Numerical method for the efficient calculation of scattering form factors

Benjamin Lillard,[1,2] Jack D. Shergold,[3] and Carlos Blanco[1,4,5]

[1]*Institute for Gravitation and the Cosmos, The Pennsylvania State University, University Park, Pennsylvania 16802, USA*

[2]*Department of Physics, University of Oregon, Eugene, Oregon 97403, USA*

[3]*Department of Mathematical Sciences, University of Liverpool, Liverpool, United Kingdom*

[4]*Department of Physics, Princeton University, Princeton, New Jersey 08544, USA*

[5]*Stockholm University and The Oskar Klein Centre for Cosmoparticle Physics, Alba Nova, 10691 Stockholm, Sweden*

(Received 30 April 2026; accepted 30 September 2026; published 6 November 2026)

Scintillating molecular crystals have emerged as prime candidates for directional dark matter detector targets. This anisotropy makes them exquisitely sensitive due to the daily modulation induced by the directional dark matter wind. However, predicting the interaction rate for arbitrary molecules requires accurate modeling of the many-body ground as well as excited states, a task that has been historically computationally expensive. Here, we present a theory and computational framework for efficiently computing dark matter scattering form factors for molecules. We introduce SCarFFF, a GPU-accelerated code to compute the fully three-dimensional anisotropic molecular form factor for arbitrary molecules. We use a full time-dependent density functional theory framework to compute the lowest-lying singlet excited states. Once the many-body electronic structure is computed, the form factors are computed in a small fraction of the time from the transition density matrix. We show that SCarFFF can compute the first 12 form factors for a molecule of 10 heavy atoms in approximately 5 s, opening the door to accurate, high-throughput material screening for optimal directional dark matter detector targets. Our code can perform the calculation in three independent ways, two semianalytical and one fully numeric, providing optimized methods for every precision goal.

DOI: 10.1103/24dn-vndz

I. INTRODUCTION

For thousands of years, scientists have studied the properties of matter by transferring momentum and energy into various systems. Our knowledge of the Universe has been greatly advanced through a cycle of increasingly precise predictions and measurements, addressing the age-old question—"what will happen if I hit it?"—across a vast range of energies and length scales. Currently, all of the types of matter we can manipulate or detect are described precisely by the Standard Model (SM) of particle physics. The primary exception is dark matter, which interacts with SM matter gravitationally [1,2], but has not yet been conclusively detected through other means. The particle description of dark matter (DM) remains one of the outstanding mysteries in modern physics.

In order to discover new particles, we need a precise understanding of our tools: namely, the SM systems that constitute our detectors. In the case of DM direct detection experiments, we need to know how a material responds when a DM particle imparts some momentum ($\mathbf{q}$) and energy (E) onto SM particles. Except for the simplest single-particle systems, predicting this SM material property is computationally expensive, especially when the material is anisotropic, i.e., not spherically symmetric. The difficulty of predicting the material response functions has limited the development of new detector designs, particularly for DM masses below 1 GeV.

In this work, we use a combination of analytic and numerical methods to overcome this limitation for DM-electron scattering. Our JULIA package, SCarFFF[1] (spherical, Cartesian, and Fourier form factors), can calculate new molecular form factors from first principles in $\lesssim$4 s per transition, for molecules around ten heavy atoms. Using SCarFFF to find the form factor and Vector Spaces for Dark Matter (VSDM), a recently published package for efficiently calculating the DM scattering rates [3–5], to calculate the DM scattering rate, a researcher can complete a daily modulation analysis including $\mathcal{O}(10)$ excited states in about one minute per new molecule.

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI. Funded by SCOAP³.

[1]SCarFFF: https://github.com/jdshergold/SCarFFF.

Our simplest method, the discrete Fourier transform of Sec. III, can be used in any system where the initial- and final-state electronic wave functions can be tabulated in position space. It can be the fastest way to calculate the form factor when imprecise results are tolerated. Our fastest analytic method, unveiled in Sec. IV, is specific to physical chemistry calculations performed using a basis of Gaussian-type orbitals. Unlike the fast Fourier transform (FFT), it can be performed without any loss of precision.

Currently, SCarFFF calculates the scalar molecular form factor, also known as the dynamic structure factor or electron loss function, which is sufficient for spin-independent DM-electron scattering [6–10]. A future version of SCarFFF will also calculate the spin response functions necessary for spin-dependent DM interactions [11–14].

II. ELECTRON SCATTERING CALCULATION

A. Spin-independent interaction rate

The rate of DM-e^- scattering events in a material is given by an integral over the DM velocity $\mathbf{v}$ and the momentum transfer $\mathbf{q}$ [15,16],

$$R_{g\to s} = \frac{N_{\rm cell}\rho_\chi/m_\chi}{128\pi^2 m_\chi^2 m_e^2}\int \frac{d^3q}{q}\eta(\mathbf{q},E)|\mathcal{M}_{g\to s}(\mathbf{q})|^2, \quad (1)$$

$$\eta(\mathbf{q},E) \equiv 2q\int d^3v g_\chi(\mathbf{v})\delta\left(E+\frac{q^2}{2m_\chi}-\mathbf{q}\cdot\mathbf{v}\right), \quad (2)$$

where m_χ and ρ_χ are the DM mass and local density, m_e is the electron mass, and $N_{\rm cell}$ is the number of target particles or unit cells in the detector. Following Ref. [5], $\eta(\mathbf{q},E)$ is the integrated DM velocity distribution, defined in terms of the lab-frame velocity distribution $g_\chi(\mathbf{v})$, which satisfies $\int d^3v g_\chi(\mathbf{v}) \equiv 1$.

Our label $g\to s$ refers to a transition from the (multiparticle) electronic ground state $|\Psi_g\rangle$ to a specific excited state $|\Psi_s\rangle$. The scattering amplitude can be written as

$$\mathcal{M}_{g\to s}(\mathbf{q}) = \langle\Psi_s|\delta H|\Psi_g\rangle, \quad (3)$$

where δH is an operator encoding the DM-electron interaction. The details of the electronic bound state can be factorized from the DM-e^- coupling into a free particle amplitude $\mathcal{M}_{\rm free}$ and a material-specific form factor $f_s(\mathbf{q})$,

$$\mathcal{M}_{1\to 2}(\mathbf{q}) = \mathcal{M}_{\rm free}(q)f_s(\mathbf{q}), \quad (4)$$

where in the notation of Refs. [6–10] we define

$$f_s(\mathbf{q}) \equiv \langle\Psi_s|\tilde{n}_e(-\mathbf{q})|\Psi_g\rangle, \quad (5)$$

where $\tilde{n}_e(-\mathbf{q})$ is the Fourier transform of the electronic density operator. It is related to the dynamic structure factor $S(\mathbf{q},\omega)$ of Ref. [6] via [5]

$$S(\mathbf{q},\omega) = \frac{2\pi}{V_{\rm cell}}f_s^2(\mathbf{q})\delta(\omega-E_s), \quad (6)$$

where S is normalized by $V_{\rm cell}$, the volume of the unit cell. For simple systems that are well described by a single particle, the form factor takes the form

$$f_s(\mathbf{q}) = \int\frac{d^3k}{(2\pi)^3}\tilde{\psi}_s^\star(\mathbf{k}+\mathbf{q})\tilde{\psi}_g(\mathbf{k}), \quad (7)$$

where $\tilde{\psi}_i$ are the momentum space wave functions of the initial and final states. As we show in Sec. II B, the form factor for more complicated multiparticle systems can often be put into this form.

It is typical to write $\mathcal{M}_{\rm free}(q)$ in terms of a q-dependent form factor $F_{\rm DM}^2$,

$$F_{\rm DM}^2(q) \equiv \frac{|\mathcal{M}_{\rm free}(q)|^2}{|\mathcal{M}_{\rm free}(\alpha m_e)|^2}, \quad (8)$$

and a constant cross section $\bar{\sigma}_e$ that parametrizes the strength of the DM-SM coupling,

$$\bar{\sigma}_e \equiv \frac{\mu_{\chi e}^2|\mathcal{M}_{\rm free}(\alpha m_e)|^2}{16\pi m_e^2 m_\chi^2}. \quad (9)$$

With these replacements, the rate integral takes the compact form

$$R_{1\to 2} = \frac{N_{\rm cell}\rho_\chi\bar{\sigma}_e}{8\pi m_\chi\mu_{\chi e}^2}\int\frac{d^3q}{q}\eta(\mathbf{q},E)F_{\rm DM}^2(q)|f_s(\mathbf{q})|^2. \quad (10)$$

Figure 1 summarizes the physical system. The interaction between dark matter (or a Standard Model particle) and an electron in a momentum eigenstate can be calculated using Feynman diagrams for free particles, yielding the

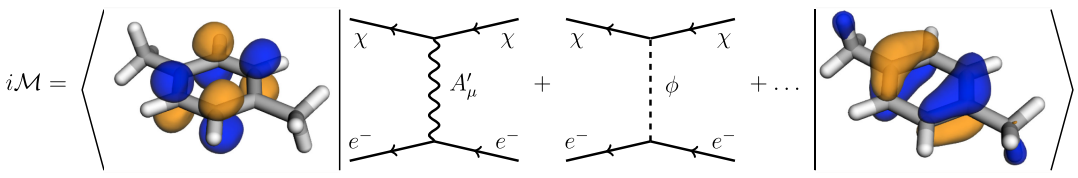


FIG. 1. A weakly interacting particle χ scatters with an electron in a multiparticle ground state, transferring to it some momentum and energy, and leaving it in a higher energy eigenstate.

$\mathcal{M}_{\text{free}}$ contribution to the amplitude. The probability of finding an electron of momentum $\mathbf{k}$ in the initial state and an electron of momentum $\mathbf{k}' = \mathbf{k} + \mathbf{q}$ in the final state is calculated using methods from physical chemistry.

B. Electronic structure calculation: Single-particle orbitals

To determine $f_s(\mathbf{q})$, we must begin with an accurate description of the initial and final multielectron states. We work in a simple linear combination of atomic orbitals (LCAO) model, where single-particle molecular orbitals are linear combinations of atomic orbitals $\phi_{aI}(\mathbf{r}) = \sum_\alpha c_\alpha \phi_\alpha(\mathbf{r} - \mathbf{r}_I)$ centered on the nuclear equilibrium locations $\mathbf{r}_I$. In our discussion of the multielectron wave functions, the Hartree-Fock ground state is a useful starting point. It is a spin-singlet Slater determinant of n electrons in the first $n' = n/2$ single-particle molecular orbitals (MOs), ordered by energy

$$|\psi_0\rangle = |\varphi_1\bar{\varphi}_1 \ldots \varphi_{n'}\bar{\varphi}_{n'}|, \tag{11}$$

where $|\varphi_i \ldots \bar{\varphi}_j|$ indicates the antisymmetric Slater determinant, with φ and $\bar{\varphi}$ denoting the two electronic spin states. The first $i \leq n'$ MOs are usually referred to as "occupied," and the remaining $a \geq n'$ orbitals "virtual" or "unoccupied." In this notation a singly excited state $|\psi_i^a\rangle$ replaces an occupied MO (φ_i) with a virtual one (φ_a). Since our primary focus is on prompt fluorescence in organic scintillators, we restrict our attention to the singlet configurations: that is,

$$|\psi_i^a\rangle = \frac{1}{\sqrt{2}}|\varphi_1\bar{\varphi}_1 \ldots \varphi_i\bar{\varphi}_a \ldots \varphi_{n'}\bar{\varphi}_{n'}| - \frac{1}{\sqrt{2}}|\varphi_1\bar{\varphi}_1 \ldots \varphi_a\bar{\varphi}_i \ldots \varphi_{n'}\bar{\varphi}_{n'}|, \tag{12}$$

$$|\Psi_s\rangle \simeq \sum_{i,a} |\psi_i^a\rangle\langle\psi_i^a|\Psi_s\rangle. \tag{13}$$

Restricting our calculation to singlet states, we can relate $|\psi_i^a\rangle$ to the $|\psi_0\rangle$ state via creation and annihilation operators $c_p^\dagger$ and c_p,

$$|\psi_i^a\rangle = c_a^\dagger c_i |\psi_0\rangle. \tag{14}$$

Triplet excited states are associated with the delayed fluorescence component of the scintillation light, i.e., phosphorescence, which is generally much less efficient than the prompt fluorescence signal. For this reason, we will postpone a detailed treatment of triplet states to future work.

