Fitting experimental dispersion data with a simulated annealing method for nano-optics applications

J Viquerat

Fitting experimental dispersion data with a simulated annealing method for nano-optics applications

J. Viquerat

23rd February 2018

Abstract

A considerable amount of materials in nanophotonics are dispersive, enabling the propagation of so called surface plasmons at their interfaces with dielectrics. Hence, a reliable fit of frequency-dependent permittivity functions with an appropriate model is a first-order necessity for the accurate design of nano-optics devices with time-domain numerical methods, such as finite-difference time-domain (FDTD) or discontinuous Galerkin time-domain (DGTD). In this paper, we present the necessary ingredients to fit experimental permittivity functions using the simulated annealing (SA) method with a generalized second-order dispersion model implemented in the DIOGENES software suite (see `https://diogenes.inria.fr/`). By scanning through different classes of materials, we come up with effective rules of thumb to make the fitting process fast and accurate.

1 Introduction

Over the course of its development, the nanophotonics field has seen a tremendous amount of devices designed for applications ranging from biosensors [CLS⁺11] to nanostructured photovoltaic cells [AP10]. A vast majority of these take advantage of the combination of a specific geometry with selected materials to obtain a desired functionality. Through the years, the design process of these devices has started relying more and more on numerical methods, either in frequency or in time-domain [GBM15]. Although it is relatively easy to take dispersive materials into account with frequency-domain methods, with time-domain methods it is necessary to fit experimental data with an adapted frequency-dependent function that can be transformed back in the temporal domain by inverse Fourier transform [Viq15], and discretized with the chosen numerical method.

In section 2, we shortly review existing dispersion models and dispersive materials, and motivate our choice of fitting algorithm. Then, in section 3, the SA technique is outlined. Section 4 represents the core contribution of this paper. We first detail the modifications introduced to improve our fitting results as well as the settings used for the algorithm, before conducting a large panel of fitting experiments on a variety of dispersive materials. Rules of thumb for accurate and efficient material fittings with the SA method are deduced. Eventually, we test the obtained fits in the context of the DIOGENES DGTD solver.

2 Dispersive materials and dispersion models

The development of dispersion models for metals originates in the work of Drude [Dru00], who devised the so-called Drude model based on the kinetic theory of gases:

$$\varepsilon_{r,\mathrm{Drude}}(\omega) = \varepsilon_\infty - \frac{\omega_d^2}{\omega^2 + i\omega\gamma_d}.$$

Although this model presents a good adequation with experimental data of noble metals for low visible frequencies, it does not account for interband transitions that arise for certains metals in the high visible frequency range. For this reason, it is common to add one or more Lorentz poles to the Drude model, thus yielding the following form:

$$\varepsilon_{r,\text{Drude-Lorentz}}(\omega) = \varepsilon_\infty - \frac{\omega_d^2}{\omega^2 + i\omega\gamma_d} - \frac{\Delta\varepsilon\omega_l^2}{\omega^2 - \omega_l^2 + i\omega\gamma_l}.$$

This method shows a good agreement with experimental data for noble metals, and has been widely implemented, often with four or more Lorentz poles [HN07]. However, this model fails to properly fit materials such as transition or post-transition metals, because of the electronic correlation existing in such materials, introducing a retardation effect [WROB13]. During the last decade, advanced dispersion models were designed to tackle dispersive materials more efficiently. Among them, the Critical Points (CP) model [VLDC11] and the Complex-Conjugate Pole-Residue Pairs (CCPRP) model [HDF06] have shown great improvements over the standard Drude-Lorentz model in fitting metals and compounds on broad frequency ranges with a limited amount of poles. As briefly sketched in [VLDC11], these two models are mathematically equivalent at the continuous level. While both of these models use coupled first-order poles with complex coefficients, in [Viq15] we introduced a generalized model exploiting real coefficients only:

$$\varepsilon_r(\omega) = \varepsilon_\infty - \sum_{l=1}^{n_p} \frac{c_l - i\omega d_l}{\omega^2 - e_l + i\omega f_l}, \tag{1}$$

where n_p is the number of second-order poles. It is easy to show that this model is also equivalent, in the continuous domain, to CP and CCPRP models. In [LSV17], we showed that model (1) is causal if $e_l \geq 0$ and $f_l \geq 0$. Additionally, it is easy to obtain a permittivity function with an imaginary part identically equal to zero by imposing $d_l \equiv$ and $f_l \equiv 0$.

In figure 1, we regroup a representative sample of dispersive materials regularly used in nano-optics devices, including noble, transition and post-transition metals, but also semiconductors, pure or compounded with metals or other semiconductors. All the experimental data comes from [Pol]. Although these different materials show very different behaviors, we will show that a permittivity function such as (1) allows to fit all of them with sufficient accuracy.

In order to fit the free parameters of (1) to experimental data, various techniques can be used. Apart from the well-known least square method, vector fitting techniques (see [GS99]) are also well developed for the CCPRP formulation. In [SLM17], the authors present a mixed method for fitting CP parameters, using an analytic minimization along with a gradient descent method. For an increasing number of poles, one can be left with a large optimization problem presenting many local minima, likely to trap optimization algorithms away from the desired global minimum. Simulated Annealing (SA) methods have proved to be particularly efficient in finding global minima in these situations, even when the initial guess is far from the optimal solution ([KGV83]). In the following section, we shortly present the SA technique and its possibilities.

