Efficient computation of the second-Born self-energy using tensor-contraction operations

Riku Tuovinen,1, a) Fabio Covito,1 and Michael A. Sentef1
Max Planck Institute for the Structure and Dynamics of Matter, Luruper Chaussee 149, 22761 Hamburg, Germany
(Dated: 15 October 2019)

In the nonequilibrium Green’s function approach, the approximation of the correlation self-energy at the second-Born level is of particular interest, since it allows for a maximal speed-up in computational scaling when used together with the Generalized Kadanoff-Baym Ansatz for the Green’s function. The present day numerical time-propagation algorithms for the Green’s function are able to tackle first principles simulations of atoms and molecules, but they are limited to relatively small systems due to unfavourable scaling of self-energy diagrams with respect to the basis size. We propose an efficient computation of the self-energy diagrams by using tensor-contraction operations to transform the internal summations into functions of external low-level linear algebra libraries. We discuss the achieved computational speed-up in transient electron dynamics in selected molecular systems.

PACS numbers: 31.15.-p, 31.25.-v, 71.10.-w

I. INTRODUCTION

A state-of-the-art computational method for out-of-equilibrium many-body physics is the nonequilibrium Green’s function (NEGF) approach1–6. Mostly due to lack of computational capabilities, the non-linear integro-differential Kadanoff-Baym equations (KBE) for the NEGF from the 1960s remained fairly elusive until their first numerical solutions were presented in 1984 by Danielewicz7 and further numerical implementations at the turn of the century8–10. During the past twenty years a considerable amount of progress has been achieved in various fields of physics employing the NEGF approach: From sub-atomic nuclear reactions11,12 to atomic and molecular scale13–22, further to condensed phase23–33 and mesoscopic systems34–39, and even to descriptions of high-energy particle physics in cosmology40–42. However, combining the KBE with ab initio descriptions of realistic materials still remains a computational challenge. This challenge results from the double-time structure of the KBE rendering the method very expensive for both CPU time and storing the objects in RAM. The Generalized Kadanoff–Baym Ansatz (GKBA) offers a simplification by reducing the two-time-propagation of the Green’s function to the time-propagation of a time-local density matrix43. The computational complexity of the time-propagation of the GKBA equations scales as the number of time steps squared instead of the cubic scaling in the double-time KBE44. When a simulation to reach longer time scales is desired, this difference in computational speed becomes immense. However, this speed-up in computational scaling is only possible for the correlation self-energy approximation at the second-Born (2B) level. The 2B approximation goes beyond mean-field description at the Hartree-Fock (HF) level but it includes the bare interaction only up to second order, i.e., higher order correlations and screening effects are neglected, like in higher order T-matrix or GW approximations45,46. However, the viability of the 2B approximation has been assessed for a large set of systems with up to moderate interaction strength47,48.

Even though the above implementations of the NEGF method have been successfully applied in many contexts, the computation of the self-energy still remains a numerical bottleneck. For larger systems to be studied, the scaling with respect to the basis size in the self-energy diagrams may be very unfavourable, making first principles simulations numerically expensive, at least in naive implementations when looping over the full basis. Recently, a dissection algorithm has been proposed and implemented49,50 for identifying and utilizing the sparsity of many-body interactions. In this paper we propose to transform the summation expressions in the self-energy diagrams using tensor-contraction operations, and to further employ external linear algebra libraries (e.g. low-level C or Fortran) taking into account, e.g., memory availability, communication costs, loop fusion and ordering51–53. (Here we consider tensors simply as multidimensional objects without deeper (differential-)geometric interpretation.) With benchmark simulations in selected molecular systems we present an efficient way to compute the 2B self-energy applicable either in full time-propagation of the KBE or in the numerically less expensive GKBA variant.