Because the Hartree-Fock $|\psi_0\rangle$ state does not include effects from electron correlations, it is not a precise description of the actual ground state $|\Psi_g\rangle$. Instead, we allow $|\Psi_g\rangle$ to receive contributions from doubly excited configurations of the form $|\psi_{ab}^{ij}\rangle$, where two of the occupied orbitals have been "promoted" to $a, b > n'$. In this basis,

$$|\Psi_g\rangle \simeq (1 - \epsilon^2)|\psi_0\rangle + \sum_{ij,ab} |\psi_{ij}^{ab}\rangle\langle\psi_{ij}|\Psi_g\rangle, \tag{15}$$

where ϵ^2 is a normalization factor.

We adopt a molecular orbital model where transitions between the ground state and excited states are found to be roots of the Casida equation [17]. Specifically, SCarFFF makes use of the linear-response time-dependent density functional theory (TD-DFT) framework. Therefore, both the positive (X_{ia}) and negative (Y_{ia}) eigenvalue solutions of the Casida equation are included, combining to form the quasiboson excitation operator,

$$Q_s = \sum_{\sigma,\tau} \sum_{ia} X_{ia;\sigma\tau}^{(s)} c_{a,\tau}^\dagger c_{i,\sigma} - Y_{ia;\sigma\tau}^{(s)} c_{i,\sigma}^\dagger c_{a,\tau}, \tag{16}$$

which acts on the ground state to produce an excited state, $Q_s|\Psi_g\rangle = |\Psi_s\rangle$, where $\sigma, \tau \in \{\alpha, \beta\}$ label the two spin channels. We specialize to singlet-singlet transitions using a restricted Kohn-Sham reference, for which the amplitudes are spin-diagonal and identical in the two spin channels: $X_{ia;\sigma\tau} = \delta_{\sigma\tau} X_{ia}$, and analogous for Y. The normalization includes both spin channels, such that $2\sum_{ia}(X_{ia}^2 - Y_{ia}^2) = 1$. Heuristically, the X_{ia} matrix characterizes the single excitation character of the transition, while Y_{ia} characterizes the correlation effects and the double excitation character of the ground state. Taking $Y \to 0$ is equivalent to the Tamm-Dancoff approximation in linear-response theory. The $N_o \times N_v$ matrices X and Y in the MO basis can be combined into a single $(N_o + N_v)^2$ square matrix in the atomic orbital (AO) basis $T_{pq}^{(s)}$,

$$T_{pq}^{(s)} = \sum_{ia} C_{pi}^{\text{occ}} [X_{ia}^{(s)} + Y_{ia}^{(s)}] C_{qa}^{\text{vir}}, \tag{17}$$

where C_{pi}^{occ} and C_{qa}^{vir} are the MO coefficient matrices that rotate from the AO basis to the diagonal MO basis.

Compared to the single-particle f_s of Eq. (7), it is no longer straightforward to identify a single pair of initial and final states. For visualizing the transition in Fig. 1, we used the singular value decomposition of $T_{pq}^{(s)}$ to identify the natural transition orbitals of Ref. [18]. SCarFFF works with $T_{pq}^{(s)}$ directly in the atomic orbital basis.

Following our previous work, the relevant single-particle observables of the many-body wave function can be expressed as a sum of single-particle matrix elements. The scattering form factor, characterizing the sth singlet transition, is given by

$$f_s(\mathbf{q}) = 2T_{ia}^{(s)} \langle\varphi_a(\mathbf{r})|e^{i\mathbf{q}\cdot\mathbf{r}}|\varphi_i(\mathbf{r})\rangle, \tag{18}$$

where the factor of 2 in the form factor comes from summing over the identical contributions from each spin channel. Our molecular orbitals in SCarFFF are expanded in a basis of Gaussian-type orbitals, such as 6-31G*, cc-pVDZ, or cc-pVTZ. More generally, SCarFFF currently supports several basis sets, constructed from up to i orbitals.

We have adopted specific exchange functionals and basis sets as a benchmark for the results of this study. However, the functionality of SCarFFF depends only on the availability of a transition density matrix and a selection of any Gaussian basis set. We suggest using the 6-31g* basis set to obtain qualitatively correct results and double or triple ζ basis sets for quantitatively robust results, e.g., cc-pVDZ or cc-pVTZ. See Appendix A for a detailed discussion of our basis-set conventions.

III. FFT METHOD

The molecular form factor $f_s(\mathbf{q})$ can be calculated using either position space or momentum space wave functions. Following Sec. II B,

$$f_s(\mathbf{q}) = \int d^3r e^{i\mathbf{q}\cdot\mathbf{r}} \Phi_{g\to s}(\mathbf{r}), \tag{19}$$

$$\Phi_{g\to s}(\mathbf{r}) = 2\sum_{pq} T^{(s)}_{pq} \phi^\star_q(\mathbf{r})\phi_p(\mathbf{r}), \tag{20}$$

where $T^{(s)}_{pq}$ is the transition density matrix for the $g \to s$ excitation in the basis of atomic orbitals ($\phi_{p,q}$). With this notation, $f_s(-\mathbf{q})$ is simply the Fourier transform of $\phi T \phi(\mathbf{r})$.

It would be prohibitively time consuming to evaluate $f_s(\mathbf{q})$ by integrating Eq. (19) at each point $\mathbf{q}$ in momentum space. However, there are highly efficient numerical routines for approximating the Fourier transform of a tabulated d-dimensional function, namely, the discrete FFT [19]. We find that the FFT is a viable method for extracting $f_s(\mathbf{q})$ from tabulated transition density functions, especially when high precision is not required.

A. The (fast) Fourier transform

We define the forward and backward Fourier transforms as, respectively,

$$\mathcal{F}[f](\mathbf{q}) = \int d^3x e^{-i\mathbf{q}\cdot\mathbf{x}} f(\mathbf{x}), \tag{21}$$

$$\mathcal{F}^{-1}[\tilde{f}](\mathbf{x}) = \int \frac{d^3k}{(2\pi)^3} e^{+i\mathbf{k}\cdot\mathbf{x}} \tilde{f}(\mathbf{k}). \tag{22}$$

With this notation, f_s is simply

$$f^\star_s(\mathbf{q}) = \mathcal{F}[\Phi^\star_{g\to s}](\mathbf{q}), \tag{23}$$

or equivalently, $f_s(\mathbf{q}) = \mathcal{F}[\Phi_{g\to s}](-\mathbf{q})$.

Compare this to the discrete Fourier transform. In three dimensions, we define the forward FFT of a 3D array $f_{abc} = f(x_a, y_b, z_c)$ as

$$F_{\alpha\beta\gamma} \equiv \sum_{a=0}^{N_x-1}\sum_{b=0}^{N_y-1}\sum_{c=0}^{N_z-1} f_{abc} \exp\left(-2\pi i\left(\frac{a\alpha}{N_x} + \frac{b\beta}{N_y} + \frac{c\gamma}{N_z}\right)\right), \tag{24}$$

for integers $\alpha = 0, 1, \ldots, N_x - 1$, $\beta = 0, 1, \ldots N_y - 1$, etc.

Let us tabulate $f(\mathbf{x})$ on a regularly spaced rectangular grid, within a volume $V = L_x L_y L_z$ bounded by

$$x \in [x_{\min}, x_{\max}], \quad y \in [y_{\min}, y_{\max}], \quad z \in [z_{\min}, z_{\max}], \tag{25}$$

where we define $L_i \equiv x^{(i)}_{\max} - x^{(i)}_{\min}$ as the length of each dimension of the box. Defining

$$u_i \equiv \frac{x_i - x^{(i)}_{\min}}{L_i}, \qquad x_i = L_i u_i + x^{(i)}_{\min}, \tag{26}$$

the Fourier transform of $f(\mathbf{x})$ is

$$\begin{aligned} F(\mathbf{k}) &= \int_V d^3x e^{-i\mathbf{k}\cdot\mathbf{x}} f(\mathbf{x}) \\ &\simeq \sum_{abc} V_{\text{pix}} \exp\left(-i\mathbf{k}\cdot[L_i u_i + x^{(i)}_{\min}]\right) f_{abc}, \end{aligned} \tag{27}$$

where f_{abc} is the value of $f(\mathbf{x})$ at the grid position $\mathbf{x} = \mathbf{x}_{abc}$, defined via

$$\mathbf{u}_{abc} = \frac{a}{N_x}\hat{x} + \frac{b}{N_y}\hat{y} + \frac{c}{N_z}\hat{z}, \tag{28}$$

with V_{pix} the volume of a single pixel in the grid,

$$V_{\text{pix}} = \frac{L_x L_y L_z}{N_x N_y N_z}. \tag{29}$$

Pulling out the overall phase from the translation in $\mathbf{x} \to \mathbf{u}$,

$$\begin{aligned} F(\mathbf{k}) \simeq V_{\text{pix}} e^{-i\mathbf{k}\cdot\mathbf{x}_{\min}} \sum_{abc} f_{abc} \exp\Bigg(-2\pi i\Bigg[&\frac{k_x L_x}{2\pi}\frac{a}{N_x} \\ &+ \frac{k_y L_y}{2\pi}\frac{b}{N_y} + \frac{k_z L_z}{2\pi}\frac{c}{N_z}\Bigg]\Bigg). \end{aligned} \tag{30}$$

Standard FFT routines, given the array f_{abc} as an input, will return $F_{\alpha\beta\gamma}$. From the 2π periodicity of the complex phase in Eq. (24), note that $(N_x - \alpha)/N_x$ describes the same Fourier mode as $-\alpha/N_x$, and so the FFT output needs to be

rearranged by cycling the arrays until the most negative frequencies ($\alpha \geq N_x/2$) appear first in the list. Once this is done, all we need to do is multiply $F_{\alpha\beta\gamma}$ by the pixel volume $V_{\rm pix}$ and the phase associated with translation from the origin to the corner of the grid,

$$F(k_x, k_y, k_z) = F\left(\frac{2\pi\alpha}{L_x}, \frac{2\pi\beta}{L_y}, \frac{2\pi\gamma}{L_z}\right) = V_{\rm pix} e^{-i\mathbf{k}\cdot\mathbf{x}_{\rm min}} F_{\alpha\beta\gamma}. \tag{31}$$

Returning to the specific example of the momentum form factor $f_s(\mathbf{q})$,

$$f_s^\star(\mathbf{q}) = \mathcal{F}[\Phi_{1\to2}^\star] \simeq V_{\rm pix} e^{+i\mathbf{q}\cdot\mathbf{x}_{\rm min}} F_{\alpha\beta\gamma}^\star = V_{\rm pix} \exp(+i\mathbf{q}\cdot\mathbf{x}_{\rm min}) F_{\alpha\beta\gamma}^\star, \tag{32}$$

where

$$\mathbf{q} = \frac{2\pi\alpha}{L_x}\hat{x} + \frac{2\pi\beta}{L_y}\hat{y} + \frac{2\pi\gamma}{L_z}\hat{z}, \tag{33}$$

and where $F_{\alpha\beta\gamma}^\star$ is to be understood as the forward FFT of the complex conjugate array,

$$f_{abc}^\star = \Phi_{1\to2}^\star(x_a, y_b, z_c). \tag{34}$$

B. Practical considerations

The primary benefits of the FFT are its simplicity and easy adaptability: once $\Phi_{g\to s}(\mathbf{r})$ has been tabulated, it no longer matters which types of basis functions were used to generate it. On the other hand, an accurate depiction of $f_s(\mathbf{q})$ requires a position space grid that is both large in volume and finely sampled in $\mathbf{r}$.