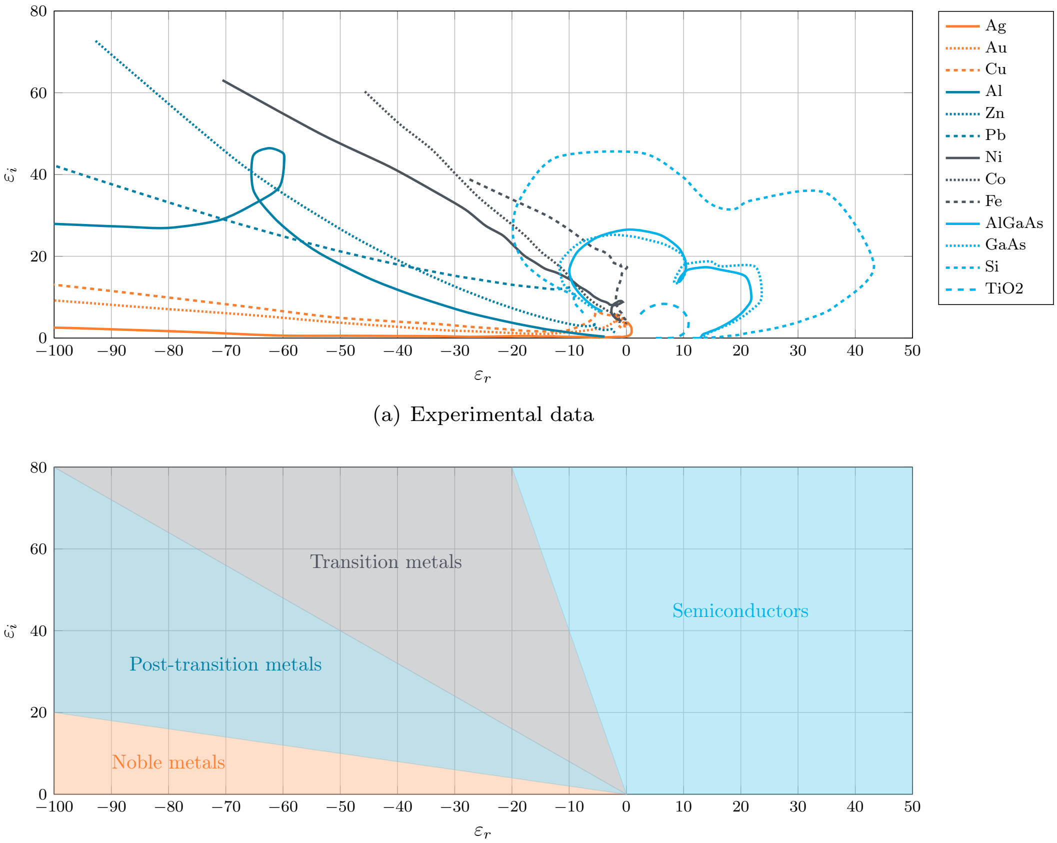
3 The simulated annealing algorithm

The simulated annealing (SA) algorithm was initially proposed by Kirkpatrick *et al* [KGV83], and is recognized to be a suitable optimization method in the context of a cost function with multiple local minima. Although it was initially designed to solve combinatorial problems (such as the well-known traveling salesman problem), it can also be exploited to find global minima of continuous functions [Loc02]. Given a cost function f defined on a continuous domain X, the goal of an optimization algorithm will be to find $x^* \in X$ such as:

$$f^* \equiv f(x^*) = \min_{x \in X} f(x).$$

The name of the SA algorithm comes from a metallurgic process, in which the controlled cooling of a molten metal allows a final state of lower energy than that obtained if the cooling process is left unconstrained. As a consequence, one of the main variable of the algorithm is the *temperature* T, which will control the probability to move from one state to another. In essence, the SA algorithm works as follows:

Additionally, we define the following quantities:



(a) Experimental data

(b) Rough classification

Figure 1 | Incomplete classification of dispersive materials based on their $\varepsilon_i = f(\varepsilon_r)$ behavior. All data comes from [Pol] and is selected in the angular frequency range $[1, 10]$ PHz when available.

Algorithm 1 Simulated annealing

```
Set an initial temperature $T \leftarrow T_0$ and choose a final temperature $T_f < T_0$
Generate a random initial solution $x \leftarrow x_0$
Compute its cost $f(x_0)$

while $T > T_f$ do
    for $v \leftarrow 1, n_v$ do                                ▷ Loop on a given number of neighbor states
        Compute a neighbor solution $x_n = n(x)$
        Compute its cost $f(x_n)$

        if $f(x_n) < f(x)$ then
            $x \leftarrow x_n$ and $f(x) \leftarrow f(x_n)$                       ▷ Accept better solution
        else
            Compute normal random number $r$
            Compute acceptance $a(f(x), f(x_n), T)$

            if $a(f(x), f(x_n), T) > r$ then                  ▷ Possibly accept poorer solution
                $x \leftarrow x_n$ and $f(x) \leftarrow f(x_n)$
            end if
        end if
    end for

    $T \leftarrow c(T)$                                                    ▷ Decrease temperature
end while
```

- N_M the total number of moves at each temperature step;
- N_A the total number of accepted moves at each temperature step;
- N_{AU} the total number of accepted uphill moves at each temperature step;
- N_D the total number of denied moves at each temperature step;
- N_{CM} the total number of cumulated moves.

As can be seen, various parameters can be tuned in the SA algorithm, namely:

- The initial and final temperature, respectively T_0 and T_f;
- The number of visited neighbors at each temperature n_v;
- The initial guess x_0;
- The acceptance function a;
- The neighbor function n;
- The cooling function c.

Since the SA algorithm is a well-established method, there is a vast literature on the topic of making wise choices for these parameters and functions. For this reason, we will not expand too long on all the existing possibilities (the reader is referred to the references that follow, and the references therein).

- The choice of the initial and final temperatures depend on the considered problem. A typical choice for T_0 is usually 1.0, but we shall see later that in some cases it is interesting to start from higher values. The choice of the final temperature may seem a little arbitrary, and it is possible to replace it with a stopping criterion based on the amount of recently-accepted better solutions at current temperature;
- Similarly, the number of visited neighbors at each temperature is an arbitrary value. However, it is reasonable to say that the algorithm will perform better if it is allowed to sample a larger amount of states at each temperature, at expense of the computational cost;

- The acceptance function, although usually simple, represents the core part of the algorithm: its role is to determine if an *uphill* solution (*i.e.* a neighbor solution which cost is higher than current solution) must be accepted or not. The most spread choice of acceptance function is the Metropolis formula [HJJ03] :

  $$a(f(x), f(x_n), T) = \min\left\{1, \exp\left(\frac{f(x) - f(x_n)}{T}\right)\right\}. \tag{2}$$

  If the neighbor solution is better, then obviously $a = 1$. On the other hand, if $f(x_n) > f(x)$, then $0 < a < 1$, and this solution will be randomly accepted. Given that T is a decreasing function of the number of iterations of the algorithm, the probability of an uphill solution being accepted will decrease as the algorithm converges toward a minimum (either global if it has succeeded, or local if it failed to escape from it);
- The goal of the neighbor function is to build a neighbor state x_n from current state x. In discrete problems, this is often simply done by randomly swapping two elements of the solution (in the traveling salesman's problem, two cities are randomly picked and swapped in the ordered list). In continuous problems, however, building neighbors introduce an additional source of freedom for the user, since the neighborhood of a state must be defined. In practice, the loop on the neighbor states can be embedded in a loop over the total number of parameters to optimize. Then, a neighbor state is obtained by randomly moving the current free parameter in a defined range. The size of the neighborhood need not to be fixed throughout the optimization process, several refinements being available in the literature [Xin11]. In the following, we denote by Δ_n the normalized size of the neighborhood of a state. $\Delta_n = 1$ indicates that the whole parameter space can be reached from any state in it, while $\Delta_n = 0.5$ indicates that the neighborhood of a given state is limited to half the size of the parameter space around it;
- The choice of the cooling function is the topic of an abundant literature. One of the most common choice might be the exponential schedule [NA98]:

  $$T_{n+1} = \alpha T_n,$$

  where $0 < \alpha < 1$ is a constant factor. Typical values range from 0.8 to 0.99, directly impacting the performance and the computational cost of the algorithm.

The variety of choices for each algorithm parameter must be adapted to the considered problem. In the next section, we will focus on the problem of fitting experimental permittivity data to existing models containing up to a few tens of free parameters.

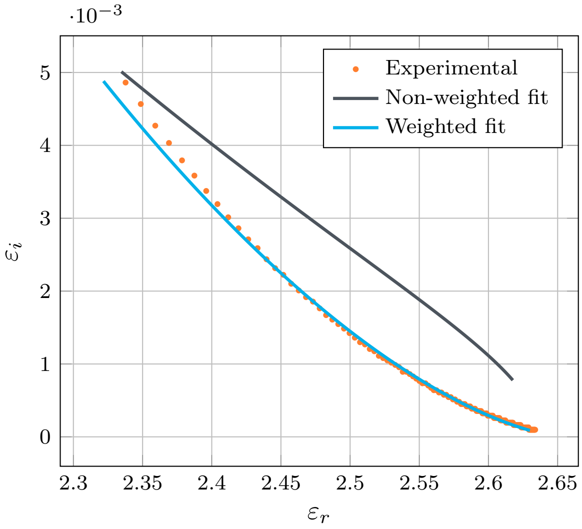


Figure 2 | Fit of Al2O3 with 1 second-order pole using non-weighted and weighted cost functions.