a) riku.tuovinen@mpsd.mpg.de
II. MODEL AND METHOD

We consider a finite and quantum-correlated electronic system described by a time-dependent Hamiltonian

\[ \hat{H}(t) = \sum_{ij} h_{ij}(t) \hat{c}_i^\dagger \hat{c}_j + \frac{1}{2} \sum_{ijkl} v_{ijkl}(t) \hat{c}_i^\dagger \hat{c}_j^\dagger \hat{c}_k \hat{c}_l, \]  

where \( i, j, k, l \) label a complete set of one-particle states \( \{ \varphi(r) \} \), and \( \hat{c}^{(i)} \) are the annihilation (creation) operators for electrons from (to) these states. Although we assume, for simplicity, spin-compensated electrons and invariance under spin rotations, the whole consideration could easily be generalized to include also spin degrees of freedom.\(^{47,54-57}\) The objects henceforth described will be diagonal in spin space. The one-body contribution to the Hamiltonian,

\[ h_{ij}(t) = \int dr \varphi_i^*(r) h(r, t) \varphi_j(r), \]  

may have an explicit time dependence, describing, e.g., pump-probe spectroscopies or voltage pulses. These would enter in \( h(r, t) = -\frac{1}{2} \nabla^2 + w(r, t) - \mu \) as external fields \( w \). We also introduced the chemical potential \( \mu \) and we absorbed it into the equilibrium description of the one-body part of the Hamiltonian. Atomic units, \( \hbar = m = e = 1 \), are used throughout. The two-body part accounts for interactions between the electrons with the standard two-electron Coulomb integrals

\[ v_{ijkl}(t) = \int dr \int dr' \varphi_i^*(r) \varphi_j^*(r') \varphi_k(r) \varphi_l(r') / |r - r'|. \]  

Even though the Coulomb interaction itself is instantaneous, in Eq. (1) we allow the strength of the two-body part to be time-dependent to describe, e.g., interaction quenches or adiabatic switching. For real-valued basis functions \( \varphi \) the Coulomb integrals in Eq. (3) follow 8-point permutation symmetry

\[ v_{ijkl} = v_{jikl} = v_{iljk} = v_{klji} = v_{ljik} = v_{klij} = v_{ijlk}. \]  

which can be verified by permuting dummy integration variables and by complex conjugation. The following discussion is not limited to this choice, however, and also complex and spin-dependent basis functions could be used.

To calculate time-dependent nonequilibrium quantities we use the equations of motion for the one-particle Green’s function on the Keldysh contour \( \gamma^4-6\). This object is defined as

\[ G_{ij}(z, z') = -i \langle T_\gamma [\hat{c}_i(z) \hat{c}^*_j(z')] \rangle, \]  

where \( T_\gamma \) is the contour ordering operator and the variables \( z, z' \) specify the location of the Heisenberg-picture operators \( \hat{c} \) on the Keldysh contour. The contour has a forward and a backward branch on the real-time axis, \([t_0, \infty[, \) and also a vertical branch on the imaginary axis, \([t_0, t_0 - i\beta] \) with inverse temperature \( \beta \). The Green’s function includes detailed information about particle propagation, and important physical quantities such as electric currents or photoemission spectra can be extracted from it. The Green’s function \( G \) satisfies the integro-differential equations of motion\(^5\)

\[ [i \partial_z - h(z)] G(z, z') = \delta(z, z') + \int d\bar{z} \Sigma(z, \bar{z}) G(\bar{z}, z') \]  

(6)

\[ G(z, z') \left[ -i \bar{\partial}_{z'} - h(z') \right] = \delta(z, z') + \int d\bar{z} G(z, \bar{z}) \Sigma(\bar{z}, z') \]  

(7)

where all objects are matrices with respect to the basis of one-particle states \( \{ \varphi(r) \} \). The self-energy \( \Sigma \) accounts for the electronic interactions. While some two-particle quantities, such as interaction energies and double occupancies, can also be computed from this picture\(^58,59\), the introduction of the self-energy transforms the many-body problem to an effective quasiparticle picture, and higher order correlations, such as the pair distribution function, are not directly accessible\(^60,61\). Depending on the arguments \( z, z' \), the Green’s function, \( G(z, z') \), and the self-energy, \( \Sigma(z, z') \), defined on the time contour have components lesser (\(<\)), greater (\(>\)), retarded (\(R\)), advanced (\(A\)), left (\(|\rangle\)), right (\(|\rangle\)) and Matsubara (\(M\))^5. Typically, one concentrates on the particle and hole propagation in terms of \( G^<(t, t') \) and \( G^>(t, t') \) where the time arguments \( t \) and \( t' \) refer to the (real) times when a particle is added or removed from the system. Furthermore, the one-particle reduced density matrix (1RDM) \( \rho(t) \equiv -i G^<(t, t) \) from which one could compute the expectation value of any one-body operator. Taking the equal-time limit \( (t' \to t^+) \) one obtains from Eqs. (6) and (7)