From Eq. (33), the "pixel" size in the momentum space grid is given by $2\pi/L$, where L is the length of the integration volume. Generating a 0.1 keV grid in $f_s(\mathbf{q})$ would necessitate an $L \approx 124$ Å box size, which is 2 orders of magnitude larger than a typical bond length. Simultaneously, we still need to sample $\Phi_{g\to s}(\mathbf{r})$ finely enough to capture all of its relevant features, otherwise Eq. (27) will be inaccurate. Even if we set $\Phi_{g\to s}(\mathbf{r}) \to 0$ for $\mathbf{r} \gg$ few $\times\, a_0$ far away from the molecule, such a 3D grid can easily exceed the available computer memory.

Even so, the FFT can be highly efficient at generating low-to-medium precision form factors. Figure 2 shows the convergence toward accuracy as the integration volume is expanded, using the first excited state of p-xylene as an example. Paraxylene is symmetric under central inversion, $\mathbf{r} \to -\mathbf{r}$, which implies that $f_s = i\,\mathrm{Im}\, f_s$ for any excited states with a nonzero transition dipole moment [14]. Initially, $\Phi_{g\to s}$ is tabulated within a rectangular box of dimensions $L_{x,y,z} \simeq (12.0, 9.7, 7.4)$ Å, which includes a margin of $5a_0$ in all directions away from the atomic positions. This scale = 1 FFT does a poor job of recreating f_s, even qualitatively, except for the locations of the $f_s = 0$ nodes. The grid spacing $\Delta x = 0.07$ Å is more than sufficient to capture the small-scale features in $\Phi_{g\to s}(\mathbf{r})$, so the imprecision is due to $L_{x,y,z}$.

In Figs. 2(b) and 2(c), the integration region is extended to scale $\times$ (12.0 Å) in every direction, for scale = 2 and scale = 3. The $L \approx 24$ Å box correctly reproduces all of the features in $f_s(\mathbf{q})$, with a pixelation that is apparent to the eye. At $L \approx 36$ Å, $f_s(\mathbf{q})$ resolves into a clearer image,

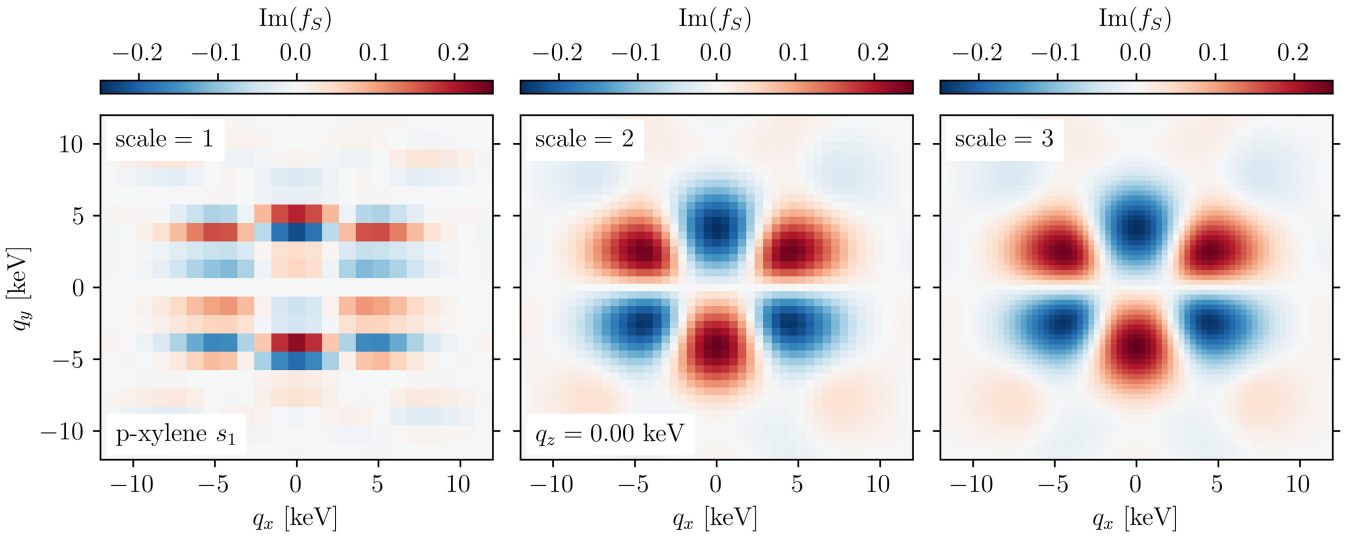


FIG. 2. The accuracy of the FFT method is controlled in part by the volume of the sampled region. The "scale = 1" volume encloses the atomic coordinates in a rectangular box with a $5a_0$ margin in all directions. Its qualitative accuracy is poor, although it correctly identifies the order of magnitude. In the "scale = 2" and "scale = 3" examples, we extend the integration volume by a factor of 2 or 3 before applying the FFT, yielding more accurate and finely grained pictures of $f_s(\mathbf{q})$, shown here in the $q_z = 0$ plane.

without revealing any substantial differences with the scale = 2 version. Whether the inaccuracy in f_s will affect the scattering rate in Eq. (10) depends largely on the DM model in question: the accuracy is probably sufficient for heavy mediator models, $F_{\rm DM} = 1$. For light mediators, $F_{\rm DM} = (\alpha m_e)^2/q^2$, the enhancement at small $q \lesssim \alpha m_e$ will magnify the pixelation error.

In conclusion, while the FFT method is simple to implement, we must ensure that the position space and momentum space grids are both sufficiently fine-grained to capture all of the physical effects in the system. A compromise can often be found between accuracy and speed. In the next section, we introduce a pair of analytic methods that evaluate $f_s(\mathbf{q})$ exactly, at arbitrary values of $\mathbf{q}$. By "exact," we mean that we integrate Eq. (19) without any loss of precision.

IV. ANALYTIC METHOD

The Fourier transform method, while conceptually simple, suffers from the grid resolution issues discussed in Sec. III B. As such, SCarFFF also implements methods based on analytic expressions for the form factor, which we will now discuss.

Our starting point is the expression for the form factor, Eq. (19), written in the atomic orbital basis as

$$f_s(\mathbf{q}) = 2\sum_{\alpha\beta,IJ}\int d^3r \phi^*_\alpha(\mathbf{r}-\mathbf{r}_I)\phi_\beta(\mathbf{r}-\mathbf{r}_J)T^{(s)}_{\alpha\beta}e^{i\mathbf{q}\cdot\mathbf{r}}, \quad (35)$$

with $T^{(s)}_{\alpha\beta}$ the transition density matrix for singlet state s in the AO basis, and ϕ_α the Gaussian-type orbital (GTO) corresponding to orbital α, defined in Appendix A. Each of these orbitals can be expanded in terms of sums of primitives, which can, in turn, be expanded in terms of sums of their individual Cartesian polynomial terms. The indices I and J refer to individual atoms within the molecule, located at positions $\mathbf{r}_I$ and $\mathbf{r}_J$, respectively.

To simplify notation and avoid confusion, we note that each Cartesian term belongs a primitive, which, in turn, belongs an orbital and atom within the molecule. We will therefore express everything in terms of the Cartesian term indices i and j in what follows, so that, e.g., $T^{(s)}_{ij}$ corresponds to the component of $T^{(s)}_{\alpha\beta}$ implied by Cartesian terms i and j, and $\mathbf{r}_i$ similarly refers to the position $r_{\mathbf{I}}$. This reduces the outer sum to a simpler one over i and j.

Expanding out the GTOs, we therefore find

$$f_s(\mathbf{q}) = 2\sum_{i,j}T^{(s)}_{ij}C_{ij}\int_{-\infty}^{\infty}dx(x-x_i)^{a_i}(x-x_j)^{a_j} \times \exp\left(iq_x - \frac{(x-x_i)^2}{2\sigma_i^2} - \frac{(x-x_i)^2}{2\sigma_i^2}\right)\int_{-\infty}^{\infty}dy\ldots, \quad (36)$$

where C_{ij} contains all of the primitive scale factors, normalization coefficients, and Cartesian term weights for this ij pair, and where a_i and a_j are the exponents of the Cartesian terms, e.g., $a = 1$ for the p_x orbital or $a = 0$ for an s orbital. We define b_i and c_i similarly for the y and z components of an orbital.

Each of these one-dimensional integrals can be solved by first completing the square on the polynomial appearing in the exponent, followed by a redefinition of the integral measure to leave

$$f_s(\mathbf{q}) = 2\sum_{i,j}T^{(s)}_{ij}C_{ij}\exp\left(-\frac{r_{ij}^2}{2\bar{\sigma}_{ij}^2} + i\mathbf{q}\cdot\mathbf{R}_{ij}\right) \times \int_{-\infty}^{\infty}dx(x+\mathcal{X}_{ij}-x_i)^{a_i}(x+\mathcal{X}_{ij}-x_j)^{a_j} \times \exp\left(iq_x x - \frac{x^2}{2\sigma_{ij}^2}\right)\int_{-\infty}^{\infty}dy\ldots, \quad (37)$$

where we have defined

$$\sigma_{ij} = \frac{\sigma_i\sigma_j}{\bar{\sigma}_{ij}}, \qquad \bar{\sigma}_{ij} = \sqrt{\sigma_i^2+\sigma_j^2}, \quad (38)$$

and

$$\mathcal{X}_{ij} = \sigma_{ij}^2\left(\frac{x_i}{\sigma_i^2}+\frac{x_j}{\sigma_j^2}\right), \qquad x_{ij} = |x_i - x_j|, \quad (39)$$

with similar expressions for $\mathcal{Y}_{ij}$ and $\mathcal{Z}_{ij}$. $\mathbf{R}_{ij}$ is simply the vector $(\mathcal{X}_{ij}, \mathcal{Y}_{ij}, \mathcal{Z}_{ij})$. To further facilitate the integral, the polynomials can be conveniently rewritten using the binomial theorem as

$$(x+\mathcal{X}_{ij}-x_i)^{a_i}(x+\mathcal{X}_{ij}-x_j)^{a_j} = \sum_{A=0}^{\alpha_{ij}}b_{ij}^A x^A, \quad (40)$$

with $\alpha_{ij} = a_i + a_j$, and the b coefficients defined by

$$b_{ij}^A = \sum_{k=\max\{0,a_i-A\}}^{\min\{a_i,\alpha_{ij}-A\}}\left(\frac{a_i}{k}\right)\left(\frac{a_j}{\alpha_{ij}-k-A}\right) \times (\mathcal{X}_{ij}-x_i)^k(\mathcal{X}_{ij}-x_j)^{\alpha_{ij}-k-A}, \quad (41)$$

where once again, we define similar b coefficients for the y and z directions indexed by B and C. This allows us to write each of the integrals in terms of differential operators and perform the resulting Gaussian integral to leave

$$f_s(\mathbf{q}) = 2(2\pi)^{\frac{3}{2}} \sum_{i,j} T_{ij}^{(s)} M_{ij}, e^{i\mathbf{q}\cdot\mathbf{R}_{ij}} \times \sum_{A=0}^{\alpha_{ij}} b_{ij}^{A} (-i\partial_{q_x})^A \exp\left(-\frac{q_x^2\sigma_{ij}^2}{2}\right) \sum_{B=0}^{\beta_{ij}} \ldots, \quad (42)$$

where concretely,

$$M_{ij} = d_i' d_j' k_i k_j \sigma_{ij}^3 \exp\left(-\frac{r_{ij}^2}{2\bar{\sigma}_{ij}^2}\right), \quad (43)$$

where $d_i' = d_i \xi_i N_i$, with the orbital normalization factors ξ, N, and d defined in Appendix A. Finally, noticing that the derivative term can be rewritten in terms of the probabilist's Hermite polynomials

$$(-i\partial_{q_x})^A \exp\left(-\frac{q_x^2\sigma_{ij}^2}{2}\right) = (i\sigma_{ij})^A \mathrm{He}_A(q_x\sigma_{ij}) \times \exp\left(-\frac{q_x^2\sigma_{ij}^2}{2}\right), \quad (44)$$

we arrive at a useful expression for the form factor, written in terms of three tensors, each depending on just one of the three Cartesian momentum coordinates

$$f_s(\mathbf{q}) = 2(2\pi)^{\frac{3}{2}} \sum_{i,j} T_{ij}^{(s)} M_{ij} \mathcal{V}_{ij}^{(x)}(q_x) \mathcal{V}_{ij}^{(y)}(q_y) \mathcal{V}_{ij}^{(z)}(q_z), \quad (45)$$

where each $\mathcal{V}$ tensor is defined by

$$\mathcal{V}_{ij}^{(x)}(q_x) = \exp\left(iq_x\mathcal{X}_{ij} - \frac{q_x^2\sigma_{ij}^2}{2}\right) \sum_{A=0}^{\alpha_{ij}} b_{ij}^{A} (i\sigma_{ij})^A \times \mathrm{He}_A(q_x\sigma_{ij}). \quad (46)$$

For those more inclined to use the physicist's Hermite polynomials, these are related by

$$\mathrm{He}_n(x) = 2^{-\frac{n}{2}} H_n\left(\frac{x}{\sqrt{2}}\right). \quad (47)$$

The expression (45) is particularly useful for evaluating the form factor, as each of the $\mathcal{V}$ tensors is a sum of simple polynomials in each of the momenta. Additionally, as we will discuss at length in Sec. V C, being able to decompose the form factor into the three momentum directions can be particularly useful when evaluating certain kinds of integrals involving the form factor. We therefore implement Eq. (45) as one of the methods in SCarFFF.