4 Fitting dispersive materials

4.1 Cost function

For the following fitting experiments, we define the cost function:

$$f(x) = \sum_{j=1}^{N_s} \left[\frac{\left|\varepsilon^r_{\text{exp}}(\omega_j) - \varepsilon^r_{\text{fit}}(\omega_j, x)\right|}{\Delta\varepsilon^r_{\text{exp}}} + \frac{\left|\varepsilon^i_{\text{exp}}(\omega_j) - \varepsilon^i_{\text{fit}}(\omega_j, x)\right|}{\Delta\varepsilon^i_{\text{exp}}} \right], \tag{3}$$

where N_s is the total number of samples in the experimental data (assumed to be much larger than the total number of free parameters of the model), the $\cdot^r$ and $\cdot^i$ exponents respectively represent the real and imaginary parts, while $\Delta\varepsilon^r_{\text{exp}}$ and $\Delta\varepsilon^i_{\text{exp}}$ respectively correspond to the range of values of the real and imaginary parts of the experimental permittivity.

By weighting the distance between the fit and the experimental data by $\frac{1}{\Delta\varepsilon^r_{\text{exp}}}$ and $\frac{1}{\Delta\varepsilon^i_{\text{exp}}}$, we ensure that the algorithm grants a similar importance to the real and the imaginary parts of the permittivity function. Indeed, in the case where the figure of merit $\frac{\varepsilon^i}{\varepsilon^r}$ is globally low or high over the frequency range, the absolute distance $\left|\varepsilon^r_{\text{exp}}(\omega_j) - \varepsilon^r_{\text{fit}}(\omega_j)\right| + \left|\varepsilon^i_{\text{exp}}(\omega_j) - \varepsilon^i_{\text{fit}}(\omega_j)\right|$ may not be representative of the overall quality of the fit. To illustrate this, we compare in figure 2 the fits obtained with 1 second-order pole for Al2O3, with and without such a weighting. Obviously, the weighting significantly improves the quality of the fit, providing an acceptable solution with a single pole.

4.2 Parameters range

A typical range of angular frequencies in plasmonics is $\omega \in [1, 10]$ PHz. Hence, to reduce the parameter space to be scanned by the algorithm, we work with a normalized angular frequency $\tilde{\omega} = \omega \times 10^{-15}$. Given the form of (1), one would expect c_l and e_l to be roughly in $[0, 100]$ PHz^2, while d_l and f_l would be in $[0, 10]$ PHz. Hence, we rewrite (1) as:

$$\varepsilon_r(\omega) = \varepsilon_\infty - \sum_{l=1}^{n_p} \frac{\tilde{c_l}^2 - i\tilde{\omega}\tilde{d_l}}{\tilde{\omega}^2 - \tilde{e_l}^2 + i\tilde{\omega}\tilde{f_l}}, \tag{4}$$

with the following relations:

$$\begin{aligned}
\tilde{\omega} &= 10^{-15} \times \omega, \\
\tilde{c_l} &= 10^{-15} \times \sqrt{c_l}, \\
\tilde{d_l} &= 10^{-15} \times d_l, \\
\tilde{e_l} &= 10^{-15} \times \sqrt{e_l}, \\
\tilde{f_l} &= 10^{-15} \times f_l.
\end{aligned}$$

By doing so, we scale all parameters to the same range of values, thus restricting the parameter space for the SA algorithm to the form $[1, \varepsilon_{\max}] \times [0, p_{\max}]^{4n_p}$. In the remaining of this paper, we set $\varepsilon_{\max} = 10$ and $p_{\max} = 10$.

4.3 Incorporating constraints to the cost function

In the context of this study, we wish to incorporate parameter constraints to the cost function. In order to respect the parameter range determined above, we use a simple penalization of the form:

$$f_\rho(x) = f(x) + \rho \max\left(0, -x + u, x - v\right). \tag{5}$$

where

$$u = {}^t\left(1, 0, ..., 0\right), v = {}^t\left(\varepsilon_{\max}, p_{\max}, ..., p_{\max}\right),$$

and ρ is a well-chosen penalization parameter. For the current problem, we observed that $\rho = 1000$ or higher provided satisfying results. In practice, this penalization prevents ε_∞ to be lower than 1, and the parameters of (4) to be negative [GMW81].

4.4 Sign of $\varepsilon^i_{\text{fit}}$

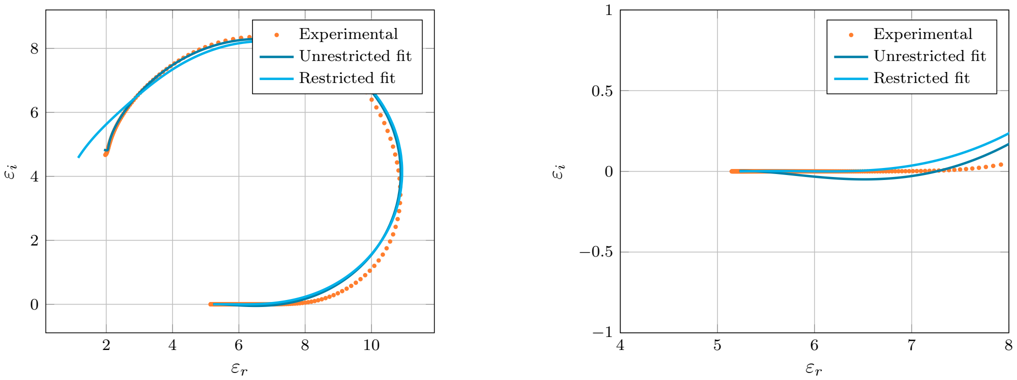
As an additional refinement, we modify the cost function in order to discriminate negative imaginary parts. To do so, at each evaluation of $f(x)$, we evaluate if $\varepsilon^i_{\text{fit}}(\omega) < 0$ for all ω in the range of interest. If so, the cost function is increased in the fashion of previous section so as to rule out current state x. This is particularly important in cases where $\varepsilon^i_{\text{fit}}$ is close to zero on a wide frequency range. As an example, we compare in figure 3 fits of TiO2 with and without this restriction. Although the unrestricted fit provides a better approximation of the experimental data, the negative imaginary part of the permittivity would yield an unstable behavior in any time-domain numerical method. As can be seen, the restricted fit respects this condition by decreasing more slowly to zero.

4.5 Non-absorbing materials

In some cases, experimental data can be found where the imaginary part is identically zero. As we shall see, it is enough to set a lower and an upper bound equal to zero for all d_l and f_l parameters, thus reducing the model to:

$$\varepsilon_r(\omega) = \varepsilon_\infty - \sum_{l=1}^{n_p} \frac{\tilde{c_l}^2}{\tilde{\omega}^2 - \tilde{e_l}^2}.$$

Here, one easily recognizes the Sellmeier's law [Goo11]. Although it does not perform as well as model (4), it can be very useful in cases where even a very low imaginary part is not acceptable. As an example, figure 4 shows fits obtained with and without this feature for Si3N4 data where the imaginary part is identically zero.



(a) Entire fit

(b) Zoom

Figure 3 | Fit of TiO2 with 3 second-order poles with and without restriction on the sign of $\varepsilon^i_{\mathbf{fit}}$.

