. To this end, we introduce an additional polarization source, a dipole moment μ , which, similar to the charge, is characterized by a Gaussian distribution of width equal to R_{μ_i} [69]:

$$\rho_{\mu_i}(\mathbf{r}) = \frac{|\mu_i|}{\pi^{3/2} R_{\mu_i}^3} \hat{n}_i \cdot \nabla_{\mathbf{r}_i} \left[\exp\left(-\frac{|\mathbf{r} - \mathbf{r}_i|^2}{R_{\mu_i}^2}\right) \right] \quad (10)$$

In the resulting ω FQF μ approach [69], the charges and dipole equations are solved simultaneously, yielding the following complex linear system:

$$\begin{bmatrix} \bar{\mathbf{K}} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{3N} \end{bmatrix} \begin{pmatrix} \mathbf{T}^{q\mu} & \mathbf{T}^{\mu\mu} \\ \mathbf{T}^{\mu q} & \mathbf{T}^{\mu\mu} \end{pmatrix} - \begin{pmatrix} z_q(\omega)\mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & z_\mu(\omega)\mathbf{I}_{3N} \end{pmatrix} \begin{pmatrix} \mathbf{q}(\omega) \\ \boldsymbol{\mu}(\omega) \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{K}} \\ \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{V}(\omega) \\ \mathbf{0} \end{pmatrix} \quad (11)$$

where $\mathbf{T}^{q\mu}$, and $\mathbf{T}^{\mu\mu}$ are the charge-dipole and dipole-dipole interaction kernels, which can be derived by proper differentiation of \mathbf{T}^{qq} in Eq. 6.[80,85] In Eq. 11, $z_\mu(\omega)$ is defined as:

$$z_\mu(\omega) = -\frac{1}{\alpha^{IB}(\omega)} \quad (12)$$

where $\alpha^{IB}(\omega)$ is the interband polarizability that is extracted from the interband permittivity function, either calculated or experimentally measured [69].

Eq. 11 is valid if all the system's atoms are of the same nature, i.e., for a homogeneous nanomaterial. To describe heterogeneous systems, i.e. characterized by two atom types, Eq. 11 must be recast in the following form [86]:

$$\begin{bmatrix} \left(\frac{1}{2}(\bar{\mathbf{K}} + \bar{\mathbf{H}}(\omega))\right) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{3N} \end{bmatrix} \begin{pmatrix} \mathbf{T}^{q\mu} & \mathbf{T}^{\mu\mu} \\ \mathbf{T}^{\mu q} & \mathbf{T}^{\mu\mu} \end{pmatrix} - \begin{pmatrix} z_q(\omega)\mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & z_\mu(\omega)\mathbf{I}_{3N} \end{pmatrix} \begin{pmatrix} \mathbf{q}(\omega) \\ \boldsymbol{\mu}(\omega) \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{K}} \\ \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{V}(\omega) \\ \mathbf{0} \end{pmatrix} \quad (13)$$

where $z_{\mu,i}(\omega) = -1/\alpha_i^{IB}(\omega)$ and:

$$\alpha_i^{IB}(\omega) \equiv \alpha_{A,i}^{IB,hetero}(\omega) = \left(\frac{N_{A,i}+1}{\alpha_A^{IB}(\omega)} + \frac{N_{B,i}}{\alpha_B^{IB}(\omega)} \right)^{-1} \quad (14)$$

where $N_{A,i}$ and $N_{B,i}$ are the number of nearest neighbors of A and B type, respectively. Furthermore, in Eq. 13 the following quantities and matrices are introduced:

$$\bar{H}_{ij}(\omega) = H_{ij}(\omega) - \sum_k H_{ik}(\omega)\delta_{kj} \quad (15)$$

$$H_{ij}(\omega) = \frac{n_j \tau_j (1 - i\omega\tau_i)}{n_i \tau_i (1 - i\omega\tau_j)} K_{ji} \quad (16)$$

It is worth noting that, differently from ω FQ and ω FQF μ , the off-diagonal elements of the left-hand side matrix in this case depend on the external frequency; however, this does not overall affect the general structure of the linear system. Furthermore, Eq. 13 reduces to Eq. 11 for a homogenous system.

2.3. Boundary Element Method

In BEM, the plasmonic nanostructure is represented as a continuous surface which is discretized into a mesh of triangular elements known as *tesserae*. The key quantities computed in this approach are the surface charges, located at the barycenter of each *tessera*. These surface charges determine the system's optical response and depend on two factors: the geometry of the nanostructure and the complex, frequency-dependent permittivity of both the material and its surrounding medium.[26,71,87]

In *plasmonX*, two BEM methods are implemented: the Dielectric Polarizable Continuum Model (DPCM) [71,87,88] and the Integral Equation Formalism PCM (IEFPCM) [71,74]. In the former, the BEM charges are calculated by solving the following linear system of equations:

$$\left[2\pi \begin{pmatrix} \varepsilon_2(\omega) + \varepsilon_1(\omega) \\ \varepsilon_2(\omega) - \varepsilon_1(\omega) \end{pmatrix} \mathbf{A} + \mathbf{F} \right] \boldsymbol{\sigma}(\omega) = -\mathbf{E}_{\bar{n}}(\omega). \quad (17)$$

while in IEFPCM, the charges are instead obtained as follows:

$$\left[2\pi \begin{pmatrix} \varepsilon_2(\omega) + \varepsilon_1(\omega) \\ \varepsilon_2(\omega) - \varepsilon_1(\omega) \end{pmatrix} \mathbf{I}_N + \mathbf{F} \right] \mathbf{S} \mathbf{A}^{-1} \boldsymbol{\sigma}(\omega) = -(2\pi \mathbf{I}_T + \mathbf{F}) \mathbf{V}(\omega). \quad (18)$$

In Eqs. 17 and eq. (18), $\boldsymbol{\sigma}(\omega)$ are the discretized BEM surface charges, $\varepsilon_2(\omega)$ and $\varepsilon_1(\omega)$ are the frequency-dependent dielectric permittivities of the metal and the surrounding environment, respectively. The matrix \mathbf{A} is diagonal and contains the area of each *tessera*, and \mathbf{I}_N denotes the identity matrix for a system composed of N *tesserae*. In the case of BEM in DPCM formulation, $\mathbf{E}_{\bar{n}}$ represents the component of the external electric field projected along the vector normal to each *tessera*. In the IEFPCM formulation, \mathbf{V} corresponds to the electric potential evaluated at the barycenter of each *tessera* [71,89].

The \mathbf{S} and \mathbf{V} matrices depend on the geometric configuration of the mesh. The elements of the \mathbf{S} matrix are given by [71,89]:

$$S_{ij} = \begin{cases} A_j, & \text{if } i = j \\ \frac{A_j}{|\mathbf{s}_i - \mathbf{s}_j|}, & \text{if } i \neq j \\ 1.0694\sqrt{4\pi A_i}, & \text{if } i = j \end{cases}$$

where A_i , and \mathbf{s}_i are the area and position of the i -th *tessera*, respectively.