We now note that Eq. (45) is not a the sole useful analytic expression for the form factor. For several reasons, which we will discuss in Sec. V D, it is often more convenient and computationally efficient to compute the form factor in spherical coordinates. To do so, we first rewrite the product of Hermite polynomials contained in each of the $\mathcal{V}$ tensors, packaged alongside several of the other coefficients, in terms of a sum of monomials, to arrive at

$$f_s(\mathbf{q}) = 2(2\pi)^{\frac{3}{2}} \sum_{i,j} T_{ij}^{(s)} \exp\left(-\frac{q^2\sigma_{ij}^2}{2}\right) e^{i\mathbf{q}\cdot\mathbf{R}_{ij}} \times \sum_{u,v,w} \mathcal{D}_{ij}^{uvw} q_x^u q_y^v q_z^w, \quad (48)$$

where u runs from 0 to α_{ij}, and v and w follow the same pattern for the y and z directions, while the $\mathcal{D}$ tensor is defined by

$$\mathcal{D}_{ij}^{uvw} = M_{ij} \mathcal{C}_{ij}^u \mathcal{C}_{ij}^v \mathcal{C}_{ij}^w, \quad (49)$$

with

$$\mathcal{C}_{ij}^u = i^u \sum_{m=0}^{\lfloor \frac{\alpha_{ij}-u}{2} \rfloor} b_{ij}^{u+2m} \sigma_{ij}^{2(u+m)} \frac{(u+2m)!}{m!u!2^m}, \quad (50)$$

and similar in v and w for the y and z components. The next step is somewhat more complicated and involves projecting the monomials onto a basis of spherical harmonics. As the derivation is rather opaque, we leave this in Appendix B for the more interested reader and simply quote the resulting form factor here,

$$f_s(\mathbf{q}) = 2(2\pi)^{\frac{3}{2}} \sum_{i,j} T_{ij}^{(s)} \exp\left(-\frac{q^2\sigma_{ij}^2}{2}\right) e^{i\mathbf{q}\cdot\mathbf{R}_{ij}} \times \sum_{u,v,w} \mathcal{D}_{ij}^{uvw} q^n \sum_{\lambda \in \{n,n-2,\ldots\}} \sum_{\mu=-\lambda}^{\lambda} \Theta_{\lambda\mu}^{uvw} Y_\lambda^\mu(\hat{q}), \quad (51)$$

with the $\Theta_{\lambda\mu}^{uvw}$ coefficients also defined in Appendix B, $n = u + v + w$, and where Y_λ^μ denotes a complex spherical harmonic, with the Condon-Shortley phase included. After a careful reindexing and reordering of the summation order, this expression can be simplified to

$$f_s(\mathbf{q}) = 2(2\pi)^{\frac{3}{2}} \sum_{i,j} T_{ij}^{(s)} \exp\left(-\frac{q^2\sigma_{ij}^2}{2}\right) e^{i\mathbf{q}\cdot\mathbf{R}_{ij}} \times \sum_{n} q^n \sum_{\lambda \in \{n,n-2,\ldots\}} \sum_{\mu=-\lambda}^{\lambda} \mathcal{W}_{ij,\lambda\mu}^n Y_\lambda^\mu(\hat{q}), \quad (52)$$

where

$$\mathcal{W}_{ij,\lambda\mu}^n = \sum_{\substack{u,v,w \\ u+v+w=n}} \mathcal{D}_{ij}^{uvw} \Theta_{\lambda\mu}^{uvw}. \quad (53)$$

Owing to its symmetries and selection rules, explicitly, its invariance under $i \leftrightarrow j$, along with the conditions

$n \geq \lambda \geq |\mu|$, $|\mu| \leq u + v$, and that n and λ must share the same parity, the $\mathcal{W}$ tensor turns out to be a particularly efficient object to compute and forms the basis for the spherical grid method implemented in SCarFFF. However, we have yet to completely disentangle the radial and angular components of the form factor. The next step in this process is to perform a plane-wave expansion of the phase factor

$$e^{i\mathbf{q}\cdot\mathbf{R}_{ij}} = 4\pi \sum_{L,M} i^L j_L(qR_{ij}) Y_L^M(\hat{q}) Y_L^{M*}(\hat{R}_{ij}), \tag{54}$$

with j_L the spherical Bessel function of order L. The spherical harmonic that depends on $\hat{q}$ resulting from this expansion can be coupled to an existing one from the monomial-to-spherical transformation to a Gaunt coefficient, as

$$Y_\lambda^\mu(\hat{q}) Y_L^M(\hat{q}) = \sum_\ell \mathcal{G}^{\ell m}_{\lambda\mu,LM} Y_\ell^m(\hat{q}) \tag{55}$$

with the Gaunt coefficient defined by

$$\mathcal{G}^{\ell m}_{\lambda\mu,LM} = (-1)^m \sqrt{\frac{(2\lambda+1)(2L+1)(2\ell+1)}{4\pi}} \times \begin{pmatrix} \lambda & L & \ell \\ \mu & M & -m \end{pmatrix} \begin{pmatrix} \lambda & L & \ell \\ 0 & 0 & 0 \end{pmatrix}, \tag{56}$$

where the large bracketed objects are Wigner-3j symbols, whose selection rules fix $M = m - \mu$ hereafter. We note that the Gaunt coefficients can also be written as the integral of a product of three spherical harmonics, but that $\mathcal{G}^{\ell m}_{\lambda\mu,LM}$ is more efficiently calculated from the Wigner-3j symbols.

This leaves us with our final expression for the form factor,

$$f_s(\mathbf{q}) = \sum_{\ell,m} Y_\ell^m(\hat{q}) \mathcal{R}_{\ell m}(q), \tag{57}$$

which conveniently factorizes the angular dependence from the $|\mathbf{q}|$-dependent function $\mathcal{R}_{\ell m}(q)$. All of the difficult analytic structure is contained within the $\mathcal{R}$ tensor,

$$\mathcal{R}_{\ell m}(q) = 4(2\pi)^{\frac{5}{2}} \sum_{i,j} T_{ij}^{(s)} \exp\left(-\frac{q^2\sigma_{ij}^2}{2}\right) \sum_L i^L \times j_L(qR_{ij}) Y_L^{M*}(\hat{R}_{ij}) \sum_n q^n \sum_{\lambda,\mu} \mathcal{W}^n_{ij,\lambda\mu} \mathcal{G}^{\ell m}_{\lambda\mu,LM}, \tag{58}$$

where we remind the reader that $M = m - \mu$.

The sum over ℓ formally runs from 0 to ∞, but we show in Fig. 3 that this sum can be safely truncated at finite $\ell_{\max}$. For relatively simple molecules such as benzene (C_6H_6) and p-xylene (C_8H_{10}), the sum over ℓ converges by $\ell_{\max} = 12$. Larger molecules, e.g., anthracene ($C_{14}H_{10}$), may require $\ell_{\max}$ in the range of 18–24, particularly at high momenta ($q \gg \alpha m_e$). As a 1D proxy for the 3D form factor, Fig. 3 shows the isotropic angular average, defined

$$\langle f_s^2 \rangle_\Omega \equiv \int \frac{d\Omega}{4\pi} |f_s(\mathbf{q})|^2. \tag{59}$$

From the orthogonality properties of spherical harmonics, it can be shown that

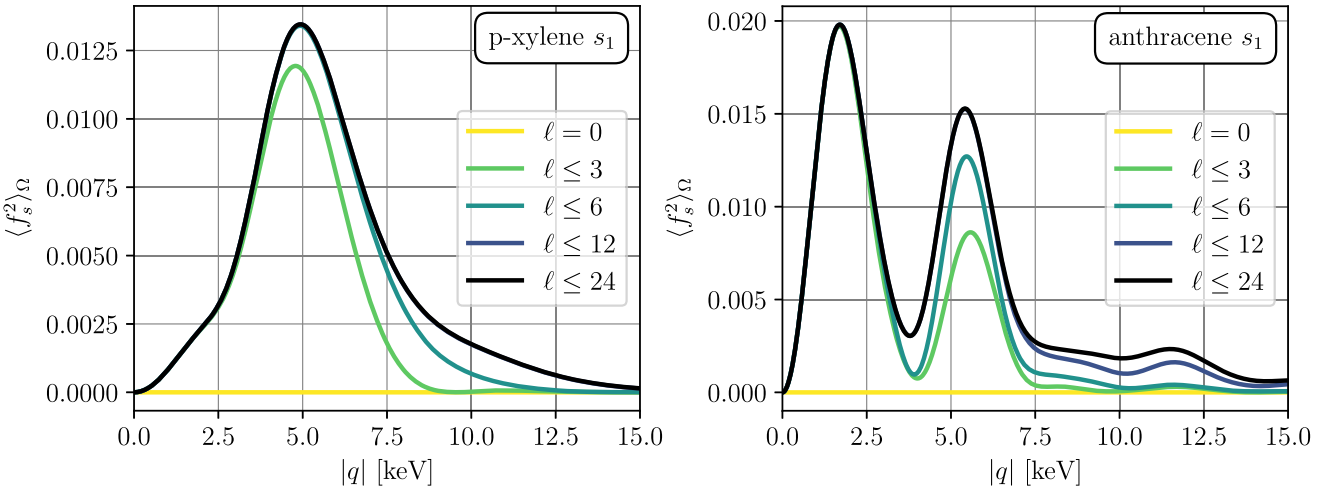


FIG. 3. We show the isotropic average form factor $(4\pi)^{-1} \int d\Omega |f_s|^2$ for the first excited states of p-xylene and anthracene, calculated with the spherical analytic method of Sec. V D. This f_s is found by a sum over spherical harmonic modes $f_s(\mathbf{q}) = \sum_{\ell m} Y_l^m(\hat{q}) \mathcal{R}_{lm}(q)$, which we truncate at finite ℓ. We find that the first excited state of p-xylene is very well described by the $\ell \leq 12$ angular modes, while an accurate representation of anthracene above $q > 7$ keV requires the $\ell > 12$ modes.

$$\langle f_s^2 \rangle_\Omega = \frac{1}{4\pi} \sum_{\ell m} |\mathcal{R}_{\ell m}(q)|^2. \tag{60}$$

In an isotropic medium such as a fluid or glass, the scattering rate depends only on $\langle f_s^2(\mathbf{q}) \rangle_\Omega$, so it is quite convenient that it can be extracted directly from the $\mathcal{R}_{\ell m}^2$ sum.

A. Summary

We implement Eq. (57) within SCarFFF to tabulate the form factor on a uniform spherical grid. This has advantages over the other two methods, as it is far more robust against numerical issues than the FFT method, but far faster to materialize the full 3D form factor grid than the Cartesian method, at the expense of a more difficult derivation. We will now move on to discuss the numerical implementations of each method in detail.