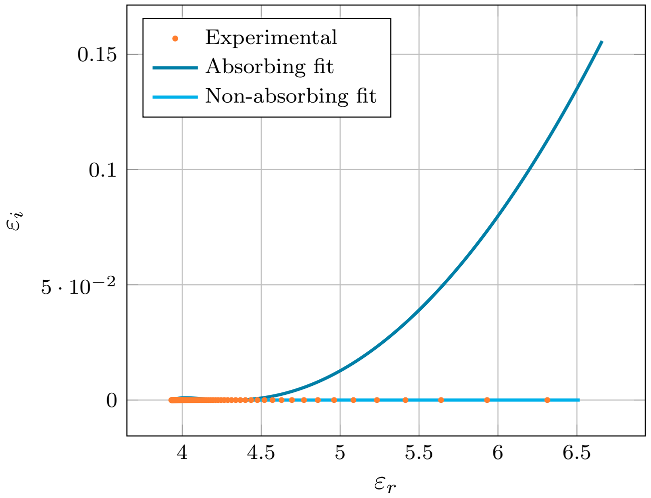


Figure 4 | Fit of Si3N4 with 2 second-order poles using absorbing and non-absorbing fits.

4.6 Initial guess and stopping criterion

After performing a vast amount of fittings on various kinds of materials, we observed that in most cases the initial guess x_0 only had a minor influence on the final result. For this reason, in the remaining of this paper, we set $x_0 \equiv 0$. Regarding the influence of the final temperature T_f, we replace it with a stopping criterion: the algorithm exits when no move has been accepted during the last temperature step. Unless stated otherwise, we set the size of the neighborhood to be the full parameter space, and the acceptance function equal to (2). Finally, the number of visited neighbors at each temperature step is chosen equal to $1000 \times (4n_p + 1)$, *i.e.* the free parameters are moved one by one, 1000 trials being made for each of them.

4.7 Noble metals

In this section, we assess the influence of the initial temperature T_0 and the decreasing temperature factor α on the obtained f^* for noble metals (gold, silver and copper). Results are presented in figure 5. For slow cooling (typically, $\alpha = 0.99$), one observes that the best cost is remarkably stable with respect to T_0, with a slight minimum between $T_0 = 0.1$ and $T_0 = 1$. As α is decreased, the behavior becomes more erratic, and no clear trend can be deduced. This effect is called *quenching*, and is known to cause suboptimal results with the SA technique [KGV83]. In figure 6, we illustrate the difference in behavior for the algorithm between a slow cooling schedule ($\alpha = 0.99$), and a quenching situation ($\alpha = 0.8$) by plotting f^* both as a function of T and N_{CM} processed by the algorithm, with $T_0 = 1000$. As can be seen, with a low value of α the optimization process gets stuck in a local minimum and fails to escape from it, although the stopping criterion is met at a much lower temperature than for a high value of α. This is however less visible for gold and copper than for silver. From these first experiments, we select the couple $(\alpha = 0.99, T_0 = 0.1)$ as a good starting candidate for future fits.

Some additional comments on the algorithm insights can be made, especially on the acceptance function and the choice of the neighborhood size. On figure 8(a), we plot N_A, N_{AU} and N_D as functions of T for gold experimental data. As can be seen, roughly half of the accepted moves are uphill ones, underlining the local-minima-escaping feature of the SA algorithm. Broadly speaking, it is usually recommended to have between 50 and 60% of uphill accepted moves to obtain satisfying results. The best fits obtained with four poles for noble metals are shown in figure 7.

4.8 Acceptance function

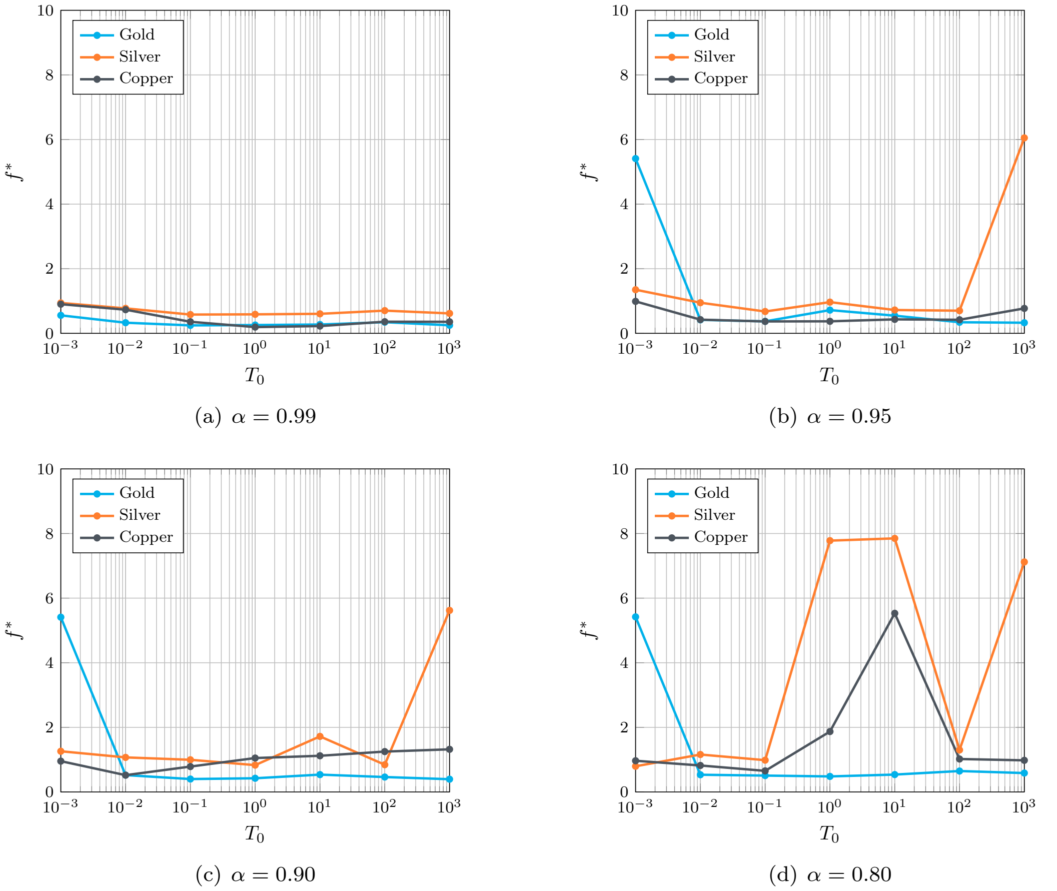
Regarding the acceptance function, the Metropolis formula (2) is not the only available choice. Another usual option is the Barker function [Sch97]:

$$a(f(x), f(x_n), T) = \frac{1}{1 + \exp\left(-\frac{f(x)-f(x_n)}{T}\right)}. \tag{6}$$

With this criterion, some downhill steps might be rejected if they do not substantially improve the solution. As T gets lower, downhill moves are more likely accepted, while uphill moves get more and more rejected. This is illustrated in figure 8(b), where we can also see that the 50% rule of thumb is respected. To test this acceptance function further, we reproduce the graph of figure 5 for $\alpha = 0.99$ (computed with Metropolis function) with the Barker function (results are displayed on figure 9). Although we observe some slightly better results for $T_0 = 0.01$, using the Barker formula as acceptance function brings no major improvement in the final result of the algorithm, and we shall use the Metropolis function in the remaining of this paper.