\[ \frac{d}{dt} G^>(t, t) = [h(t) + \Sigma_{HF}(t), G^<(t, t)] + I(t), \]  

(8)

where we defined the collision integral

\[ I(t) = \int_{t_0}^{t} dt' \left[ \Sigma^<(t, t') G^>(t', t) - \Sigma^>(t, t') G^<(t', t) \right. \]

\[ + \left. G^<(t, t') \Sigma^>(t', t) - G^>(t, t') \Sigma^<(t', t) \right]. \]  

(9)

In addition, in Eq. (8) we separated the time-local and time-non-local contributions to the self-energy as \( \Sigma = \Sigma_{HF} + \Sigma_c \), the former being referred to as the Hartree-Fock (HF) self-energy and the latter the correlation self-energy, see Fig. 1. This allows for the extraction of a time-local effective single-particle Hamiltonian, \( h(t) + \Sigma_{HF}(t) \). The collision integrals therefore incorporate only the correlation self-energies \( \Sigma_c \). Importantly, the self-energies depend on the Green’s functions themselves, \( \Sigma[G] \), and therefore the equation of motion needs to be solved self-consistently. The correlation self-energies are typically obtained by a diagrammatic expansion where terms can be systematically summed up.
FIG. 1. Diagrammatic representations of the Hartree-Fock (a) and the second-Born (b) correlation self-energies. The straight lines denote electronic Green’s functions whereas the wiggly lines denote the electronic interactions. The internal indices are summed over. Each diagram comes with a prefactor \((-1)^{N_{\text{loop}}-N_{\text{int}}}\) where \(N_{\text{loop}}\) is the number of loops and \(N_{\text{int}}\) is the number of interaction lines\(^5\). The direct terms with a loop furthermore take an overall spin-degeneracy factor \(\xi\), which in this case is \(\xi = 2^{56,62}\).

As can be seen from Eq. (13) computing the full self-energy matrix by direct looping takes \(N_h^8\) operations where \(N_h\) is the size of the basis. However, it is possible to reduce this scaling to \(\propto N_h^3\) by grouping and reorganizing the objects in Eq. (13)\(^{39,65,67–70}\). We address this more thoroughly in Sec. III C. It is also to be noted that the 2B self-energy is non-local in time, i.e., this computation needs to be performed for two times \(t\) and \(t'\), and it is important to keep track of the correct time arguments in the objects \(v\), \(G\) and \(\tilde{G}\). While the 2B approximation together with the GKB allows for a maximal speed-up in computational scaling compared to the full two-time KBE (\(T^2\) vs. \(T^3\), \(T\) being the total propagation time), GKB simulations with \(GW\) and \(T\)-matrix approximations to the self-energy have also been performed\(^{71,72}\).

We note that the NEGF method using the 2B self-energy, sometimes referred to as second-order Green’s function (GF2)\(^{19,48,69,73}\), can be related to the second-order Möller-Plesset perturbative expansion (MP2)\(^{74}\). The construction of the MP2 correction is similar to the 2B self-energy, although in the NEGF approach the self-energy enters nonlinearly into an updated Green’s function and this procedure is continued until convergence is reached, whereas the MP2 can be related to the first step of this iteration\(^{69,73,75}\).