V. MOLECULAR FORM FACTORS WITH SCarFFF

Our numerical package, SCarFFF, implements three methods for the fast and precise computation of molecular form factors: tabulation on a spherical grid, on a Cartesian grid, and using the fast Fourier transform. Here we describe the numerical implementation of each method, as well as its advantages and disadvantages with respect to the other methods. For the impatient, we give an at-a-glance overview of the numerical package in Fig. 4.

A. Molecular geometry, DFT, and TD-DFT

All three methods begin with a computation of the molecular geometry, followed by a ground-state DFT calculation. The resulting ground state is then used as an input to a TD-DFT calculation of the excited-state transition energies and transition density matrices.

The molecular geometry optimization proceeds in several steps. First, we use the RDKit [20] implementation of the ETKDGv3 algorithm to generate several conformations of the molecule, which we subsequently optimize using the universal force field (UFF) method. We then take the lowest energy UFF optimized geometry and optionally pass this to PySCF [21] and geomeTRIC [22] to optimize further using DFT. This optional last step is significantly slower than the preceding optimization steps and offers modest corrections of $\mathcal{O}(0.1\%)$ to the distance matrix entries of small molecules, such as benzene, up to $\mathcal{O}(0.5\%–3\%)$ for intermediate-sized molecules, such as *trans*-stilbene, with comparable corrections to the excited-state energies in both cases. However, these small corrections to the geometry lead to large corrections to oscillator strength, on average $\mathcal{O}(30\%)$ for the first ten excited states of *trans*-stilbene, with a maximum of $\mathcal{O}(70\%)$, due to better encoding of molecular symmetries.

To compute the ground- and excited-state energies, we use TD-DFT within PySCF with one of the SCarFFF-supported basis sets. SCarFFF does not depend on the choice exchange-correlation functional, and so in theory any can be used. Throughout this work, we have used B3LYP, although we note that more modern and precise functionals such as, e.g., ωB97M-V may be more appropriate for certain molecules. The excited-state energies and atomic coordinates, along with the basis-dependent transition matrices in the atomic orbital basis are then saved to disk in HDF5 format. We also save the computed oscillator strengths for future verification of our form factor results, which we will discuss in Sec. VI. The calls to PySCF, including the optional ones to optimize the geometry using DFT, are by far the slowest step in the form factor pipeline. For this reason, we make use of GPU4PySCF, which can be toggled on with an optional `use_gpu` flag in the provided scripts.

B. Fast Fourier transform method

We now move on to discussing the bulk of the SCarFFF package, beginning with the conceptually simplest approach, the FFT method. This approach directly implements Eq. (19) as the Fourier transform of the transition density $\Phi^*_{g\to s}(\mathbf{x})$, with s denoting the sth excited state.

The first step in the form factor computation is common to all methods. The results from the TD-DFT and geometry optimization, along with the set of prenormalized, according to Eq. (A4), GTO scale factors and Gaussian widths are read in for the specified basis set. These are used to construct a hierarchical data structure composed of one-dimensional arrays defining the molecule, with index mappings from Cartesian term (e.g., the x^2 in the $3d_{x^2-y^2}$ orbital), to GTO primitive, to orbital in a given atom, and finally to atom within the molecule. This structure saves on both memory and computation time by storing intermediate objects in terms of the largest object in the hierarchy, while still allowing us to efficiently sum over the smallest.

Specific to the FFT method, we then construct a spatial grid on which to tabulate the transition density, Eq. (19). This spatial grid covers all $x \in [-x_{\rm lim}, x_{\rm lim}]$, with resolution Δx defined, in turn, by

$$x_{\rm lim} = \frac{\pi}{\Delta q_x}, \qquad \Delta x = \frac{2\pi}{q_{x,\rm lim}}, \tag{61}$$

with the user-specified momentum grid resolution and maximum, Δq and $q_{\rm lim}$, respectively. A similar grid is constructed in the y and z directions. Next, the value of the N_o orbitals in the molecule are computed at each grid point, where they are contracted with the full set of transition density matrices to form the transition density function $\Phi_{g\to s}(\mathbf{r})$. This process features two computationally expensive steps: the tabulation of the orbital values and the matrix multiplication. These computational costs scale as

$$c_{\text{orb}} \simeq k_1 N_p, \qquad c_{\text{matmul}} \simeq k_2 N_t N_o^2, \tag{62}$$

where N_p is the total number of primitives across all orbitals in the molecule, typically around $4N_o$ for a small basis set such as 6-31g*, N_t is the number of transitions computed per molecule, and $k_1 \simeq 100$ and $k_2 \simeq 2$ some constant prefactors. The cost ratio is therefore approximately

$$\frac{c_{\text{orb}}}{c_{\text{matmul}}} \simeq \frac{200}{N_o N_t}, \tag{63}$$

which for a typical molecule with a few hundred orbitals scales roughly as $1/N_t$ and is therefore dominated by the matrix multiplication for all but single transition computations. To alleviate this cost, we implement GPU acceleration of the transition density tabulation via custom CUDA kernels, which significantly speeds up the computation at the small expense of a few data transfers. This speedup is particularly noticeable in the matrix multiplication step, which benefits greatly from parallelized GPU operations.

As an additional efficiency measure, we also implement thresholding of small values during the transition density tabulation. This simply sets any GTO primitive whose Gaussian exponent would be sufficiently large, such that the primitive value drops below the threshold value, to zero. This avoids the expensive exponential evaluation that dominates the cost of k_1. As an additional measure, if all orbitals at a grid point would be zero, we simply skip the expensive matrix multiplication for that point. This last step makes a particularly large difference on spatial grids that extend far beyond the molecule, which as discussed in Sec. III B are necessary for good momentum resolution of the form factor.

The final step is to perform the FFT, which is done using the JULIA wrapper of FFTW [19]. As this step scales approximately linearly in the total number of grid points, it is far faster than the preceding steps. The issue, however, as compared to the analytic methods, is that the FFT step requires the materialization of both the real space transition grids as well as the momentum space form factor grids. This leads to an additional 50% memory overhead, potentially limiting the size of the grids that can be constructed with this method. The FFT method, while fast, can also suffer from numerical artifacts due to insufficient real space resolution and grid extent. A real space grid that is too coarse fails to resolve the peaks at each atomic site. On the other frontier, due to the FFT algorithm assuming periodic functions, a grid sampled only where the molecular wave function is large will experience errors due to unphysical tiling of the molecule. These can issues can, in turn, be alleviated by increasing q_{lim} and decreasing Δq, at a cost in both memory and computation.

We show the timing of the FFT method on the test molecule set in Fig. 5, both with and without GPU acceleration, in Table I. By comparison with the other two methods, we see that, especially on the CPU path, the FFT method is by far the fastest implementation, and that the GPU speeds up the form factor tabulation by an average factor of 2, irrespective of grid size. It would seem, therefore, that the FFT method is the best choice when running on systems without GPU access, especially when memory is less of an issue, to get a good estimate for molecular form factors. We stress, however, that the accuracy of the 100^3 grid point FFT is not comparable in accuracy to the same grid using either of the other two methods. For reference, the three plots in Fig. 2 used grids of approximately 136^3, 342^3, and 512^3 many points, and only the latter two examples produced approximately correct values of $f_s(\mathbf{q})$.

For this reason, the FFT method may be best suited for fast searches through large numbers of excited states, where qualitative accuracy is sufficient. Quantitative accuracy from the FFT generally requires a larger grid and correspondingly slower evaluation times. We will discuss the various verification methods implemented in SCarFFF that can be used to alleviate, or at least estimate the magnitude of, these inaccuracies in Sec. VI.

C. Cartesian grid method

We now turn our discussion to the Cartesian method. This implements Eq. (45) to tabulate the form factor on a uniform Cartesian grid in (q_x, q_y, q_z). As in Sec. V B, the first step is to create the hierarchical data structure defining the molecule and its transitions. From here, the Cartesian method diverges from the FFT method by evaluating the Fourier integral analytically and exploiting the separability in each of the momentum directions.

The next step is precomputing the small matrices of Cartesian pair coefficients defined in Eqs. (43), (38), (39), and (41), which are reused over many computations and, owing to their small size, can be safely stored in memory. Given these, SCarFFF then constructs the one-dimensional $\mathcal{V}$ tensors defined in Eq. (46). To speed up the implementation, the Hermite polynomials are generated using recurrence relations up to maximum Cartesian order α_{ij} for each pair.

The information contained in the $\mathcal{V}$ tensors, along with the small array of pair coefficients M_{ij} and transition density matrices are sufficient to completely define the form factor. This is particularly useful when the form factor is to be used in separable integrals of the form

$$I_{\text{sep}} = \int d^3q f(q_x) g(q_y) h(q_z) f_s(\mathbf{q}), \tag{64}$$

where f, g, and h are some arbitrary functions each depending on, at most, one coordinate. Using the

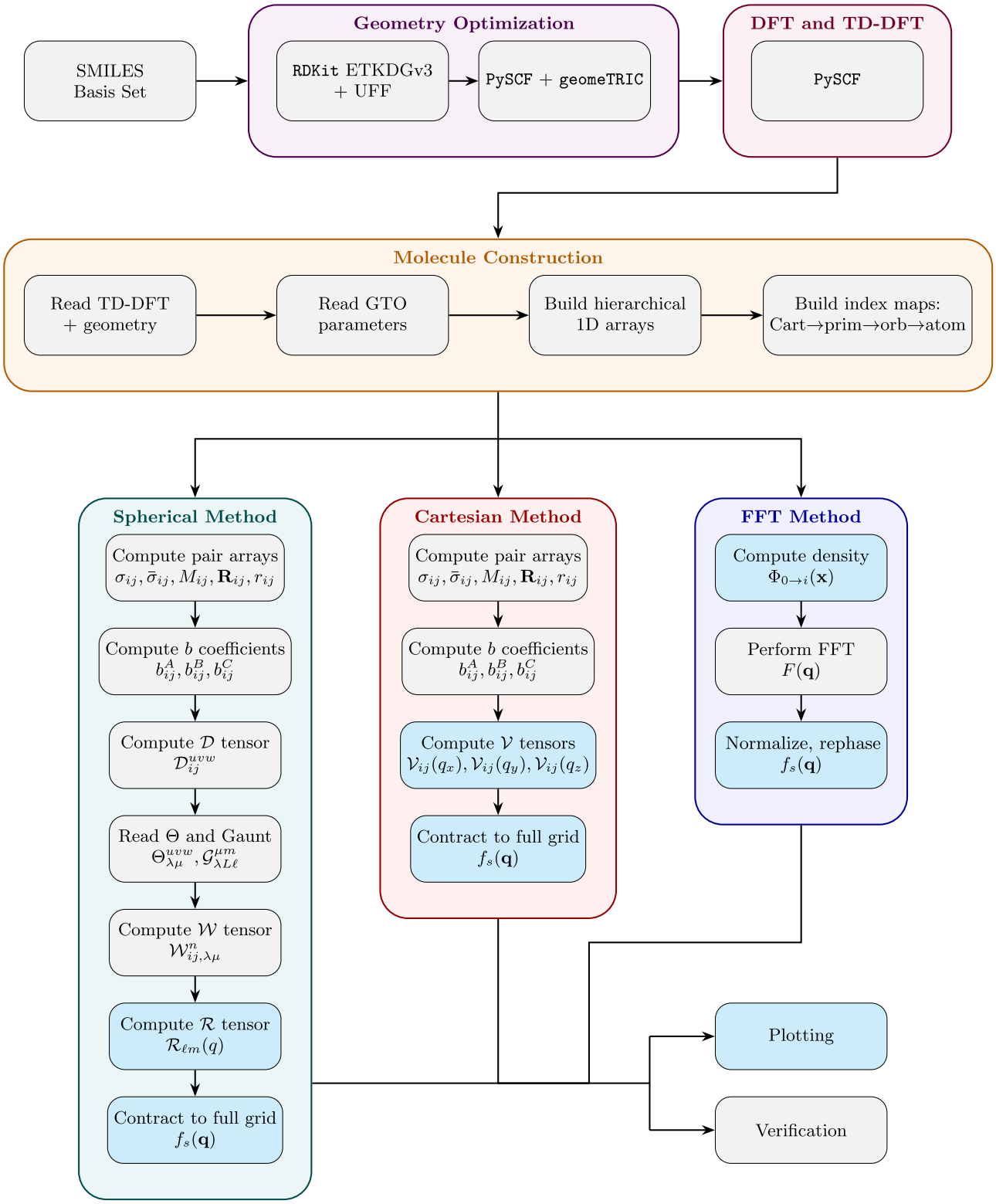


FIG. 4. Workflow of SCarFFF. Given a simplified molecular input line entry system string and a basis set, we first optimize the geometry and perform the DFT and TD-DFT computations. We then build data structures defining the molecule and pass it through one of the three methods. See the text for more details. Boxes highlighted in blue represent outputs of the code.