4.9 Neighborhood size

Regarding the size of the neighborhood, no clear strategy seems to emerge from the literature [HJJ03]. In figure 10, we assess the impact of the normalized neighborhood size Δ_n on f^* for noble metals, with $\alpha = 0.99$ and $T_0 = 0.1$. Again, it is hard to draw a clear trend from these results, although we observe no significant variations in the final fit quality over the Δ_n range. For that reason, we will use $\Delta_n = 1$ in the remaining of the paper.



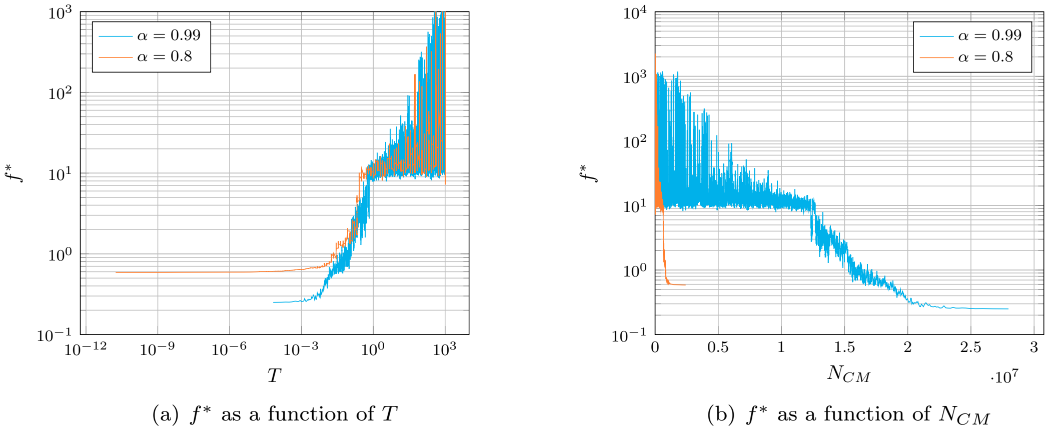
(a) $\alpha = 0.99$

(b) $\alpha = 0.95$

(c) $\alpha = 0.90$

(d) $\alpha = 0.80$

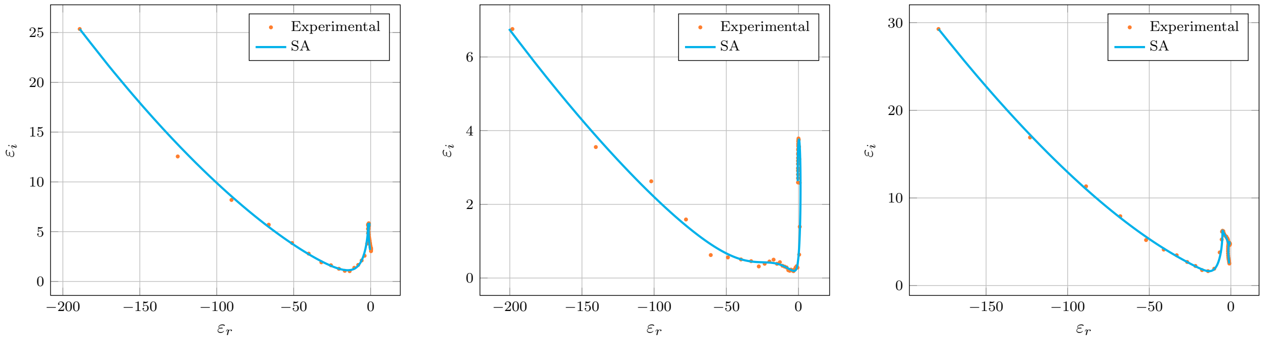
Figure 5 | Influence of T_0 on f^* for noble metals with $n_p = 4$, $x_0 = 0$ and variable values of α. For values α close to 1, we observe a remarkable stability of f^* with respect to T_0. When α decreases, the behavior is wilder, and no clear trend can be deduced.



(a) f^* as a function of T

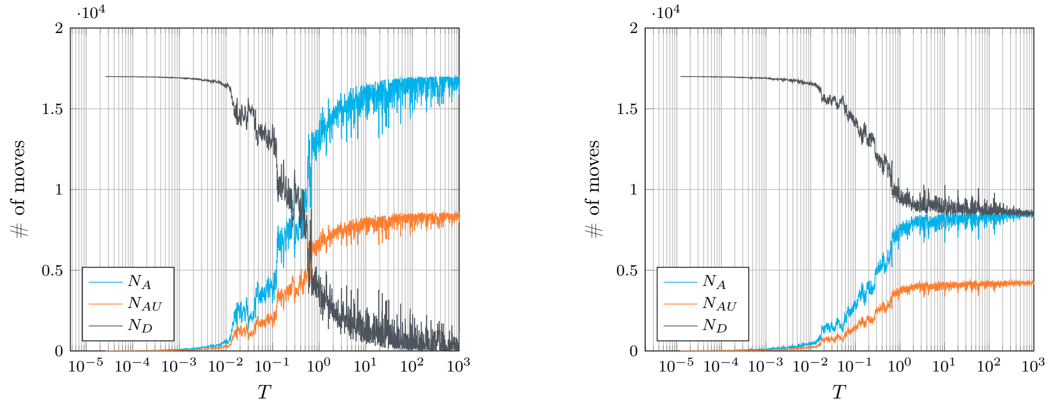
(b) f^* as a function of N_{CM}

Figure 6 | f^* as a function of the temperature for gold, with $\alpha = 0.99$ and $\alpha = 0.8$. On the left figure, we observe the effect of quenching, leading to a poor local minimum despite a very low final temperature. On the right, we see that, because of the fast cooling, very few states are investigated for $\alpha = 0.8$, leading to the poor solution.



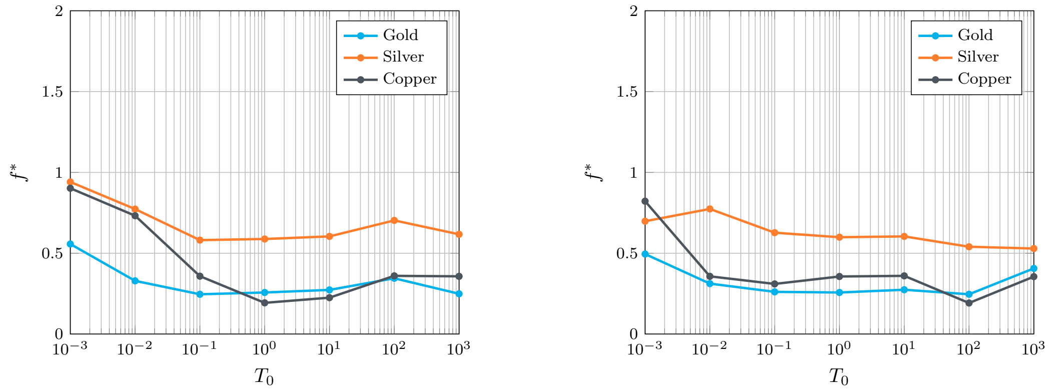
(a) Gold (b) Silver (c) Copper

Figure 7 | Best fits obtained with four poles for gold, silver and copper.