Next, we consider three different cases for the interaction vertex: (1) Diagonal basis where the Coulomb integrals take the Hubbard-like form \(v_{ijkl} = U_i \delta_{ij} \delta_{ik} \delta_{kl}\); (2) Symmetric basis where the Coulomb integrals allow for non-diagonal or long-range interactions \(v_{ijkl} = V_{ij} \delta_{ik} \delta_{jk}\) but the 4-point vertex is symmetric (density-density type interaction); and (3) The general basis of the full Coulomb integral \(v_{ijkl}\). From the resulting structures of
the internal summations in the self-energy diagrams, we identify matrix or tensor operations. Instead of simply looping over the basis indices, employing well-established linear algebra libraries for the matrix and tensor operations\textsuperscript{51–53} may speed up the construction of the self-energy.

We denote matrix multiplication by “$\times$” and entrywise multiplication (Hadamard or Schur product) by “$\circ$”. For example, in Fortran and Mathematica the entrywise products are done through simple multiplication operators “:*” whereas the matrix product is done through the “matmul” or “*” operators. In C++ with the Armadillo library\textsuperscript{76} the symbol “$\%$” is used for entrywise products whereas “$*$” is a matrix product. In Python with the NumPy (np) numerical library\textsuperscript{77} the entrywise product can be done with the function “np.multiply” whereas the matrix or more general tensor multiplication can be done via the “np.dot” or the “np.einsum” functions.

A. Diagonal basis

For a diagonal basis, $v_{ijkl} = U_i \delta_{ij} \delta_{ik} \delta_{il}$, Eq. (13) is simplified as

$$\Sigma_{ij} = U_i U_j G_{ij} \bar{G}_{ji} G_{ij}, \quad (14)$$

and the computational cost of constructing the full matrix therefore scales as $N^2$. In this simple case there are no further contractions to perform as the internal summations were already explicitly resolved due to the Kronecker $\delta$’s in the interaction vertex. Because in many practical implementations entrywise multiplication between two objects is only possible when they have the same dimension, we rewrite the on-site interaction $U_i$ instead as the diagonal part $V_{ii}$ of a matrix. The resulting expression can then be recasted in matrix form as an entrywise product

$$\Sigma = \text{diag}(V) \circ \text{diag}(V) \circ G \circ \bar{G}^T \circ G. \quad (15)$$

We anticipate that this is a faster construction for the whole self-energy matrix instead of looping over the basis indices $i, j$ in Eq. (14) when passing the matrix operations in Eq. (15) to an external linear algebra library.

B. Symmetric basis

For a symmetric basis, $v_{ijkl} = V_{ij} \delta_{il} \delta_{jk}$, Eq. (13) is simplified as

$$\Sigma_{ij} = 2 \sum_{kl} V_{ik} V_{jl} G_{ij} \bar{G}_{ik} G_{kl} - \sum_{kl} V_{ik} V_{jl} \bar{G}_{ik} G_{kl} G_{kj}. \quad (16)$$

We first consider the first term of Eq. (16), i.e., the second-order bubble diagram, and visualize the contraction path for efficient computation. The expression can be manipulated as

$$\Sigma_{ij}^{cb} = 2 \sum_{kl} V_{ik} V_{jl} G_{ij} (\bar{G}^T)_{kl} G_{kl}$$

$$= 2 \sum_{kl} V_{ik} V_{jl} G_{ij} (\bar{G}^T \circ G)_{kl}$$

$$= 2 \sum_{l} V_{jl} G_{ij} \sum_{k} V_{ik} (\bar{G}^T \circ G)_{kl}$$

$$= 2 \sum_{l} V_{jl} G_{ij} [V \times (\bar{G}^T \circ G)]_{il}$$

$$= 2 \sum_{l} V_{jl} G_{ij} \{(V \times (\bar{G}^T \circ G))^T\}_{li}$$

$$= 2G_{ij} \sum_{l} V_{jl} \{(V \times (\bar{G}^T \circ G))^T\}_{li}$$

$$= 2G_{ij} \{(V \times [(V \times (\bar{G}^T \circ G))^T])^T\}_{ij}$$

$$= 2\{G \circ [(V \times [(V \times (\bar{G}^T \circ G))^T])^T]\}_{ij}, \quad (17)$$

where we identified matrix transposes, entrywise products and matrix multiplications. The procedure outlined above, unfortunately, makes the final expressions less readable, but in the end the full self-energy matrix (for the bubble diagram part) may be constructed as a one-liner $\Sigma_{ij}^{cb} = 2G \circ [(V \times [(V \times (\bar{G}^T \circ G))^T])^T]$. However, as mentioned earlier, one must keep track of the time arguments, i.e., reading from left the first $V$ is evaluated at $t'$ and the second $V$ is evaluated at $t$.