Two definitions of the \mathbf{F} matrix, which contains the surface derivative of the Green function, are implemented in *plasmonX*. The first is an "approximate" method, valid for spherical and other simple-shaped geometries (e.g., rods), which evaluates the matrix elements as [71,89]:

$$F_{ij} = \begin{cases} \frac{(\mathbf{s}_i - \mathbf{s}_j) \cdot \bar{\mathbf{n}}_j}{|\mathbf{s}_i - \mathbf{s}_j|^3} \cdot A_j, & \text{if } i \neq j \\ 1.0694 \cdot \frac{\sqrt{4\pi A_i}}{2R}, & \text{if } i = j \end{cases}$$

where $\bar{\mathbf{n}}_i$ is the normal vector of the i -th *tessera*, and R denotes the radius of the sphere enclosing the entire set of *tesserae* (Fig. 1). It is worth mentioning that some formulations instead determine the diagonal elements through linear combinations of the off-diagonal terms [90].

The second implementation involves a numerically more accurate evaluation of the surface derivative of the quasistatic Green function, which we refer to as the "accurate" method. In this case, the entries of the \mathbf{F} matrix are computed by solving the following surface integral [75,87]:

$$\int_{\Omega} \frac{\partial G(\mathbf{s}, \mathbf{s}')}{\partial n} d\Omega, \quad G(\mathbf{s}, \mathbf{s}') = \frac{1}{|\mathbf{s} - \mathbf{s}'|} \quad (19)$$

where $G(\mathbf{s}, \mathbf{s}')$ is the quasistatic Green function evaluated between surface points \mathbf{s} and \mathbf{s}' . The symbol Ω denotes the surface of the nanoparticle, and ∂n indicates differentiation with respect to the surface normal. For additional details on the "exact" integration procedure, the reader is referred to Refs. [87] and [75].

3. Getting Started

plasmonX is written in Fortran 2008 and is designed to be compiled and executed on multiple platforms with minimal user intervention. To facilitate

Table 1: List of the input options for running `plasmonX.py` script.

Option	Required	Description
<code>-i</code>	Yes	Input file in YAML format
<code>-o</code>	No	Output file name (default: <code>input_file.log</code>)
<code>-omp</code>	No	Number of OpenMP threads to use (default: all available)
<code>-mem</code>	No	Available Memory in GB (default: system maximum)
<code>-h</code>	No	Print help message and exit

its usage, the code architecture integrates a Python-based interface for input parsing and a CMake-based build system. The Python front-end is responsible for reading the simulation parameters from a human-readable YAML input file. This modular approach allows for a clear separation between input specification and high-performance numerical execution in the Fortran backend.

The full source code is available on GitHub [91] and makes use of two external components managed as git submodules. The pre-release version of `plasmonX` associated with this manuscript is archived on Zenodo [92]. To ensure a complete and functional installation, it is highly recommended to clone the repository using Git over SSH with the `--recursive` option. Once the repository is cloned, a minimal Python environment must be configured. All required dependencies are listed in a `requirements.txt` or `requirements.yaml` file, depending on whether the user chooses `pip` or `conda` for package management.

The compilation of the Fortran code is handled entirely through CMake [93] with a custom Bash script (`setup.sh`) which is provided to streamline configuration across different operating systems. This script sets up the build directory, detects the available compilers, applies appropriate flags (e.g., OpenMP support), and runs the CMake configuration, with the default compiler flags and recommended build settings documented in the `plasmonX` user manual (see Ref. [97]). The project is then compiled via standard `make` targets. The internal test suite is currently characterized by 83 tests, which ensure the correct compilation and linking with the external submodules. The tests are run by using `ctest`, which exploits the `runtest` package [94], which is set as a submodule of the project. For each test case, the test suite includes a reference run log, generated on a Dell XPS 13, 11th Gen Intel(R) Core(TM) i7-1185G7 @ 3.00GHz, 32 GB RAM, which reports the wall-time at the end of the run, allowing users to compare results and sanity-check their installation.

The code supports Linux, macOS, and Windows environments (via Windows Subsystem for Linux) [95]. It has been tested with gfortran versions 9 through 13, and it is compatible with both standard LAPACK/BLAS libraries and the Intel Math Kernel Library (MKL) [96], which is highly recommended for enhanced performance on Linux and Windows environments. The complete configuration and compilation workflow depends on the architecture and is amply documented within the user manual [97].

4. Running and Input Configuration

Once installed and tested, the `plasmonX` driver script can be executed by using the following command:

```
PATH/T0/BUILD/plasmonX.py [-h] -i input_file.yaml
                        [-o output_file]
                        [-omp omp_threads]
                        [-mem memory_available]
```

The only required option is `-i`, which specifies the YAML input file. Optional arguments allow the user to set the output filename, number of OpenMP threads, and available memory for the calculation (see also Tab. 1). The default output filename is `input_file.log`, while the number of OpenMP threads and the available memory are set to the maximum available in the architecture exploited for the calculation. A help message can be accessed with the `-h` flag.

The input to `plasmonX` is defined entirely by a YAML-formatted configuration file, typically named `input_file.yaml`. The input is composed of various sections, each dedicated to a specific aspect of the simulation.

what:

```
\# defines the type of calculation
```

Algorithm 1 Structure of `plasmonX`.

```
1: Read the input file
2: Build target object from input, through runtime polymorphism
   target_ ← BEM / ωFQ / ωFQFμ
3: if Algorithm = inversion then
4:   Solve each frequency by direct inversion
5: else if Algorithm = iterative then
6:   if Algorithm = iterative on the fly then
7:     Iterative solve with on-the-fly procedure
8:   else
9:     Iterative solve with on-memory procedure
10:  end if
11: end if
12: Clean scratch; save data for post processing; deallocations
```

```
energy, static/dynamic response, restart]
forcefield:
\# defines the static/dynamic forcefield and the interaction
field:
\# defines the field parameters [static/dynamic, frequency]
algorithm:
\# defines the numerical algorithm to solve response
output:
\# defines the parameters of the output printing
control:
\# defines the parameters to control the calculation of a
atom_types:
\# defines the parameters of the atom types for parameters: simulation
\# defines the parameters of the \wfq and \omegaFQFμ models input
\# describes the atomic or implicit geometry in \AAngstrom
```

This structure allows the user to define atomistic and continuum models in a unified format, while also keeping the input compact and readable. A number of default values (see `plasmonX` documentation [97]) are automatically provided if certain sections are omitted, enabling simple and minimal configurations for the most user-friendly experience possible. To provide an overview of the `plasmonX` structure, Algorithm 1 summarizes the main driver workflow of the code. Further details on each section, including a complete list of available keywords and examples of `plasmonX` minimal input (i.e., exploiting default values), are provided in the documentation and examples (see Ref. [97]), and are discussed in the following.

5. Specify the Nanostructure

`plasmonX` potentially supports any kind of nanostructure by providing the atomic positions (for ωFQ and $\omega FQF\mu$ simulations) or the mesh tessellation (for BEM). Atomic positions are provided in the `input_geometry` section, either by explicitly specifying the xyz coordinates by YAML inline string, or by specifying an XYZ file (case sensitive), which allows referencing external standard-format coordinates. The BEM mesh file is generally provided in the BEM section, under the keyword `mesh`. In this case, only msh files formatted in version 2.2 0 8 (ASCII and double precision) are allowed. `plasmonX` is agnostic to the specific material or symmetry of the nanostructure, and it is capable of handling arbitrary arrangements of atoms or continuum morphologies. The input geometry and the mesh are interpreted in units of Ångstrom.

Fig. 1: Nanostructure morphologies that can be created by `p1asmonX` through the `GEOM` interface.

Fig. 2: Elapsed time required to solve the ω FQ linear system as a function of the number of atoms for selected graphene disks at the plasmon resonance frequency by using inversion and iterative (on the fly) algorithms implemented in `p1asmonX`.