(a) Metropolis function (b) Barker function

Figure 8 | N_A, N_{AU} and N_D as functions of T for gold, with $\alpha = 0.99$ and $T_0 = 1000$. For both acceptance functions, the rule of thumb $N_{AU} \simeq 0.5 \times N_A$ is respected.



(a) Metropolis function (b) Barker function

Figure 9 | Influence of T_0 on f^* for noble metals using the Metropolis and Barker criteria as acceptance functions. Although we can note some differences, the Barker acceptance function yields no clear improvement over the Metropolis one. Please notice that the scale on the y axis is different from that of figure 5.

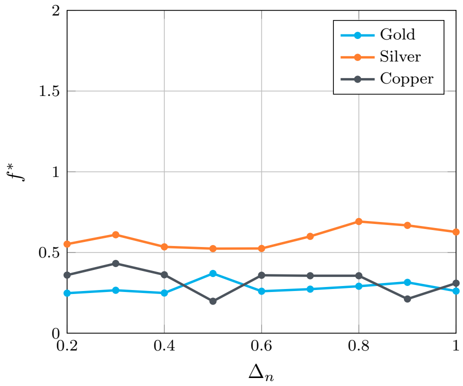


Figure 10 | Influence of Δ_n on f^* for noble metals with $n_p = 4$, $x_0 = 0$, $\alpha = 0.99$ and $T_0 = 0.1$. It is difficult to draw a clear trend from these results, although we observe no significant variations.

4.10 Transition and post-transition metals

In this section, we perform the same test as before with varying T_0 for transition and post-transition metals (see figure 11). For $\alpha = 0.99$, the results for transition metals offer more contrast than that of noble metals, with good fits obtained for $T_0 = 0.1$. However, we see that using a slower cooling schedule with $\alpha = 0.999$ provides more robust results with regard to the variations of T_0. It also improves the best fit obtained in the $T_0 \in [0.1, 1]$ region. For post-transition metals, results are stable for any value of T_0 above 10^{-2}, and significant improvement is observed for higher value of α.
The best fits obtained with four poles for transition and post-transition metals are shown in figure 12.

4.11 Semiconductors

As previously, we assess the influence of T_0 and α on the fits of semiconductors. As could be expected from figure 1, the permittivity functions of such materials are difficult to fit. Hence, they benefit from slower cooling schedules, as can be seen in figure 13. As for other materials, $T_0 = 0.1$ is a robust value across all materials. Below, the algorithm provides poor solutions, especially for TiO2.
The best fits obtained with four poles for semiconductors are shown in figure 14.

4.12 Computational performances

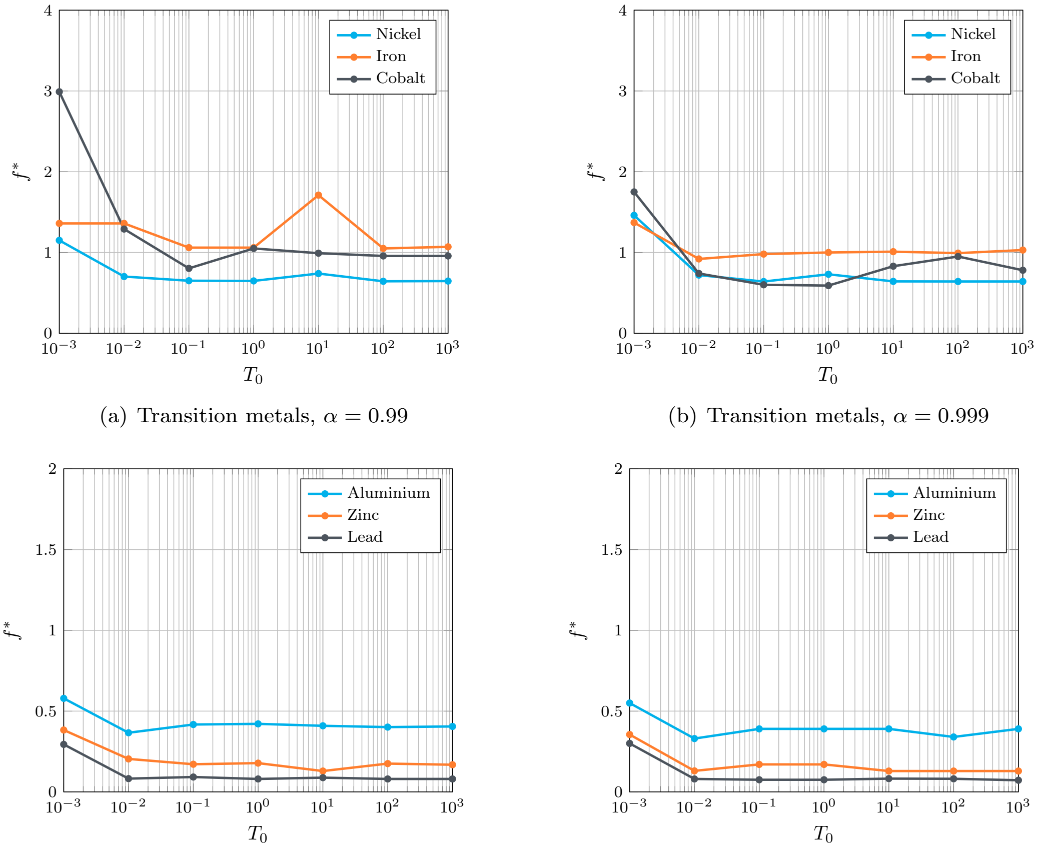
The SA algorithm requires a large amount of evaluation of the cost function. Each evaluation of the latter implies to loop over all experimental data, and for each data point, evaluate the permittivity provided by the current state x. As a result, the performance of the fitting procedure strongly depends on:

- The amount of experimental data points N_s provided by the user;
- The number of poles n_p required by the user;
- The number of neighbor states n_v.

For reasonable values of N_s, the algorithm can be performed on any desktop machine, with running times under a few minutes for standard values of n_p. As an example, we show in figure 15 the time required to fit gold ($N_s = 49$), zinc ($N_s = 148$) with different values of n_p on a standard desktop machine with a 3 GHz Intel Core i7 processor.

5 Numerical results

As a test of usability and stability of the obtained fits in time-domain computations, we consider the case of a doubly-periodic 100 nm-thick slab composed of a given material embedded in air. We successively test