Contractions on the internal summations in the self-energy diagrams do not always yield a favourable path. If we take the second term in Eq. (16), i.e., the second-order exchange diagram, obtaining an expression similar to Eq. (17) is not possible for the full self-energy matrix. However, for the diagonal part of the exchange diagram we obtain

$$\Sigma_{ii}^{ex} = - \sum_{kl} V_{ik} V_{il} G_{ik} \bar{G}_{lk} G_{kli}$$

$$= - \sum_{l} (V \circ G)_{il} \sum_{k} \bar{G}_{lk} (V \circ G)_{kl}$$

$$= - \sum_{l} (V \circ G)_{il} \bar{G} \times (V \circ G)_{li}$$

$$= - \{(V \circ G) \times [\bar{G} \times (V \circ G)]\}_{ii}. \quad (18)$$

The off-diagonal parts would still need to be evaluated by explicit looping as in Eq. (16), but the above contraction path may also be combined with, e.g., the dissection algorithm of Ref.\textsuperscript{89} where chosen pairs of the Coulomb integral matrix elements (according to some cut-off energy) would be used. This further reduces the requirement for looping over the basis indices.

C. General basis

For a general basis all $v_{ijkl}$ are nonvanishing. In this case the multi-index summations in the self-energy dia-
FIG. 2. Contraction paths for the computation of the self-energy in Eq. (19). The dots denote tensor-contraction operations which could be implemented, e.g., using the np.einsum function in Python which includes (from version 1.14 onwards) optimized ordering and dispatching many operations to canonical BLAS routines.

The programs and their consequent contractions are not always easy to see, but this task can be automated using, e.g., the np.einsum_path function in Python. The information obtained for an optimal sequence of contractions may further be combined with the symmetry properties (4) and with a pre-determined subset of nonzero Coulomb integrals.

Manipulating Eq. (13) gives

\[
\Sigma_{ij} = 2 \sum_{nq} \sum_{ps} T_{nqsj} G_{ps} G_{nm} T_{mr} v_{irpm} T_{isqn} \\
= 2 \sum_{nq} \sum_{ps} T_{nqsj} G_{ps} \sum_{nm} \sum_{r} \sum_{m} v_{irpm} G_{mr} G_{nm} \\
= 2 \sum_{nq} \sum_{ps} T_{nqsj} G_{ps} \sum_{nm} \sum_{r} \sum_{m} v_{irpm} G_{mr} G_{nm} \\
\]

where we defined tensor contractions \( T^{(1,2,3)} \) and permuted indices with the help of Eq. (4), identifying similar contractions consequently. We see from the last line of Eq. (19) that for constructing the full self-energy matrix the scaling over the basis is reduced from \( N_b^2 \) to \( N_b \).

As before, the readability of the self-energy in Eq. 19 suffers a bit compared to Fig. 1 or Eq. (13). However, Eq. (19) is visualized in Fig. 2, and for the sake of efficient computation the contraction operations can be grouped together and executed essentially as a single command, where the lower-level loop fusions and orderings of operations are handled by the underlying numerical library. We emphasize that while the reorganizations of the summations in Eq. (13) to arrive at Eq. (19) have already been considered to some degree in Refs. 49, 68, 70, here we concentrate on the practical computation of the self-energy by employing efficient tensor-contraction operations with a possible contraction path shown in Fig. 2. Alternative contraction paths than the one shown in Fig. 2 are also possible.