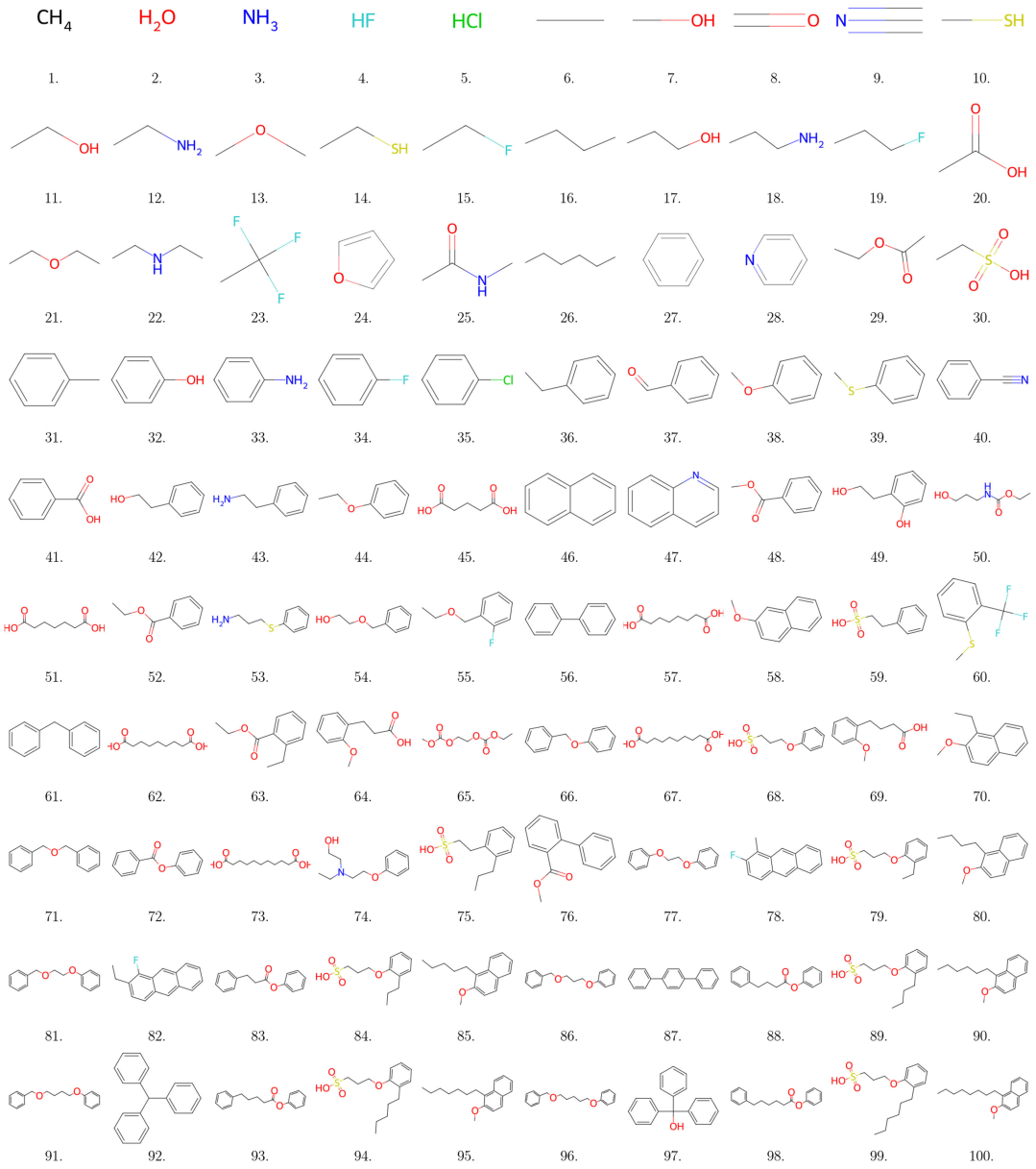


FIG. 5. The molecule set used to test SCarFFF. This test set is composed of 100 molecules composed of hydrogen, carbon, nitrogen, sulphur, and fluorine, with an exactly uniformly distributed heavy (nonhydrogen) atom count from 1 to 20. These are chosen to be a combination of long chain and aromatic compounds, so as to represent a wide spectrum of molecules.

separability of the Cartesian form factor, this can be rewritten in the form

$$I_{\text{sep}} = \sum_{i,j} T_{ij}^{(s)} M_{ij} \int dq_x f(q_x) \mathcal{V}_{ij}(q_x) \int \ldots \quad (65)$$

such that the cost of the integration scales linearly with the most densely sampled grid dimension, as opposed to the product of all three in the case of the FFT form factor. Importantly, each of these $\mathcal{V}$ tensors be can constructed in approximately

$$c_{\mathcal{V}} = k_{\mathcal{V}} N_c^2 N_{q_i} \quad (66)$$

operations, with $i \in \{x, y, z\}$, $N_c \simeq N_p \simeq 4N_o$ the total number of Cartesian terms, and $k_{\mathcal{V}} \simeq 300$. Owing to $k_{\mathcal{V}}$, the construction of the $\mathcal{V}$ tensors will, in fact, dominate over the integration time, and we can directly compare to the equivalent cost for the FFT method

$$\frac{c_{\mathcal{V}}}{c_{\text{FFT}}} \simeq \frac{2400}{N_{q_i}^2}, \quad (67)$$

where we assume N_{q_i} is approximately constant for all dimensions and approximate the time to construct the FFT

form factor by the matrix multiplication time, Eq. (62). Even for small grids with $N_{q_i} \simeq 100$, this method is therefore far better suited to form factor applications involving separable integrals than the FFT method. To facilitate this application, SCarFFF comes with an option to only compute and save the 1D $\mathcal{V}$ tensors when using the Cartesian method. Much like the FFT method, we also apply thresholding in the Cartesian method by discarding all Cartesian pair index combinations for which $|M_{ij}T_{ij}^{(s)}/(M_{ij}T_{ij}^{(s)})_{\max}| < \epsilon$ on a transition-by-transition basis, with ϵ the user-specified threshold parameter. We find that, for a threshold of 10^{-6}, only around 25% of terms survive for most molecules, speeding up computation by a factor of 4 with no measurable impact on the accuracy of the final result. With that said, we highly recommend that the user explores different values of ϵ for convergence, especially where high accuracy is required.

Where this method struggles is when the full 3D grid needs to be materialized, as this requires the outer product of the $\mathcal{V}$ tensors, followed by a contraction of the resulting object over the Cartesian term indices. The cost, per grid point, of constructing the full form factor (neglecting thresholding) is then

$$c_{\text{Cart}} \simeq k_3 N_t N_c^2, \tag{68}$$

with $k_3 \simeq 8$. Comparing this to the FFT method, we find

$$\frac{c_{\text{Cart}}}{c_{\text{FFT}}} \simeq 64, \tag{69}$$

almost 2 orders of magnitude slower. However, owing to its simplicity and robustness to numerical artifacts, this method is particularly useful as a verification tool for the other two methods. This can be done by, e.g., computing planes of particular interest or using the 1D $\mathcal{V}$ tensors to compute particular points on the full 3D grid.

To improve the usefulness of this method when the full grid is required, we add a GPU path for the final grid contraction using cuBLAS. As shown in Table I, this speeds up computation by an extraordinary factor of ~50–100, depending on the grid size. This makes it highly competitive with the FFT method in terms of speed, without suffering from the same numerical issues. Outside of the GPU path, however, we only recommend that the Cartesian method is only used for verification or for very small grids with $N_{q_i} \lesssim 50$.

D. Spherical grid method

The final method implemented by SCarFFF is the spherical grid method, which tabulates the form factor (57) on a uniform grid in (q, q_θ, q_ϕ), where $q = |\mathbf{q}|$, up to a user-specified $\ell_{\max}$.

This implementation follows the Cartesian method exactly up to the construction of the Cartesian pair coefficient matrices. From there, we construct the $\mathcal{D}$ tensor defined in Eq. (49). Naively, we would need to allocate a full $N_c^2 \times u_{\max}^3$ object to store the $\mathcal{D}$ tensor, where $u_{\max}$ is twice the maximum angular momentum in the basis set. However, as many Cartesian pairs have, e.g., $\alpha_{ij} < u_{\max}$, a significant fraction of this dense tensor would be zero. Such a dense tensor would also be inefficient due the symmetry of the $\mathcal{D}$ tensor under $i \leftrightarrow j$. As such, we store the $\mathcal{D}$ tensor in sparse coordinate (COO) format, which for a typical molecule reduces memory requirements by around 2 orders of magnitude. As an additional benefit, when constructing objects that depend on $\mathcal{D}$, we now only need to iterate over the nonzero entries, which speeds up computation by a comparable amount.

The next step is to compute the Θ coefficients defined in Appendix B, along with the Gaunt coefficients. As these coefficients are independent of the molecule and transition, they are precomputed once for a given $\ell_{\max}$ and $\lambda_{\max}$, and then reused across runs. Much like the $\mathcal{D}$ tensor, these would also be incredibly sparse objects if computed in dense form. As such, we exploit the many symmetries and selection rules for these coefficients to compute and store them in sparse COO format.

These are then contracted with the previously computed $\mathcal{D}$ tensor to form the $\mathcal{W}$ tensor defined in Eq. (53), which, once again, is stored in sparse COO format. Owing to the sparse storage, symmetry exploitation, and selection rules, all steps to this point are completed almost instantly. The next step, however, applying Eq. (58) to construct the $\mathcal{R}$

TABLE I. Run-time for the different methods, excluding molecular geometry optimization and DFT, on the 100 molecule test set, shown in Fig. 5, on grids with N_{q_i} points in each direction. For all methods, we use the cc-pVDZ basis set, a threshold of 10^{-6}, and compute the form factor for the first 12 excited states up to a maximum of $|\mathbf{q}| = 15$ keV. For the spherical method, we compute up to angular mode $\ell_{\max} = 24$. For both the pure CPU- and GPU-accelerated runs, we run SCarFFF on 24 cores of an Intel Xeon Platinum 8468 CPU, while the GPU acceleration is performed with an NVIDIA H100 SXM GPU. Not included in these benchmarks is the time for the method-independent geometry optimization and TD-DFT computations, which took 4736 s on the same hardware with GPU acceleration.