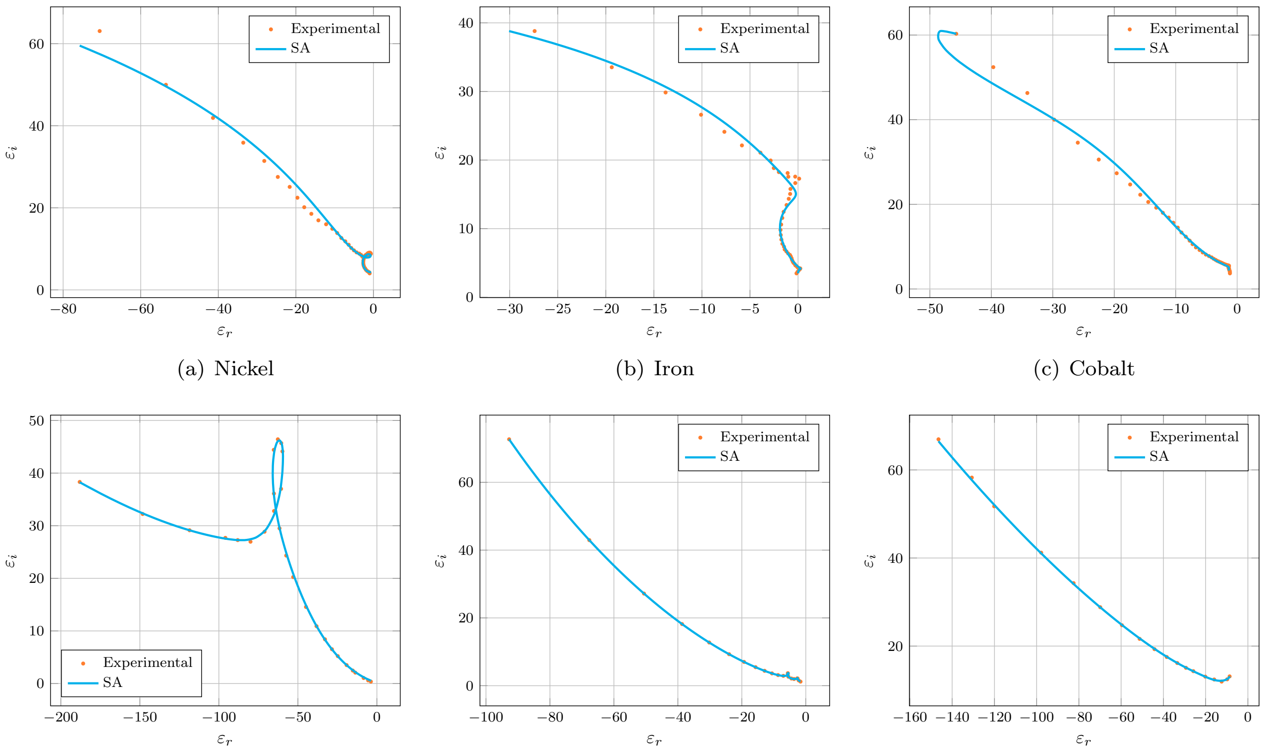
(a) Transition metals, $\alpha = 0.99$

(b) Transition metals, $\alpha = 0.999$

(c) Post-transition metals, $\alpha = 0.99$

(d) Post-transition metals, $\alpha = 0.999$

Figure 11 | Influence of T_0 on f^* for transition and post-transition metals with $n_p = 4$, $x_0 = 0$ and several values of α. Results for transition metals (left) offer more contrast than for noble ones, with good overall results for $T_0 = 0.1$ and $\alpha = 0.999$. For post-transition metals (right), results are remarkably stable for T_0 values above 10^{-2}. Notice the different scales of the y axis.



(a) Nickel

(b) Iron

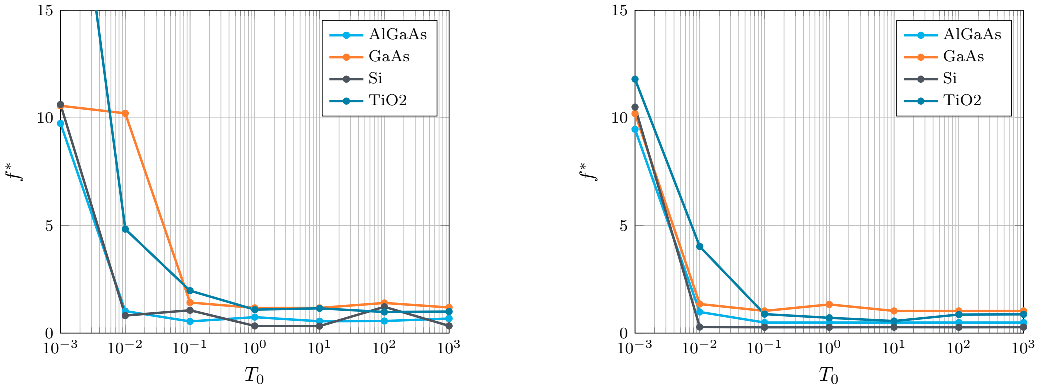
(c) Cobalt

(d) Aluminium

(e) Zinc

(f) Lead

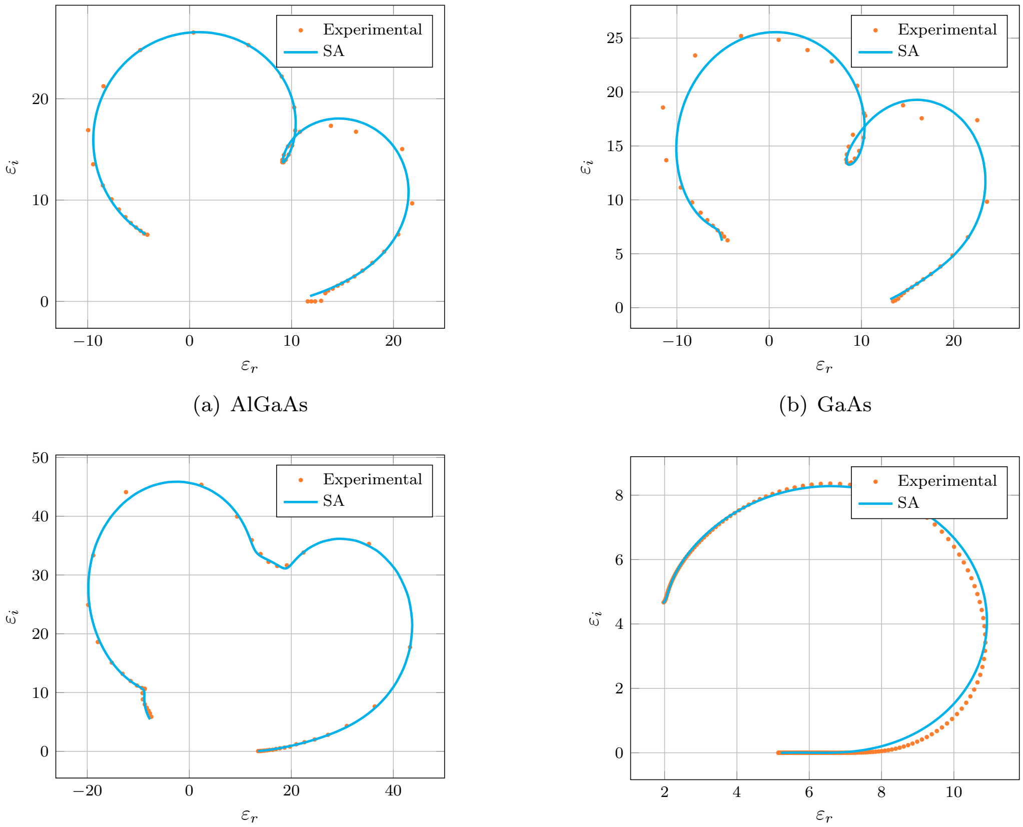
Figure 12 | Best fits obtained with four poles for nickel, iron, cobalt, aluminium, zinc and lead.



(a) Semiconductors, $\alpha = 0.99$

(b) Semiconductors, $\alpha = 0.999$

Figure 13 | Influence of T_0 and α on f^* for semiconductors with $n_p = 4$, $x_0 = 0$. Good fits are obtained for $\alpha = 0.999$ and $T_0 \geq 0.1$, while below this value the algorithm provides poor local minima as final solutions.