`p1asmonX` also provides a user-friendly input specification through the integrated interface to the `GEOM` library (commit: 1cc34d1 [98]). This interface enables the construction of both implicit and atomistic geometries by user specification of the relevant parameters, e.g., radius, shape, etc. For the implicit case, the allowed geometries are spheres and rods that are defined by specifying their type (e.g., `sphere`, `rod`) along with required parameters such as radius, length, orientation axis, and mesh resolution (we refer the interested user to the manual for further information). For atomistic geometries, `GEOM` supports a rich catalog of shapes, including spheres, rods, pyramids, icosahedra, cones, tips, and complex assemblies resembling experimental probe configurations (e.g., TERS tips). The supported morphologies through `GEOM` interface are graphically depicted in Fig. 1. The generator supports both homogeneous systems (single chemical composition) and, for some specific morphologies, heterogeneous systems (core-shell systems), with user-defined atom types for each region. Furthermore, carbon-based nanomaterials, such as graphene disks, rings, triangles, and ribbons, can be created with edge-type specification (armchair, zigzag). Again, we refer the interested user to the manual for details on the specific morphology [97]. In all cases, the resulting geometries are passed to the Python-based preprocessor, which parses the YAML input and creates the specific geometry (atomistic or mesh) through the `GEOM` interface, for the subsequent computation in the Fortran backend.

6. Solvers

All the methods implemented in `p1asmonX` are characterized by a similar formal structure [99]. In fact, in all cases, the response variables (charges and dipoles) are obtained by following a complex linear system that can be written as:

$$[\mathbf{M}(\omega) - \mathbf{Z}(\omega)]\mathbf{L}(\omega) = \mathbf{U}(\omega)\mathbf{R}(\omega) \quad (20)$$

The definition of all matrices and vectors in Eq. 20 can be recovered by comparing with Eq. 1 (ω FQ), Eq. 11 (ω FQF μ for homogeneous systems), Eq. 13 (ω FQF μ for heterogeneous systems), Eq. 17 (BEM in DPCM formulation), and Eq. 18 (BEM in IEFPCM formulation). We note that for ω FQF μ for heterogeneous systems (Eq. 13) the \mathbf{M} and \mathbf{U} matrices are frequency dependent and complex, while they are static and real for all other methods. $\mathbf{Z}(\omega)$ and $\mathbf{L}(\omega)$ are frequency-dependent and complex in all cases, while $\mathbf{R}(\omega)$ is always real and contains the external polarization sources, such as the external potential and field.

To solve Eq. 20, `p1asmonX` implements three algorithms, which are common in this kind of simulations [70,100–102]:

1. Inversion: the complex linear system is solved by resorting to LU factorization and subsequent resolution of the complex linear systems through an interface to LAPACK subroutines (ZGETRF and ZGETRS). This is the default for small systems.
2. Iterative: the complex linear system is solved by resorting to the iterative Krylov GMRES method [78]. `p1asmonX` provides an effective implementation of complex GMRES for all methods by storing the matrices in memory [70].
3. Iterative on the fly: the complex linear system is solved by resorting to GMRES [78], and the matrix product multiplications are performed on the fly, without storing the matrices in memory [70]. This is implemented for ω FQ and ω FQF μ methods only, and is the default for large systems.

For both iterative solvers, the numerical convergence is controlled by the residual norm of the linear system by means of the root-mean-square error

Fig. 3: Speedup of the ω FQF μ linear system solution using the iterative on-the-fly algorithm as a function of the OMP threads.

(RMSE). A default tolerance of $|E_0| \cdot 10^{-5}$ is employed (where $|E_0|$ is the applied electric field intensity), which we have previously shown to provide a reliable compromise between numerical accuracy and computational cost [70]. In Fig. 2, we provide a graphical depiction of the elapsed time (in seconds) of inversion and iterative (on-the-fly) algorithms as a function of the number of atoms of graphene disks with a radius ranging from 10 nm to 19 nm (from 10k to 40k atoms). The calculations are performed at the plasmon resonance frequency of each disk (10k atoms: 0.30 eV; 20k atoms: 0.26 eV; 30k atoms: 0.24 eV; 40k atoms: 0.22 eV) by imposing a Fermi Energy of 0.4 eV. For the iterative solution, the Root Mean Squared Error (RMSE) is set to 10^{-8} for an applied electric field with $E_0 = 10^{-4}$ a.u., which we have previously shown to provide the best compromise between computational cost and accuracy [70]. All calculations are performed on a machine equipped with an Intel(R) Xeon(R) Gold 6252N CPU @ 2.30GHz, 96 threads, and 512 GB of RAM. As can be seen for the smallest system (~ 10 k atoms), the inversion and iterative algorithms require almost the same elapsed time (16 s vs. 15 s). By increasing the size, the inversion algorithm follows a third power scaling ($y \propto P(x^3)$, $R^2 > 0.99$), while the iterative one follows a second power progression ($y \propto P(x^2)$, $R^2 > 0.99$), rapidly becoming the most efficient algorithm to solve the linear complex problem. For a detailed analysis of the performance of the GMRES algorithm in solving the ω FQ linear system, we refer the interested reader to Ref. [70].

The iterative on-the-fly algorithm is especially useful for large systems because the requirements in terms of RAM are particularly low, as only a few vectors are stored in memory during the run. To showcase the performance of `p1asmonX` implementation, we perform 9 calculations by increasing the number of processors from 10 to 90 with steps of 10. Calculations employ the ω FQF μ approach and are run on a Silver nanoparticle with icosahedral shape with a radius of about 10 nm, i.e. composed of about 149k atoms (about 600k variables). We then compute the speedup of the calculation as follows:

$$\text{speedup}(n) = \frac{T(n)}{T(10)} \quad (21)$$

where $T(n)$ is the elapsed time exploiting n OMP threads, while $T(10)$ is the reference obtained by using 10 OMP threads. Ideally, by moving from 10 to 90 OMP threads, the maximum speedup is 9. The computed speedups are graphically depicted in Fig. 3. The code demonstrates a nearly linear speedup up to 60 threads, with parallel efficiency (calculated as $\text{speedup}(n)/(n/10)$) remaining above 90%. Beyond this point, the efficiency gradually decreases, reaching approximately 84% at 90 threads. This decrease is consistent with the presence of parallel overheads and resource contention effects, which are typical in simulations of this kind [103]. Nevertheless, the observed speedup of 7.58 at 90 threads (relative to 10 threads) indicates that the code maintains a high degree of parallel effectiveness, which can be exploited in HPC applications.

7. Observables

Once the complex linear system defining each model is been solved for a specific ω , the total complex dipole moment $\bar{\mathbf{d}}(\omega)$ of the system can be calculated as [67,69]:

$$\omega\text{FQ} : \quad \bar{\mathbf{d}}(\omega) = \sum_i q_i(\omega)\mathbf{r}_i \quad (22)$$

$$\omega\text{FQF}\mu : \quad \bar{\mathbf{d}}(\omega) = \sum_i q_i(\omega)\mathbf{r}_i + \boldsymbol{\mu}_i(\omega) \quad (23)$$

$$\text{BEM} : \quad \bar{\mathbf{d}}(\omega) = \sum_i \sigma_i(\omega)\mathbf{s}_i \quad (24)$$

from which the complex polarizability $\bar{\alpha}$ is obtained:

$$\bar{\alpha}_{kl}(\omega) = \frac{\partial \bar{d}_k(\omega)}{\partial E_l(\omega)} \quad k, l \in \{x, y, z\} \quad (25)$$

Fig. 4: (a) Graphical depiction of spherical Ag-Au nanoalloys. (b) ω FQF μ plasmon resonance frequency (PRF) as a function of Au percentage.