Method	N_{q_i}	CPU (s)	GPU (s)	CPU/GPU
FFT	50	83	43	1.93×
	100	165	108	1.53×
	200	685	389	2.22×
Cartesian	50	1005	70	9.66×
	100	7339	166	44.21×
	200	48776	520	93.80×
Spherical	50	1779	334	5.33×
	100	3652	611	5.98×
	200	8996	1305	6.89×

tensor, is particularly expensive. We compute the spherical Bessel functions with a fast, custom implementation of Miller's downward recursion algorithm

$$j_{L-1}(x) = \left(\frac{2L+1}{x}\right) j_L(x) - j_{L+1}(x), \tag{70}$$

at each qR_{ij} point, giving us access to the full tower of Bessel functions in one sweep. For numerical stability, we always perform this step in FLOAT64 and expand compute the Bessel functions using a power series expansion for very small arguments. With this optimization, the cost of the spherical method is dominated by the accumulation into the $\mathcal{R}$ tensor and is given by

$$c_{\mathcal{R}} \simeq k_{\mathcal{R}} N_t N_q N_{\mathcal{W}} N_{\tilde{\mathcal{G}}}, \tag{71}$$

where $N_{\mathcal{W}} \simeq N_c^2$ is the number of nonzero $\mathcal{W}$-tensor entries, $N_{\tilde{\mathcal{G}}}$ is the average number of entries for each (λ, μ) combination, and $k_{\mathcal{R}} \simeq 8$. A rough estimate gives $N_{\tilde{\mathcal{G}}} \simeq (\lambda_{\max}+1)(\ell_{\max}+1)^2$. Much like the $\mathcal{V}$ tensors in the Cartesian method, the $\mathcal{R}$ tensor contains all of the molecule-specific information about the form factor. As such, it may be useful to terminate the computation early when the form factor is required for integrals where the angular and radial components are separable. We therefore provide an option within SCarFFF to compute only the $\mathcal{R}$ tensor, without materializing the full 3D form factor grid.

Otherwise, the final step is to contract with the angular grid according to Eq. (57), the cost for which is dominated by one large outer product, followed by a contraction over ℓ and m. The cost for this step is roughly

$$c_{\text{contract}} \simeq k_4 N_t N_q N_{q_\theta} N_{q_\phi} (\ell_{\max}+1)^2, \tag{72}$$

with $k_4 \simeq 2$. Whether this step dominates therefore depends heavily on the size of the angular grid. The cost ratio of this step to that of the $\mathcal{R}$-tensor materialization is

$$\frac{c_{\mathcal{R}}}{c_{\text{contract}}} \simeq \frac{4N_c^2(\lambda_{\max}+1)}{N_{q_\theta} N_{q_\phi}}, \tag{73}$$

which for a modest grid with $N_{q_\theta} = N_{q_\phi} \simeq 100$, and a typical molecule with $N_c^2 \simeq 10^5$, $\lambda_{\max} \simeq 6$, is dominated by the $\mathcal{R}$-tensor construction. Even for a larger grid with 10^3 points in each direction, we find that the $\mathcal{R}$-tensor step has a comparable cost to the contraction step.

This allows us to directly compare the spherical method to the other two methods as

$$\frac{c_{\text{spher}}}{c_{\text{FFT}}} \simeq \frac{64(\lambda_{\max}+1)(\ell_{\max}+1)^2}{N_{q_i}^2}, \tag{74}$$

and similar for the Cartesian method, with a ratio 64 times smaller, where we have assumed a similar number of grid points in each direction, for each method. For small grids with $N_{q_i} \simeq 100$, and $\ell_{\max} = 18$–24, the spherical method is approximately an order of magnitude slower than the FFT method. However, unlike the FFT method, the spherical method is far less susceptible to numerical errors and scales far more favorably with the grid size due to the decoupling of the angular and radial parts.

As with the other two methods, we also offer GPU paths for the spherical method to speed up computation. This includes custom CUDA kernels for the spherical Bessel function, spherical harmonic, and $\mathcal{R}$-tensor accumulation, along with cuBLAS for the final grid contraction. As shown in Table I, this typically speeds up execution by a factor of ~5–6, making the spherical method competitive with the FFT method in terms of speed, but with far fewer numerical challenges.

VI. VERIFICATION

To verify that SCarFFF gives the correct results, we apply two key checks. The first, and simplest, is exclusive to the FFT method. As we are performing a Fourier transform, and have access to both the transition density and form factor, a natural check is that Parseval's theorem holds. That is, the transition density and form factor should satisfy the relation

$$\int d^3r |\Phi^{g\to s}(\mathbf{r})|^2 = \int \frac{d^3q}{(2\pi)^3} |f_s(\mathbf{q})|^2. \tag{75}$$

This check can be toggled with the `check_parseval` flag when running in FFT mode. Ensuring that this equality holds can help to safeguard against numerical artifacts coming from grids with insufficient extent or resolution.

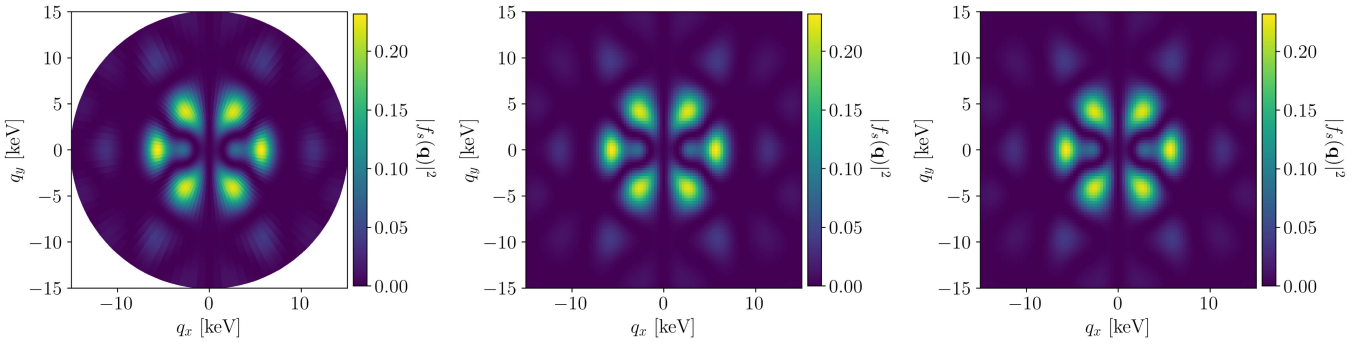
The second check is the oscillator strength reconstruction, which is available to all methods. This verification strategy checks that the oscillator strength computed from the tabulated form factor matches the PySCF output, which is computed reliably using TD-DFT. To compute the oscillator strength, we make use of the low momentum expansion of the form factor

$$f_s(\mathbf{q}) \equiv \langle \Psi_s | e^{i\mathbf{q}\cdot\mathbf{r}} | \Psi_g \rangle \simeq i\mathbf{q}\cdot\vec{\mu} + \mathcal{O}(|\mathbf{q}|^2), \tag{76}$$

up to spin factors, with $\vec{\mu} \equiv \langle \Psi_s | \mathbf{r} | \Psi_g \rangle$ the transition dipole moment, which in the basis of molecular orbitals φ_p is given by

$$\vec{\mu} = \sum_{pq} T_{pq}^{(s)} \langle \varphi_q | \mathbf{r} | \varphi_p \rangle. \tag{77}$$

This is related to the oscillator strength f_0 by



(a) Spherical (b) Cartesian (c) FFT

FIG. 6. The squared form factor $|f_s(\mathbf{q})|^2$ for the second excited-state transition of p-xylene, using (a) the spherical method with $\ell_{\max} = 24$, (b) the Cartesian method, and (c) the FFT method. We use the cc-pVDZ basis set, B3LYP exchange-correlation functional, and threshold $\epsilon = 10^{-6}$ for all three computations, and $N_{q_i} = 100$ grid points in each direction for both the spherical and Cartesian methods. To achieve the same accuracy with the FFT method, we have scaled up to $N_{q_i} = 400$. At $N_{q_i} = 100$ grid points, the FFT method disagrees by ∼10% with the analytic integration methods.

$$f_0 = \frac{2}{3} m_e \Delta E_{fi} |\vec{\mu}|^2, \tag{78}$$

where ΔE_s is the transition energy.

For the real-valued transition density matrix, $\vec{\mu}$ can be extracted from the Taylor expansion of $\mathrm{Im}(f_s)$,

$$\mathrm{Im}\, f_s(\mathbf{q}) \simeq 0 + \mathbf{q} \cdot \nabla_q \,\mathrm{Im}\, f_s(\mathbf{q})|_{q\to 0} + \ldots \tag{79}$$

$$\vec{\mu} = \nabla_q \,\mathrm{Im}\, f_s(\mathbf{q})|_{q\to 0} \tag{80}$$

or, equivalently, from the quadratic moment of the squared form factor

$$f_0 = 2 m_e \Delta E_{fi} \lim_{|\mathbf{q}|\to 0} \frac{\langle |f_s(\mathbf{q})|^2 \rangle_\Omega}{|\mathbf{q}|^2}. \tag{81}$$

In SCarFFF we perform a fit to the angular average of the squared form factor up to some user-defined $|\mathbf{q}|_{\max} \simeq 0.05$–$0.25$ keV,

$$\langle |f_s(\mathbf{q})|^2 \rangle_\Omega \simeq A + B|\mathbf{q}|^2 + C|\mathbf{q}|^4, \tag{82}$$

and extract the coefficient $B = |\vec{\mu}|^2/3$. The agreement is naturally better for lower $|\mathbf{q}|_{\max}$, but is not always feasible due to coarse grids, particularly for the FFT. The first coefficient, A, is added to the fit to subtract any numerical artifacts of nonzero form factor at $\mathbf{q} = \mathbf{0}$, while C accounts for the fact that we do not exactly take the $|\mathbf{q}| \to 0$ limit, but instead expand at finite, but small momenta. We find that for typical run parameters, e.g., a threshold of 10^{-6}, $\ell_{\max} = 24$ (if using the spherical method), and a 100^3 $\mathbf{q}$ point grid, that our oscillator strengths match those of PySCF to the 10^{-4} level or better, provided $|\mathbf{q}|_{\max} = 0.05$ keV.

In Fig. 6(c) we show the form factor for the second excited-state p-xylene computed using SCarFFF, using each of the three methods. This demonstrates the self-consistency of our package, i.e., that the results from all of our methods agree. We draw particular attention to the FFT method, however, which had to be computed on a grid with 25 keV extent and a resolution of 0.125 keV to converge to the same result as the analytic methods, for a total of $4^3 = 64$ times more grid points.

VII. CONCLUSIONS

SCarFFF offers three complementary pathways for the speedy evaluation of molecular form factors. The Cartesian and spherical analytic methods can both be run without loss of precision, generating reliable results in a fraction of the time spent on the physical chemistry calculation. When speed rather than precision is the priority, in particular, for CPU-only runs, the FFT-based method provides the fastest analysis. Our numerical method SCarFFF is publicly available as a package in JULIA, and can be found at GitHub [23].

Although SCarFFF uses PySCF for its TD-DFT calculation, the analytic method we present in this paper can be used more generally, for any system where the molecular geometry and the GTO-basis transition density matrix are known. This makes our methods highly adaptable and compatible, in principle, with any other electronic structure computation. The present version of SCarFFF captures the spin-independent molecular form factor for electronic transitions between spin-singlet states, but we intend to broaden our scope to include both types of spin dependence in future work.

ACKNOWLEDGMENTS

We would like to thank Juri Smirnov, Cameron Cook, Yonatan Kahn, Louis Hamaide, and Robert McGehee for helpful conversations. J. D. S. thanks Clare Burrage and Jonas Spinner for helpful discussions toward the end of this project.

The work of B. L. was supported in part by the U.S. Department of Energy under Grant No. DE-SC0011640. J. D. S. is supported through funding from the UK Research and Innovation Future Leader Fellowship No. MR/Y018656/1.

DATA AVAILABILITY

The data that support the findings of this article are openly available at GitHub [23].

APPENDIX A: ATOMIC ORBITAL DEFINITIONS

Each of our atomic orbitals is written as a sum of primitive basis functions as

$$\phi_\alpha(\mathbf{r}) = N_\alpha \sum_\mu \chi_\mu(\mathbf{r}), \tag{A1}$$

where $\alpha, \beta\ldots$ are used to denote orbital-specific quantities, $\mu, \nu\ldots$ are to denote primitive specific quantities, and N_α is a normalization factor. Each of these primitives is itself a sum of Cartesian terms

$$\chi_\mu(\mathbf{r}) = d_\mu \xi_\mu \exp\left(-\frac{r^2}{2\sigma_\mu^2}\right) \sum_{i\in\mu} k_{\mu i} x^{a_{\mu i}} y^{b_{\mu i}} z^{c_{\mu i}}, \tag{A2}$$

with σ_μ the Gaussian width of the primitive, and d_μ an overall scaling coefficient. Each choice of basis set (e.g., 6-31g* or cc-pVDZ) is defined by its values of d_μ and σ_μ. The coefficients k_i are the Cartesian prefactors, which could be, e.g., -1 for the y^2 term in the $3d_{x^2-y^2}$ orbital. Similarly, a_i, b_i, and c_i denote the degree of x, y, and z for orbitals, respectively. We will denote objects at the Cartesian term level with the indices i and j throughout.