(a) AlGaAs

(b) GaAs

(c) Si

(d) TiO2

Figure 14 | Best fits obtained with four poles for AlGaAs, GaAs, Si and TiO2.

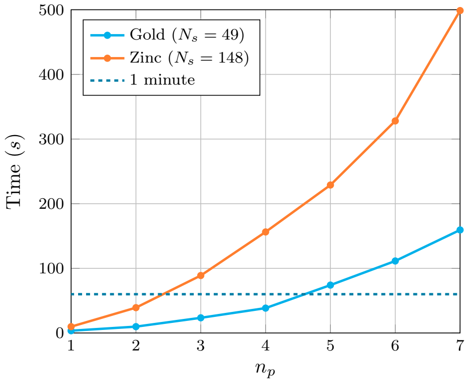


Figure 15 | Time required to fit experimental data with variable number of poles for gold and zinc.

the highest stable time-step Δt_{mat} found with the DIOGENES DGTD solver, and compare it with Δt_{vac}, *i.e.* the maximal stable time-step for the same configuration without material. In table 1, we report the ratio $\frac{\Delta t_{\mathrm{mat}}}{\Delta t_{\mathrm{vac}}}$ for the best four-pole fits obtained with all materials presented in this paper in the $\omega \in [1, 10]$ PHz range. This ratio represents the constraint imposed by the CFL condition inherent to any time-domain numerical method. An unreasonably low ratio would indicate that the set of parameters produced by the fitting procedure is unsuitable for time-domain computations. The restriction to four poles represents a trade-off between the fit accuracy and the memory and CPU overhead induced by each pole in time-domain computations [Viq15]. As can be seen, most fits provide reasonable time-steps. The case of TiO2, for which the obtained time-step is the lowest, can be explained by the large portion of curve for which $\varepsilon_i \equiv 0$ (roughly for $\omega < 4$ PHz), which represents a severe constraint on the fitting algorithm.

Table 1 | Best $\frac{\Delta t_{\mathbf{mat}}}{\Delta t_{\mathbf{vac}}}$ ratios obtained with our DGTD solver with four poles for the materials presented in this paper.

Material	$\frac{\Delta t_{\mathrm{mat}}}{\Delta t_{\mathrm{vac}}}$
Vacuum	1
Dielectric ($\varepsilon_r = 4$)	0.5
Ag	0.306
Au	0.248
Cu	0.467
Al	0.421
Zn	0.791
Pb	0.307
Ni	0.467
Co	0.245
Fe	0.713
AlGaAs	0.312
GaAs	0.276
Si	0.317
TiO2	0.0865

6 Conclusions

Our implementation of the simulated annealing technique for the fitting of dispersive material properties has proven to be efficient, providing reliable sets of parameters while never requiring more than a couple of minutes. Among the numerous intrinsic variables of the algorithm, the cooling factor α and the initial temperature T_0 are the prime parameters to tune for materials involved in most nano-optics problems. The parameter set $(\alpha = 0.99, T_0 = 0.1)$ provided good results for noble and post-transition metals. For transition metals and semi-conductors, we observed that the set $(\alpha = 0.999, T_0 = 0.1)$, although more computationally-intensive, represents a robust choice.

Eventually, we provided an evidence for the usability of the fits produced by our fitting method in any time-domain numerical solver by computing the largest stable time-step obtained with our best four-pole fit for a wide range of materials. Even better fits can be obtained by increasing the number of poles, although at the expense of the performance of the fitting algorithm and the time-domain solver. A possible next step in the improvement of the fitting method would be to incorporate the optimization of the time-step value directly in the cost function of the SA method. As of now, our SA fitting algorithm is available in the DIOGENES software package (see https://diogenes.inria.fr/).

References

[AP10] H. A. Atwater and A. Polman. Plasmonics for improved photovoltaic devices. *Nature Materials*, 9: 205 – 213, 2010.

[CLS$^+$11] T. Chung, S. Y. Lee, E. Y. Song, H. Chun, and B. Lee. Plasmonic nanostructures for nanoscale biosensing. *Sensors*, 11: 10907 – 10929, 2011.

[Dru00] P. Drude. Zur elektronentheorie der metalle. *Annalen der Physik*, 306: 566 – 613, 1900.

[GBM15] B. Gallinet, J. Butet, and J F. Martin. Numerical methods for nanophotonics: standard problems and future challenges. *Laser and Photonics Review*, 9: 577 – 603, 2015.

[GMW81] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization.* Academic Press, 1981.

[Goo11] Jan W. Gooch. *Sellmeier Equation*, pages 653–654. Springer New York, New York, NY, 2011.

[GS99] B. Gustavsen and S. Semlyen. Rational approximation of frequency domain responses by vector fitting. *Transactions on Power Delivery*, 14: 1052 – 1061, 1999.

[HDF06] M. Han, R. W. Dutton, and S. Fan. Model dispersive media in FDTD method with complex-conjugate pole-residue pairs. *Microwave and Wireless Components Letters*, 16: 119 – 121, 2006.

[HJJ03] D. Henderson, S. H. Jacobson, and A. W. Johnson. Theory and practice of simulated annealing. In *Handbook of Metaheuristics*, pages 287 – 319. Springer, 2003.

[HMK03] N. Hansen, S.D. Muller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary Computation*, 11(1):1–18, 2003.

[HN07] F. Hao and P. Nordlander. Efficient dielectric function for FDTD simulation of the optical properties of silver and gold nanoparticles. *Chemical Physics Letters*, 446: 115 – 118, 2007.

[KGV83] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220: 671 – 680, 1983.

[Loc02] M. Locatelli. Simulated annealing algorithms for continuous global optimization. In *Handbook of Global Optimization*, pages 179 – 229. Springer, 2002.

[LSV17] S. Lanteri, C. Scheid, and J. Viquerat. Analysis of a generalized dispersive model coupled to a DGTD method with application to nanophotonics. *SIAM Journal of Scientific Computing*, 39: 831 – 859, 2017.

[NA98] Y. Nourani and B. Andresen. A comparison of simulated annealing cooling strategies. *Journal of Physics A*, 31: 8373 – 8385, 1998.

[Pol] M. N. Polyanskiy. Refractive index database. https://refractiveindex.info.

[Sch97] P. C. Schuur. Classification of acceptance criteria for the simulated annealing algorithm. *Mathematics of operations research*, 22: 266 – 275, 1997.

[SLM17] H. S. Sehmi, W. Langbein, and E. A. Muljarov. Optimizing the Drude-Lorentz model for material permittivity: Method, program, and examples for gold, silver, and copper. *Physical Review B*, 95: 115444 – 115452, 2017.

[Viq15] J. Viquerat. *Simulation of electromagnetic waves propagation in nano-optics with a high-order discontinuous Galerkin time-domain method.* PhD thesis, INRIA Sophia-Antipolis, 2015.

[VLDC11] A. Vial, T. Laroche, M. Dridi, and L. Le Cunff. A new model of dispersion for metals leading to a more accurate modeling of plasmonic structures using the FDTD method. *Applied Physics A*, 203: 849 – 853, 2011.

[WROB13] C. Wolff, R. Rodriguez-Oliveros, and K. Busch. Simple magneto-optic transition metal models for time-domain simulations. *Optics Express*, 21: 12022 – 12037, 2013.

[Xin11] Z. Xinchao. Simulated annealing algorithm with adaptive neighborhood. *Applied Soft Computing*, 11: 1827 – 1836, 2011.