From $\bar{\alpha}$, absorption, scattering, and extinction cross sections can finally be evaluated as:

$$\sigma^{\text{abs}}(\omega) = \frac{4\pi\omega}{3c} \text{Tr} \{ \text{Im} \bar{\alpha}(\omega) \} \quad (26)$$

$$\sigma^{\text{sca}}(\omega) = \frac{8\pi\omega^4}{3c^4} |\bar{\alpha}(\omega)|^2 \quad (27)$$

$$\sigma^{\text{ext}}(\omega) = \sigma^{\text{abs}}(\omega) + \sigma^{\text{sca}}(\omega) \quad (28)$$

where c is the speed of light. Since `plasmonX` implements methods in the quasi-static regime, the scattering cross section is generally negligible.

To showcase the results that can be obtained with `plasmonX` we consider ten Ag-Au spherical nanoalloys with diameter $D = 3.5$ nm (1289 atoms) which are constructed by increasing the percentage of Au atoms in a Ag spherical nanoparticle created by using `GEOM` interface (see Fig. 4a), randomly replacing Ag atoms (from 0% to 100%, with a constant step of 10%). Once the structures are created, the ω FQF μ calculation is performed by running the following minimal input:

```
what:
  - dynamic response
```

```
forcefield:
  dynamic: wFQFMu
```

```
field:
  min freq: 2.0
  max freq: 4.5
  step freq: 0.01
```

```
input_geometry:
  external xyz file: Au_percentage.xyz
```

where `percentage` corresponds to the Au percentage of each nanoalloy. The Au-Ag parameters are set to the default values and are taken from Refs. [69] and [86].

The computed absorption cross sections σ^{abs} for all the studied alloys are reported in Fig. 4b. All spectra are characterized by a main peak, which redshifts and lowers in intensity as the Au concentration increases. By plotting the PRF (in nm) as a function of the percentage of Au (see Fig. 4b), a linear trend is observed ($R^2 > 0.99$, see inset). This is in agreement with Vegard's law, which is reported by various experiments [104–107]:

$$\lambda^{\text{Vegard}}(x) = (1-x)\lambda_{\text{Ag}} + x\lambda_{\text{Au}} \quad (29)$$

where x represents the Au percentage. The linear slope (1.88 nm/% Au) of the fitted line is also in very good agreement with experiments reported in Ref. [104] (~ 1.35 nm/% Au). Also, the computed intensity of the plasmon band exponentially decreases by increasing the Au percentage, perfectly reproducing experimental findings [104]. The excellent agreement provided by ω FQF μ is remarkable, especially considering that other classical approaches fail at reproducing the experimentally reported linear trend in wavelength and the exponentially decreasing intensity [108]. We conclude this section by highlighting that the considered atomistic structures are intentionally small (1289 atoms) to demonstrate that physically relevant results, that can be directly compared with experiments, can be obtained by running `plasmonX` on a personal laptop (in this case, Dell XPS 13" equipped with 11th Gen Intel(R) Core(TM) i7-1185G7 @ 3.00GHz and 32GB of RAM).

8. Post-process Analysis

By default, `plasmonX` creates a `tar.gz` archive where the most relevant information of the calculation is stored, including that for post-process analysis. `plasmonX` is distributed together with a post-process code, named

`plasmonX_analysis`, that gives access to various analyses, such as the extraction of XYZ and PQR files (for ω FQ and BEM calculations), CUBE and PLT files for plotting induced densities and fields, and CSV files for the analysis of the induced densities and fields in user-defined planes. The code is compiled together with the main code `plasmonX`, and uses the modularity and object-oriented implementation to access the specificities of each model. As for the main code, the user-provided input is managed by a Python interface. All options are presented and discussed in the user manual.

To showcase the performance of `plasmonX_analysis`, we select an atomistic Gold Ih dimer (gap separation along the x axis of 5 Å), constructed by using the `GEOM` interface and described at the ω FQF μ level (see Fig. 5a). The resulting dimer is composed of 1122 atoms. We highlight that ω FQF μ can accurately describe similar systems characterized by a subnanometer gap [10,11,14,18–20,67,109], as it physically accounts for quantum tunneling effects (see Eq. 4) [67]. The ω FQF μ calculation is performed by using the following minimal input, which exploits the default Au ω FQF μ parameters taken from Ref. [69]:

```
what:
  - dynamic response
```

```
forcefield:
  dynamic: wFQFMu
```

```
field:
  min freq: 2.0
  max freq: 3.0
  step freq: 0.01
```

```
input_geometry:
  shape: Ih
  atomtype: Au
  radius: 15.0
  distance: 5.0
  dimer_axis: +x
```

The resulting absorption cross section is graphically depicted in Fig. 5b, and clearly shows a main plasmon peak at 2.28 eV. By using `plasmonX_analysis`, the nature of the plasmon peak can be analyzed by calculating the density induced at the PRF by an external field polarized along the x axis. The resulting induced density is graphically depicted in Fig. 5c, and is obtained by running the following command:

```
PATH/TO/BUILD/plasmonX_analysis.py -i output_file.tar.gz \
-freq 2.28 -field_dir x
```

which creates a CUBE file that can be used to graphically plot the induced density. The graphical rendering can be performed by using an external visualization tool (e.g. `VMD`[110], as in the present case). Fig. 5c shows that the plasmon peak is associated with a boundary dipolar plasmon (BDP), i.e. the two plates are characterized by an induced plasmon with dipolar character in the direction of the polarization field. As stated above, `plasmonX_analysis` also gives access to the calculation of the induced field, which is a fundamental quantity for surface-enhanced spectroscopies [5,111–113]. In this case, we compute the electric field in the xy plane as induced by the same external field as above, by using the following command:

```
PATH/TO/BUILD/plasmonX_analysis.py -i output_file.tar.gz -w
-freq 2.28 -field_dir x -p
-n_plane 1
```

The resulting 2D map is graphically depicted in Fig. 5d. This is achieved by utilizing the Python `Matplotlib` package; remarkably, `plasmonX_analysis` provides the user with a Python script that can be modified for customization. Fig. 5c shows that at the PRF, the studied dimer sustains a highly localized hot spot in the middle of the gap, correctly reproducing ab initio results for similar systems.

Fig. 5: (a) Graphical depiction of Au₅₆₁ dimer; (b-d) ω FQF μ absorption cross section (b), induced density (c), and enhanced electric field computed at the PRF (2.28 eV) using a polarization field along the x axis.

9. Conclusions and Outlook

In this work, we have presented `plasmonX`, a novel open-source platform for simulating the linear response of plasmonic nanostructures. The code features a unified implementation of both atomistic and continuum electrodynamics models, ω FQ and ω FQF μ and BEM in both DPCM and IEFPCM formalisms. Thanks to its modular architecture, flexible input design, and efficient numerical solvers, which include iterative and memory-efficient “on-the-fly” implementations, `plasmonX` enables accurate and scalable simulations from hundreds to millions of atoms.

Several key developments are planned to further expand the scope and efficiency of `plasmonX`. First, we aim to extend the current frequency-domain formalism to a real-time (RT) framework, allowing the simulation of ultrafast plasmonic phenomena, by implementing our recently developed RT- ω FQ and RT- ω FQF μ methods [114]. Second, we are actively working on generalizing the models to describe condensed phase systems, such as colloids and nanostructures dissolved in an external environment, by implementing the recently developed multiscale framework which integrates ω FQ and ω FQF μ with polarizable embedding [115].