The normalization coefficients are fixed such that individual primitives are normalized to

$$\int d^3r |\chi_\mu(\mathbf{r})|^2 = d_\mu^2, \tag{A3}$$

while contracted orbitals are normalized to unity. This amounts to

$$\xi_\mu = \left[\sum_{i,j\in\mu} k_{\mu i} k_{\mu j} \prod_{K\in\{\alpha,\beta,\gamma\}} \sigma_\mu^{1+K_{ij}^{\mu\mu}} \Gamma\left(\frac{K_{ij}^{\mu\mu}+1}{2}\right)\right]^{-\frac{1}{2}}, \tag{A4}$$

$$N_\alpha = \left[\sum_{\mu,\nu} d_\mu d_\nu \xi_\mu \xi_\nu \sum_{i\in\mu, j\in\nu} k_{\mu i} k_{\nu j} \prod_{K\in\{\alpha,\beta,\gamma\}} (\sqrt{2}\sigma_{\mu\nu})^{1+K_{ij}^{\mu\nu}} \Gamma\left(\frac{K_{ij}^{\mu\nu}+1}{2}\right)\right]^{-\frac{1}{2}}, \tag{A5}$$

with $\alpha_{ij}^{\mu\nu} = a_{\mu i} + a_{\nu j}$, $\beta_{ij}^{\mu\nu} = b_{\mu i} + b_{\nu j}$, $\gamma_{ij}^{\mu\nu} = c_{\mu i} + c_{\nu j}$, and $\sigma_{\mu\nu}$ as defined in Eq. (38).

As a concrete example, we take a p_z orbital, which has just a single Cartesian term with $(a, b, c, k) = (0, 0, 1, 1)$, i.e.,

$$\chi_\mu(\mathbf{r}) = d_\mu \xi_\mu z \exp\left(-\frac{r^2}{2\sigma_\mu^2}\right), \tag{A6}$$

such that

$$\xi_\mu = \left(\frac{1}{2}\pi^{\frac{3}{2}}\sigma_\mu^5\right)^{-\frac{1}{2}}. \tag{A7}$$

In a toy orbital with two primitives, labeled with $\mu \in \{0, 1\}$, this would then have an overall normalization

$$N_\alpha = \left[d_0^2 + d_1^2 + 2d_0 d_1 \left(\frac{2\sigma_0\sigma_1}{\sigma_0^2+\sigma_1^2}\right)^{\frac{5}{2}}\right]^{-\frac{1}{2}}. \tag{A8}$$

APPENDIX B: MONOMIAL-TO-SPHERICAL COEFFICIENTS

The key feature that sets the spherical grid method above the Cartesian grid method is the ability to decouple the $|\mathbf{q}|$ grid computation from the angular (q_θ, q_ϕ) grid computation. However, in order to go from a Cartesian to spherical grid, we need to map monomials in q_x, q_y, and q_z onto a basis of spherical harmonics. That is, we require the coefficients $\Theta_{\lambda\mu}^{uvw}$ satisfying

$$q_x^u q_y^v q_z^z = |\mathbf{q}|^n \sum_{\lambda,\mu} \Theta_{\lambda\mu}^{uvw} Y_\lambda^\mu(\hat{q}), \tag{B1}$$

where $n = u + v + w$. The first step toward this goal is to write the Cartesian momentum components in terms of the $\ell = 1$ spherical harmonics as

$$q_i = |\mathbf{q}| \sqrt{\frac{4\pi}{3}} \sum_m c_{im} Y_1^m(\hat{q}), \tag{B2}$$

where the coefficients follow from the transformation from Cartesian coordinates to the spherical basis

$$c_{xm} = \frac{1}{\sqrt{2}}\{1, 0, -1\}, \tag{B3}$$

$$c_{ym} = \frac{i}{\sqrt{2}}\{1, 0, 1\}, \tag{B4}$$

$$c_{zm} = \{0, 1, 0\}, \tag{B5}$$

with the entries corresponding to $m = \{-1, 0, 1\}$. This decomposition into spherical tensor operators reveals several properties of the Θ coefficients. First, as only q_x and q_y can raise or lower angular momenta, μ must have the same parity as $u + v$ and must also satisfy $|\mu| \leq u + v$. Second, as the coupling of n rank-1 spherical tensor operators can result in at most a rank-n tensor, we must have $\lambda \leq n$. Finally, and more subtly, we know that under spatial inversions, $\prod_{i=1}^{n} q_i \to (-1)^n \prod_{i=1}^{n} q_i$, while $Y_\lambda^\mu(\hat{q}) \to (-1)^\lambda Y_\lambda^\mu(\hat{q})$. As a result, only coefficients where λ and n have the same parity survive. The full set of selection rules is then

$$0 \leq \lambda \leq n, \qquad \lambda \equiv n \ (\mathrm{mod}\ 2), \tag{B6}$$

$$|\mu| \leq \min\{\lambda, u + v\}, \qquad \mu \equiv u + v \ (\mathrm{mod}\ 2), \tag{B7}$$

which can be leveraged to significantly speed up the computation of the Θ coefficients.

Using Eq. (B2), we can rewrite the monomial in q_i as

$$q_x^u q_y^v q_z^w = |\mathbf{q}|^n \left(\frac{4\pi}{3}\right)^{\frac{n}{2}} \sum_{\{m_i\}} \left[\left(\prod_{i=1}^{u} c_{xm_i} \right) \left(\prod_{i=u+1}^{u+v} c_{ym_i} \right) \left(\prod_{i=u+v+1}^{n} c_{zm_i} \right) \left(\prod_{i=1}^{n} Y_1^{m_i}(\hat{q}) \right) \right], \tag{B8}$$

where the sum runs over all combinations of n magnetic quantum numbers, $m_i \in \{-1, 0, 1\}$. The next step is to couple the $\ell = 1$ spherical harmonics to yield an expression of the form

$$\prod_{i=1}^{n} Y_1^{m_i}(\hat{q}) = \sum_\lambda \mathcal{K}_\lambda^n(\{m_i\}) Y_\lambda^{\mu_n}(\hat{q}), \tag{B9}$$

with $\mu_n = \sum_{i=1}^{n} m_i$, and $\lambda \in \{n, n-2, \ldots\}$, which follows from our earlier parity arguments. The form of the $\mathcal{K}$ coefficients is found by repeatedly applying the spherical harmonic contraction rule given in Eq. (55). This leads to the recursive definition of the $\mathcal{K}$ coefficients

$$\mathcal{K}_\lambda^1(\{m_i\}) = \delta_{\lambda 1}, \tag{B10}$$

$$\mathcal{K}_\lambda^n(\{m_i\}) = \sum_L \mathcal{G}^{\lambda\mu_n}_{L\mu_{n-1},1m_n} \mathcal{K}_L^{n-1}(\{m_i\}), \tag{B11}$$

where in the second line, $L \in \{n-1, n-3, \ldots\}$, and $m_n = \mu_n - \mu_{n-1}$. Applying Eq. (B9) and reordering the sums yields the desired form of the Θ coefficients

$$\Theta_{\lambda\mu}^{uvw} = \left(\frac{4\pi}{3}\right)^{\frac{n}{2}} \sum_{\{m_i\}} \left[\left(\prod_{i=1}^{u} c_{xm_i} \right) \left(\prod_{i=u+1}^{u+v} c_{ym_i} \right) \left(\prod_{i=u+v+1}^{n} c_{zm_i} \right) \mathcal{K}_\lambda^n(\{m_i\}) \delta_{\mu_n \mu} \right]. \tag{B12}$$

[1] F. Zwicky, Die Rotverschiebung von extragalaktischen Nebeln, Helv. Phys. Acta **6**, 110 (1933).

[2] Vera C. Rubin and W. Kent Ford, Jr., Rotation of the Andromeda Nebula from a spectroscopic survey of emission regions, Astrophys. J. **159**, 379 (1970).

[3] Benjamin Lillard, Partial rate matrix for dark matter scattering, Phys. Rev. Lett. **134**, 221003 (2025).

[4] Benjamin Lillard, Wavelet-harmonic integration methods, Phys. Rev. D **111**, 123006 (2025).

[5] Benjamin Lillard and Aria Radick, Vector Spaces for Dark Matter (VSDM): Fast direct detection calculations with Python and JULIA, SciPost Phys. Codebases **68**, 1 (2026).

[6] Tanner Trickle, Zhengkang Zhang, Kathryn M. Zurek, Katherine Inzani, and Sinéad M. Griffin, Multi-channel direct detection of light dark matter: Theoretical framework, J. High Energy Phys. 03 (2020) 036.

[7] Yonit Hochberg, Yonatan Kahn, Noah Kurinsky, Benjamin V. Lehmann, To Chin Yu, and Karl K. Berggren, Determining dark-matter–electron scattering rates from the dielectric function, Phys. Rev. Lett. **127**, 151802 (2021).

[8] Simon Knapen, Jonathan Kozaczuk, and Tongyan Lin, Dark matter-electron scattering in dielectrics, Phys. Rev. D **104**, 015031 (2021).

[9] Robert Lasenby and Anirudh Prabhu, Dark matter–electron scattering in materials: Sum rules and heterostructures, Phys. Rev. D **105**, 095009 (2022).
[10] Christian Boyd, Yonit Hochberg, Yonatan Kahn, Eric David Kramer, Noah Kurinsky, Benjamin V. Lehmann, and To Chin Yu, Directional detection of dark matter with anisotropic response functions, Phys. Rev. D **108**, 015015 (2023).
[11] Riccardo Catena and Nicola A. Spaldin, Linear response theory for light dark matter-electron scattering in materials, Phys. Rev. Res. **6**, 033230 (2024).
[12] Asher Berlin, Alexander J. Millar, Tanner Trickle, and Kevin Zhou, Determining spin-dependent light dark matter rates from neutron scattering, Phys. Rev. D **112**, 035021 (2025).
[13] Yonit Hochberg, Majed Khalaf, Alessandro Lenoci, and Rotem Ovadia, Determining (all) dark matter-electron scattering rates from material properties, arXiv:2510.25835.
[14] Pierce Giffin, Benjamin Lillard, Pankaj Munbodh, and Tien-Tien Yu, Simplified spin dependence in dark matter direct detection, Phys. Rev. D **113**, 115018 (2026).
[15] Rouven Essig, Marivi Fernandez-Serra, Jeremy Mardon, Adrian Soto, Tomer Volansky, and Tien-Tien Yu, Direct detection of sub-GeV dark matter with semiconductor targets, J. High Energy Phys. 05 (2016) 046.
[16] Carlos Blanco, Yonatan Kahn, Benjamin Lillard, and Samuel D. McDermott, Dark matter daily modulation with anisotropic organic crystals, Phys. Rev. D **104**, 036011 (2021).
[17] Mark E Casida, Time-dependent density functional response theory for molecules, in *Recent Advances in Density Functional Methods: (Part I)* (World Scientific, Singapore, 1995), pp. 155–192.
[18] Richard L. Martin, Natural transition orbitals, J. Chem. Phys. **118**, 4775 (2003).
[19] Matteo Frigo and Steven G. Johnson, The design and implementation of FFTW3, Proc. IEEE **93**, 216 (2005).
[20] Greg Landrum *et al.*, The RDKit documentation: An overview of the RDKit, https://www.rdkit.org/docs/Overview.html (2025), accessed: 2025-12-17.
[21] Qiming Sun *et al.*, Recent developments in the PySCF program package, J. Chem. Phys. **153**, 024109 (2020).
[22] Lee-Ping Wang and Chenchen Song, Geometry optimization made simple with translation and rotation coordinates, J. Chem. Phys. **144**, 214108 (2016).
[23] Jack D. Shergold, 10.5281/zenodo.23660912 (2026).