A further avenue involves the development of mixed implicit/explicit strategies, in which part of the nanostructure is described using a full atomistic treatment (for instance, the surface) while the remaining (the core) is treated at the BEM level. We have recently proved that such partitioning drastically reduces the computational cost without reducing the overall accuracy of the ω FQ/ ω FQF μ methods [113]. Finally, to extend the applicability of `plasmonX` to large and extended nanostructures, we plan to incorporate retardation effects, enabling the description of plasmonic phenomena beyond the quasistatic regime. To this end, we aim to implement fast multipole algorithms [116], which would enable linear or near-linear scaling of the long-range electrostatic interactions, making `plasmonX` suitable for simulations involving several millions of atoms.

Data availability

The data supporting this study are provided in the paper. The full simulation data discussed in Sections 7 and 8 are available upon reasonable request.

CRedit authorship contribution statement

Tommaso Giovannini: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Project administration, Methodology, Investigation, Conceptualization; **Pablo Grobas Illobre:** Writing – original draft, Visualization, Validation, Software, Methodology, Investigation; **Piero Lafiosca:** Validation, Software, Methodology; **Luca Nicoli:** Validation, Software, Methodology; **Luca Bonatti:** Validation, Software, Methodology; **Stefano Corni:** Writing – review & editing, Supervision, Methodology, Conceptualization; **Chiara Cappelli:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (Grant Agreement No. 818064). We gratefully acknowledge the Center for High-Performance Computing (CHPC) at SNS for providing the computational

infrastructure. CC and PGI acknowledge funding from MUR-FARE Ricerca in Italia: Framework per l’attrazione ed il rafforzamento delle eccellenze per la Ricerca in Italia - III edizione. Prot. R20YTA2BKZ. TG acknowledges financial support from the University of Rome Tor Vergata “Ricerca Scientifica di Ateneo 2024” INNOVATIONS. SC acknowledges funding from the European Union under the grant ERC-2015-CoG-681285.

Reference

- [1] S.A. Maier, *Plasmonics: fundamentals and applications*, Springer Science & Business Media, 2007.
- [2] M.I. Stockman, Nanoplasmonics: past, present, and glimpse into future, *Opt. Expr* 19 (22) (2011) 22029–22106.
- [3] D.K. Gramotnev, S.I. Bozhevolnyi, Plasmonics beyond the diffraction limit, *Nat. Photon* 4 (2) (2010) 83–91.
- [4] T. Ishida, T. Murayama, A. Taketoshi, M. Haruta, Importance of size and contact structure of gold nanoparticles for the genesis of unique catalytic processes, *Chem. Rev* 120 (2) (2019) 464–525.
- [5] J. Langer, D.J.D. Aberasturi, J. Aizpurua, R.A. Alvarez-Puebla, B. Auguie, J.J. Baumberg, G.C. Bazan, S.E.J. Bell, A. Boisen, A.G. Brolo, J. Choo, D. Cialla-May, V. Deckert, L. Fabris, K. Faulds, F.J. G.D. Abajo, R. Goodacre, D. Graham, A.J. Haes, C.L. Haynes, C. Huck, T. Itoh, M. Käll, J. Kneipp, N.A. Kotov, H. Kuang, E.C.L. Ru, H.K. Lee, J.-F. Li, X.Y. Ling, S.A. Maier, T. Mayerhöfer, M. Moskovits, K. Murakoshi, J.-M. Nam, S. Nie, Y. Ozaki, I. Pastoriza-Santos, J. Perez-Juste, J. Popp, A. Pucci, S. Reich, B. Ren, G.C. Schatz, T. Shegai, S. Schlöcker, L.-L. Tay, K.G. Thomas, Z.-Q. Tian, R.P.V. Duyne, T. Vo-Dinh, Y. Wang, K.A. Willets, C. Xu, H. Xu, Y. Xu, Y.S. Yamamoto, B. Zhao, L.M. Liz-Marzán, 14, 2020. Present and future of surface-enhanced raman scattering.
- [6] L.M. Liz-Marzán, Tailoring surface plasmons through the morphology and assembly of metal nanoparticles, *Langmuir* 22 (1) (2006) 32–41.
- [7] M. Grzelczak, J. Pérez-Juste, P. Mulvaney, L.M. Liz-Marzán, Shape control in gold nanoparticle synthesis 37 (2008) 1783–1791.
- [8] V. Gianini, A.I.F.-D.íguez, S.C. Heck, S.A. Maier, Plasmonic nanoantennas: fundamentals and their use in controlling the radiative properties of nanoemitters, *Chem. Rev* 111 (6) (2011) 3888–3912.
- [9] K.L. Kelly, E. Coronado, L.L. Zhao, G.C. Schatz, The optical properties of metal nanoparticles: the influence of size, shape, and dielectric environment, 2003.
- [10] J.A. Scholl, A.L. Koh, J.A. Dionne, Quantum plasmon resonances of individual metallic nanoparticles, *Nature* 483 (7390) (2012) 421.
- [11] J.A. Scholl, A.G. á Etxarri, A.L. Koh, J.A. Dionne, Observation of quantum tunneling between two plasmonic nanoparticles, *Nano Lett* 13 (2) (2013) 564–569.
- [12] K.J. Savage, M.M. Hawkeye, R. Esteban, A.G. Borisov, J. Aizpurua, J.J. Baumberg, Revealing the quantum regime in tunnelling plasmonics, *Nature* 491 (7425) (2012) 574–577.
- [13] R. Esteban, A.G. Borisov, P. Nordlander, J. Aizpurua, Bridging quantum and classical plasmonics with a quantum-corrected model, *Nat. Commun* 3 (2012) 825.
- [14] R. Esteban, A. Zuggarramurdi, P. Zhang, P. Nordlander, F.J.G. á Vidal, A.G. Borisov, J. Aizpurua, A classical treatment of optical tunneling in plasmonic gaps: extending the quantum corrected model to practical situations, *Faraday Discuss* 178 (2015) 151–183.
- [15] J.J. Baumberg, J. Aizpurua, M.H. Mikkelsen, D.R. Smith, Extreme nanophotonics from ultrathin metallic gaps, *Nat. Mater* 18 (7) (2019) 668–678.
- [16] X. Chen, P. Liu, L. Jensen, Atomistic electrodynamics simulations of plasmonic nanoparticles, *J. Phys. D Appl. Phys* 52 (36) (2019) 363002.
- [17] T.V. Teperik, P. Nordlander, J. Aizpurua, A.G. Borisov, Quantum effects and nonlocality in strongly coupled plasmonic nanowire dimers, *Opt. Express* 21 (22) (2013) 27306–27325.
- [18] W. Zhu, R. Esteban, A.G. Borisov, J.J. Baumberg, P. Nordlander, H.J. Lezec, J. Aizpurua, K.B. Crozier, Quantum mechanical effects in plasmonic structures with subnanometre gaps, *Nat. Commun* 7 (2016) 11495.
- [19] M. Urbietu, M. Barbry, Y. Zhang, P. Koval, D. Sánchez-Portal, N. Zabala, J. Aizpurua, Atomic-scale lightning rod effect in plasmonic picocavities: A classical view to a quantum effect, *ACS Nano* 12 (1) (2018) 585–595.
- [20] D. Marinica, A. Kazansky, P. Nordlander, J. Aizpurua, A.G. Borisov, Quantum plasmonics: nonlinear effects in the field enhancement of a plasmonic nanoparticle dimer, *Nano Lett* 12 (3) (2012) 1333–1339.
- [21] A. Campos, N. Troc, E. Cottancin, M. Pellarin, H.-C. Weissker, J. Lermé, M. Kociak, M. Hillenkamp, Plasmonic quantum size effects in silver nanoparticles are dominated by interfaces and local environments, *Nat. Phys* 15 (3) (2019) 275–280.
- [22] M. Barbry, P. Koval, F. Marchesin, R. Esteban, A. Borisov, J. Aizpurua, D. Sánchez-Portal, Atomistic near-field nanoplasmatics: reaching atomic-scale resolution in nanooptics, *Nano Lett* 15 (5) (2015) 3410–3419.
- [23] J. Zhao, A.O. Pinchuk, J.M. McMahon, S. Li, L.K. Ausman, A.L. Atkinson, G.C. Schatz, Methods for describing the electromagnetic properties of silver and gold nanoparticles, *Acc. Chem. Res* 41 (12) (2008) 1710–1720.
- [24] L. Jensen, C.M. Aikens, G.C. Schatz, Electronic structure methods for studying surface-enhanced raman scattering, *Chem. Soc. Rev* 37 (5) (2008) 1061–1073.
- [25] S.M. Morton, D.W. Silverstein, L. Jensen, Theoretical studies of plasmonics using electronic structure methods, *Chem. Rev* 111 (6) (2011) 3962–3994.
- [26] S. Corni, J. Tomasi, Enhanced response properties of a chromophore physisorbed on a metal particle, *J. Chem. Phys* 114 (8) (2001) 3739–3751.
- [27] E. Ringe, J.M. McMahon, K. Sohn, C. Cobley, Y. Xia, J. Huang, G.C. Schatz, L.D. Marks, R.P.V. Duyne, Unraveling the effects of size, composition, and substrate on the localized surface plasmon resonance frequencies of gold and silver nanocubes: a systematic single-particle approach, *J. Phys. Chem. C* 114 (29) (2010) 12511–12516.
- [28] X. Chen, J.E. Moore, M. Zekarias, L. Jensen, Atomistic electrodynamics simulations of bare and ligand-coated nanoparticles in the quantum size regime, *Nat. Commun* 6 (2015) 8921.

- [29] L.L. Jensen, L. Jensen, Atomistic electrodynamic model for optical properties of silver nanoclusters, *J. Phys. Chem. C* 113 (34) (2009) 15182–15190.
- [30] X. Chen, L. Jensen, Morphology dependent near-field response in atomistic plasmonic nanocavities, 2018.
- [31] T.P. Rossi, A. Zugarumardi, M.J. Puska, R.M. Nieminen, Quantized evolution of the plasmonic response in a stretched nanorod, *Phys. Rev. Lett* 115 (23) (2015) 236804.
- [32] T.P. Rossi, P. Erhart, M. Kuisma, Hot-carrier generation in plasmonic nanoparticles: The importance of atomic structure., *ACS Nano* 14 (8) (2020) 9963–9971.
- [33] M. Kuisma, A. Sakko, T.P. Rossi, A.H. Larsen, J. Enkovaara, L. Lehtovaara, T.T. Rantala, Localized surface plasmon resonance in silver nanoparticles: Atomistic first-principles time-dependent density-functional theory calculations, *Phys. Rev. B* 91 (11) (2015) 115431.
- [34] R.L. Kumawat, G.C. Schatz, Efficient modeling of structural, electronic, and optical properties of silver and gold metal nanoclusters and alloys using optimized scc-dftb parameters, *J. Phys. Chem. C* 129 (2) (2025) 1348–1361.
- [35] S. Agostino, R. Rinaldi, G. Cuniberti, F.D. Sala, Density functional tight binding for quantum plasmonics, *J. Phys. Chem. C* 122 (34) (2018) 19756–19766.
- [36] N.S. Chellam, S. Chaudhuri, A. Ghosal, S.K. Giri, G.C. Schatz, Density functional tight-binding enables tractable studies of quantum plasmonics, 2025.
- [37] Z. Liu, M.B. Oviedo, B.M. Wong, C.M. Aikens, Plasmon-induced excitation energy transfer in silver nanoparticle dimers: A real-time tddft investigation, *J. Chem. Phys* 156 (15) (2022).
- [38] C.J. Herring, M.M. Montemore, Recent advances in real-time time-dependent density functional theory simulations of plasmonic nanostructures and plasmonic photocatalysis, *ACS Nanoscience Au* 3 (4) (2023) 269–279.
- [39] M. Chaudhary, H.-C. Weissker, Optical spectra of silver clusters and nanoparticles from 4 to 923 atoms from the tddft u method, *Nat. Commun* 15 (1) (2024) 9225.
- [40] N. Asadi-Aghbolaghi, R. Ruger, Z. Jamshidi, L. Visscher, Td-dft tb: An efficient and fast approach for quantum plasmonic excitations, *J. Phys. Chem. C* 124 (14) (2020) 7946–7955.
- [41] U. Hohenester, G. Unger, Nanoscale electromagnetism with the boundary element method, *Phys. Rev. B* 105 (7) (2022) 075428.
- [42] C. Mystlidis, X. Zheng, A. Xomalis, G.A. Vandenbosch, A potential-based boundary element implementation for modeling multiple scattering from local and nonlocal plasmonic nanowires, *Adv. Theory Simul* 6 (3) (2023) 2200722.
- [43] C. Ciraci, J.B. Pendry, D.R. Smith, Hydrodynamic model for plasmonics: a macroscopic approach to a microscopic problem, *ChemPhysChem* 14 (6) (2013) 1109–1116.
- [44] C. Ciraci, F.D. Sala, Quantum hydrodynamic theory for plasmonics: Impact of the electron density tail, *Phys. Rev. B* 93 (20) (2016) 205405.
- [45] W. Yan, N.A. Mortensen, M. Wubs, Green's function surface-integral method for nonlocal response of plasmonic nanowires in arbitrary dielectric environments, *Phys. Rev. B* 88 (2013) 155414. <https://link.aps.org/doi/10.1103/PhysRevB.88.155414>. <https://doi.org/10.1103/PhysRevB.88.155414>
- [46] G. Toscano, J. Straubel, A. Kwiatkowski, C. Rockstuhl, F. Evers, H. Xu, N.A. Mortensen, M. Wubs, Resonance shifts and spill-out effects in self-consistent hydrodynamic nanoplasmonics, *Nature Commun* 6 (1) (2015) 1–11.
- [47] C. Mystlidis, X. Zheng, G.A. Vandenbosch, Opensans: a semi-analytical solver for nonlocal plasmonics, *Comput. Phys. Commun* 284 (2023) 108609.
- [48] Y. Eremin, A. Doicu, T. Wriedt, Discrete sources method for modeling of the influence of the non-local effect on the absorption of bimetallic core-shell non-spherical plasmonic nanoparticles, *J. Quantit. Spectrosc. Radiat. Transf* 277 (2022) 107994.
- [49] X. Zheng, C. Mystlidis, A. Xomalis, G.A. Vandenbosch, A boundary integral equation formalism for modeling multiple scattering of light from 3d nanoparticles incorporating nonlocal effects, *Adv. Theory Simul* 5 (12) (2022) 2200485.
- [50] L. Li, S. Lanteri, N.A. Mortensen, M. Wubs, A hybridizable discontinuous galerkin method for solving nonlocal optical response models, *Comput. Phys. Commun* 219 (2017) 99–107.
- [51] M. Moferdt, T. Kiel, T. Sproll, F. Intraivaia, K. Busch, Plasmonic modes in nanowire dimers: a study based on the hydrodynamic drude model including nonlocal and nonlinear effects, *Phys. Rev. B* 97 (7) (2018) 075431.
- [52] L. Huber, U. Hohenester, A computational maxwell solver for nonlocal feibelman parameters in plasmonics, *J. Phys. Chem. C* 129 (5) (2025) 2590–2598.
- [53] B.T. Draine, P.J. Flatau, Discrete-dipole approximation for scattering calculations, *JOSA A* 11 (4) (1994) 1491–1499.
- [54] A. Taflov, S.C. Hagness, M. Picket-May, Computational electromagnetics: the finite-difference time-domain method, Elsevier, Amsterdam, The Netherlands, Amsterdam, The Netherlands, 2005.
- [55] F.J.G. de Abajo, A. Howie, Retarded field calculation of electron energy loss in inhomogeneous dielectrics, *Phys. Rev. B* 65 (11) (2002) 115418.
- [56] V. Myroshnychenko, E. Carbó-Argibay, I. Pastoriza-Santos, J. Pérez-Juste, L.M. Liz-Marzán, F.J.G. de Abajo, Modeling the optical response of highly faceted metal nanoparticles with a fully 3d boundary element method, *Adv. Mater* 20 (22) (2008) 4288–4293.
- [57] B. Mennucci, S. Corni, Multiscale modelling of photoinduced processes in composite systems, *Nat. Rev. Chem* 3 (5) (2019) 315–330.
- [58] L. Bonatti, G. Gil, T. Giovannini, S. Corni, C. Cappelli, Plasmonic resonances of metal nanoparticles: Atomistic vs. continuum approaches, *Front. Chem* 8 (2020) 340.
- [59] J. Marcheselli, D. Chateau, F. Lerouge, P. Baldeck, C. Andraud, S. Parola, S. Baroni, S. Corni, M. Garavelli, I. Rivalta, Simulating plasmon resonances of gold nanoparticles with bipyramidal shapes by boundary element methods, *J. Chem. Theory Comput* 16 (6) (2020) 3807–3815.
- [60] E. Coccia, J. Fregoni, C. Guido, M. Marsili, S. Pipolo, S. Corni, Hybrid theoretical models for molecular nanoplasmonics, *J. Chem. Phys* 153 (20) (2020).
- [61] N.A. Mortensen, Mesoscopic electrodynamic at metal surfaces: -from quantum-corrected hydrodynamics to microscopic surface-response formalism, *Nanophotonics* 10 (10) (2021) 2563–2616.
- [62] P.E. Stamatopoulou, C. Tserkezis, Finite-size and quantum effects in plasmonics: manifestations and theoretical modelling, *Opt. Mater. Expr* 12 (5) (2022) 1869–1893.
- [63] J.M. Fitzgerald, P. Narang, R.V. Craster, S.A. Maier, V. Giannini, Quantum plasmonics, *Proceedings of the IEEE*, 104, 2016.
- [64] J.J. Baumberg, Picocavities: a primer, *Nano Lett* 22 (14) (2022) 5859–5865.
- [65] F. Benz, M.K. Schmidt, A. Dreismann, R. Chikkaraddy, Y. Zhang, A. Demetriadou, C. Carnegie, H. Ohadi, B.D. Nijs, R. Esteban, J. Aizpurua, J.J. Baumberg, Single-molecule optomechanics in "picocavities, *Science* 354 (doi:10.1126/science.aah5243) (2016) 726–729. <https://www.science.org/doi/abs/10.1126/science.aah5243>. <https://doi.org/10.1126/science.aah5243>
- [66] T. Giovannini, L. Nicoli, S. Corni, C. Cappelli, The electric field morphology of plasmonic picocavities, *Nano Letters* 25 (27) (2025) 10802–10808.
- [67] T. Giovannini, M. Rosa, S. Corni, C. Cappelli, A classical picture of subnanometer junctions: an atomistic drude approach to nanoplasmonics, *Nanoscale* 11 (13) (2019) 6004–6015.
- [68] T. Giovannini, L. Bonatti, M. Polini, C. Cappelli, Graphene plasmonics: Fully atomistic approach for realistic structures, *J. Phys. Chem. Lett* 11 (18) (2020) 7595–7602.
- [69] T. Giovannini, L. Bonatti, P. Lafosca, L. Nicoli, M. Castagnola, P.G. Illobre, S. Corni, C. Cappelli, Do we really need quantum mechanics to describe plasmonic properties of metal nanostructures?, *ACS Photonics* 9 (2022) 3025–3034.
- [70] P. Lafosca, T. Giovannini, M. Benzi, C. Cappelli, Going beyond the limits of classical atomistic modeling of plasmonic nanostructures, *J. Phys. Chem. C* 125 (43) (2021) 23848–23863.
- [71] J. Tomasi, B. Mennucci, R. Cammi, Quantum mechanical continuum solvation models, *Chem. Rev* 105 (8) (2005) 2999–3094.
- [72] J. Tomasi, B. Mennucci, E. Cancès, The IEF version of the PCM solvation method: an overview of a new method addressed to study molecular solutes at the QM ab initio level, *J. Mol. Struct.: THEOCHEM* 464 (1) (1999) 211–226.
- [73] B. Mennucci, E. Cancès, J. Tomasi, Evaluation of solvent effects in isotropic and anisotropic dielectrics and in ionic solutions with a unified integral equation method: Theoretical bases, computational implementation, and numerical applications, *J. Phys. Chem. B* 101 (49) (1997) 10506–10517.
- [74] E. Cancès, B. Mennucci, J. Tomasi, A new integral equation formalism for the polarizable continuum model: Theoretical background and applications to isotropic and anisotropic dielectrics, *J. Chem. Phys* 107 (8) (1997) 3032–3041.
- [75] U. Hohenester, A. Trügler, Mnpbem - a matlab toolbox for the simulation of plasmonic nanoparticles, *Comput. Phys. Commun* 183 (2) (2012) 370–381. <https://www.sciencedirect.com/science/article/pii/S0010465111003274>.
- [76] Y. Saad, Iterative methods for sparse linear systems, 2003.
- [77] G. Meurant, J.D. Tebbens, Krylov Methods for Nonsymmetric Linear Systems: From Theory to Computations, Springer, 2020.
- [78] Y. Saad, M.H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comp* 7 (3) (1986) 856–869.
- [79] L. Bonatti, L. Nicoli, T. Giovannini, C. Cappelli, In silico design of graphene plasmonic hot-spots, *Nanoscale Adv* 4 (10) (2022) 2294–2302.
- [80] A. Mayer, Formulation in terms of normalized propagators of a charge-dipole model enabling the calculation of the polarization properties of fullerenes and carbon nanotubes, *Phys. Rev. B* 75 (4) (2007) 045407.
- [81] M. Pelton, G.W. Bryant, Introduction to metal-nanoparticle plasmonics, 5, John Wiley & Sons, 2013.
- [82] A.H. Castro-Neto, F. Guinea, N.M. Peres, K.S. Novoselov, A.K. Geim, The electronic properties of graphene, *Rev. Mod. Phys* 81 (1) (2009) 109.
- [83] Z. Fang, S. Thongrattanasiri, A. Schlather, Z. Liu, L. Ma, Y. Wang, P.M. Ajayan, P. Nordlander, N.J. Halas, J. F. García de Abajo, Gated tunability and hybridization of localized plasmons in nanostructured graphene, *ACS Nano* 7 (3) (2013) 2388–2395.
- [84] A. Liebsch, Surface-plasmon dispersion and size dependence of mie resonance: silver versus simple metals, *Phys. Rev. B* 48 (15) (1993) 11317.
- [85] T. Giovannini, A. Puglisi, M. Ambrosetti, C. Cappelli, Polarizable qm/mm approach with fluctuating charges and fluctuating dipoles: the qm/qf_μ model, *J. Chem. Theory Comput* 15 (4) (2019) 2233–2245.
- [86] L. Nicoli, P. Lafosca, P.G. Illobre, L. Bonatti, T. Giovannini, C. Cappelli, Fully atomistic modeling of plasmonic bimetallic nanoparticles: nanoalloys and core-shell systems, *Front. Photonics* 4 (2023) 1199598.
- [87] A. Trügler, Optical properties of metallic nanoparticles, Springer, 2011.
- [88] S. Miertuš, E. Scrocco, J. Tomasi, Electrostatic interaction of a solute with a continuum. a direct utilization of ab initio molecular potentials for the prevision of solvent effects, *Chem. Phys* 55 (1) (1981) 117–129.
- [89] R. Cammi, J. Tomasi, Remarks on the use of the apparent surface charges (asc) methods in solvation problems: Iterative versus matrix-inversion procedures and the renormalization of the apparent charges, *J. Comput. Chem* 16 (12) (1995) 1449–1458.
- [90] E.O. Purisima, S.H. Nilar, A simple yet accurate boundary element method for continuum dielectric calculations, *J. Comput. Chem* 16 (6) (1995) 681–689. <https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.540160604>. <https://doi.org/10.1002/jcc.540160604>
- [91] GitHub Repository, 2025. <https://github.com/plasmonX/plasmonX>.
- [92] T. Giovannini, P.G. Illobre, P. Lafosca, L. Nicoli, L. Bonatti, S. Corni, C. Cappelli, plasmonx: an open-source code for nanoplasmonics, 2026. <https://doi.org/10.5281/zenodo.18151754>
- [93] C. Kitware, Cross-Platform Make, 2023. <https://cmake.org>.
- [94] C. Kitware, CMake Testing Tool, 2023. <https://cmake.org/cmake/help/latest/manual/ctest.1.html>.
- [95] Windows Subsystem for Linux Documentation, Microsoft, 2023. <https://learn.microsoft.com/en-us/windows/wsl/>.
- [96] Intel oneAPI Math Kernel Library (MKL), Technical Report, Intel Corporation, 2023. <https://www.intel.com/content/www/us/en/developer/tools/oneapi/oneapi.html>.
- [97] plasmonx manual, 2025. GitHub Repository, <https://plasmonx.readthedocs.io>.
- [98] P.G. Illobre, Geom: Nanostructure geometry generator, 2025. <https://github.com/pgrobasillobre/geom>.
- [99] E.E.Q. Rodriguez, P. Lafosca, T. Giovannini, C. Cappelli, Quantum dynamics of dissipative polarizable media, *Phys. Rev. B* 111 (23) (2025) 235144.
- [100] F. Lipparini, L. Lagardere, B. Stamm, E. Cancès, M. Schieders, P. Ren, Y. Maday, J.-P. Piquemal, Scalable evaluation of polarization energy and associated forces in polarizable molecular dynamics: I. toward massively parallel direct space computations, *J. Chem. Theory*

- Comput 10 (4) (2014) 1638–1651.
- [101] M. Bugeanu, R.D. Remigio, K. Mozgawa, S.S. Reine, H. Harbrecht, L. Frediani, Wavelet formulation of the polarizable continuum model. ii. use of piecewise bilinear boundary elements, *Phys. Chem. Chem. Phys.* 17 (47) (2015) 31566–31581.
- [102] U. Hohenester, Making simulations with the mnpbem toolbox big: Hierarchical matrices and iterative solvers, *Comput. Phys. Commun.* 222 (2018) 209–228.
- [103] G. Hager, G. Wellein, *Introduction to High Performance Computing for Scientists and Engineers*, CRC Press, 2010.
- [104] S. Link, Z.L. Wang, M.A. El-Sayed, Alloy formation of gold-silver nanoparticles and the dependence of the plasmon absorption on their composition, *J. Phys. Chem. B* 103 (18) (1999) 3529–3533.
- [105] D. Rioux, S. Vallières, S. Besner, P.M. noz, E. Mazur, M. Meunier, An analytic model for the dielectric function of au, ag, and their alloys, *Adv. Opt. Mater.* 2 (2) (2014) 176–182.
- [106] G.C. Papavassiliou, Surface plasmons in small au-ag alloy particles, *J. Phys. F: Met. Phys* 6 (4) (1976) L103.
- [107] S. Ristig, O. Prymak, K. Loza, M. Gocyla, W. Meyer-Zaika, M. Heggen, D. Raabe, M. Epple, Nanostructure of wet-chemically prepared, polymer-stabilized silver-gold nanoalloys (6 nm) over the entire composition range, *J. Mater. Chem. B* 3 (23) (2015) 4654–4662.
- [108] L.K.S. rensen, A.D. Utyushev, V.I. Zakomirnyi, H.A. gren, Atomistic description of plasmonic generation in alloys and core shell nanoparticles, *Phys. Chem. Chem. Phys.* 23 (1) (2021) 173–185.
- [109] H. Duan, A.I. F.-D. ríguez, M. Bosman, S.A. Maier, J.K. Yang, Nanoplasmonics: classical down to the nanometer scale, *Nano Lett* 12 (3) (2012) 1683–1689.
- [110] W. Humphrey, A. Dalke, K. Schulten, VMD - Visual Molecular Dynamics, *Journal of Molecular Graphics* 14 (1996) 33–38.
- [111] P. Lafiosca, L. Nicoli, L. Bonatti, T. Giovannini, S. Corni, C. Cappelli, Qm/classical modeling of surface enhanced raman scattering based on atomistic electromagnetic models, *J. Chem. Theory Comput* 19 (12) (2023) 3616–3633.
- [112] T. Giovannini, S. Gómez, C. Cappelli, Modeling raman spectra in complex environments: From solutions to surface-enhanced raman scattering, *J. Phys. Chem. Lett* 16 (12) (2025) 3106–3121.
- [113] P.G. Illobre, P. Lafiosca, L. Bonatti, T. Giovannini, C. Cappelli, Mixed atomistic-implicit quantum/classical approach to molecular nanoplasmonics, *J. Chem. Phys.* 162 (4) (2025).
- [114] P. Lafiosca, L. Nicoli, S. Pipolo, S. Corni, T. Giovannini, C. Cappelli, Real-time formulation of atomistic electromagnetic models for plasmonics, *J. Phys. Chem. C* 128 (41) (2024) 17513–17525.
- [115] L. Nicoli, S. Sodomaco, P. Lafiosca, T. Giovannini, C. Cappelli, Atomistic multiscale modeling of colloidal plasmonic nanoparticles, *ACS Phys. Chem. Au* 4 (6) (2024) 669–678.
- [116] E. Darve, The fast multipole method: numerical implementation, *J. Comput. Phys.* 160 (1) (2000) 195–240.