**IV. NUMERICAL BENCHMARKS**

For the three different cases presented in the previous section, (1) diagonal, (2) symmetric and (3) general bases, we now present sample numerical simulations for the purpose of benchmarking and assessing the validity and accuracy of the alternative implementations of the 2B self-energy. For test cases we choose molecular systems falling into each of the categories: 1D Hubbard chains which can be related to, e.g., conjugated polymers 63, 79–81 with local (1) and long-range interactions (2). We set the hopping energy between nearest-neighbors \( J = -1 \), the on-site electron-electron interaction \( U = 1 \), and the long-range interaction between particles at atomic sites \( i \) and \( j \) as in the Ohno model \( V_{ij} = U/\sqrt{1+|i-j|^2} \). For the case (3) we take a CH\(_4\) molecule with a general one-particle Kohn-Sham (KS) basis obtained from density-functional theory (DFT) using Octopus 84. Using this DFT calculation, the one- and two-body matrix elements [Eqs. (2) and (3)] are then constructed in the corresponding KS basis; a more detailed explanation can be found in Ref. 49.

We implement the explicit loops over the basis indices [Eqs. (14), (16), and (19)] in C++. In the cases (1) and (2) we employ the matrix operations [Eqs. (15), (17), and (18)] using the Armadillo library (version 9.200.5),76 and in the case (3) we employ the tensor operations [Eq. (19) and Fig. 2] using the NumPy library (version 1.15.1) in Python. We perform the comparisons using a regular desktop computer with an Intel Core i5-4460 @ 3.2 GHz with 6 MB cache, running on 64-bit architecture using Ubuntu 18.04 operating system incorporating the Linux kernel 4.15.0 and the GCC 7.3.0 compiler. The comparisons are done using only a single core to better benchmark the computational cost.

We perform a time-propagation à la GKBA of \( N_t \) time steps with length \( \delta \). For the sake of simpler computation, in this work we do not employ any predictor-corrector schemes. For the polymer chain we take \( N_b = 10 \) atomic sites and start the time-propagation from an initial state where \( N_b/2 \) particles are trapped to the \( N_b/2 \) leftmost sites by applying a strong confinement potential 63. This configuration relaxes once the time evolution is started. For the CH\(_4\) molecule we represent the 4 electrons by \( N_b = 10 \) basis functions, and we start the time-propagation from a HF initial state, which can be obtained from a separate (time-independent) calculation, and then suddenly switch on the many-body correlations in the 2B self-energy. This sudden process can be interpreted as an interaction quench introducing transient dynamics.
FIG. 3. Time-dependent 1RDM elements for the three different systems studied: (a) Diagonal basis with local interaction, (b) Symmetric basis with local and long-range interaction, and (c) General basis with the full Coulomb integrals. The insets show the relative difference between the two curves in the main plots.

For the case (1) we take $N_t = 5000$ time steps of length $\delta = 0.01$, for the case (2) $N_t = 2000$ time steps of length $\delta = 0.025$, and for the case (3) we take $N_t = 1000$ time steps of length $\delta = 0.05$. The reason for the varying number of time steps between the investigated cases is that a calculation with $N_t = 1000$ would be too fast to execute in case (1) for a meaningful comparison of runtimes, whereas $N_t > 1000$ in case (3) would lead to unpractically long execution times for the sake of the present study. Here we are not too concerned about the physical mechanisms taking place during the transient oscillations or how accurate the 2B self-energy is compared to more sophisticated approximations, but our aim is simply to assess the validity of the proposed computation scheme, and to compare execution runtimes.

In Fig. 3 we show the transient dynamics of the three cases discussed above. The execution runtimes for each of these simulations are shown in Tab. I. We confirm that within numerical accuracy, both looping over the basis indices and employing tensor-contraction operations, give the same result. Importantly, the execution runtimes are brought down by employing the tensor-contraction operations in the computation of the 2B self-energy. Furthermore, we have checked by increasing the number of time steps that the runtimes increase accordingly, i.e., the gain factors in Tab. I remain roughly similar. For additional validation we have compared our data in Fig. 3(c) against the CHEERS code and we find perfect agreement. We note in passing that an ill-advised looping over the full basis in Eq. (13) ($\propto N_b^8$) instead of the reduced looping in Eq. (19) ($\propto N_b^5$) would result in considerably higher execution runtimes.

As the number of basis functions $N_b = 10$ was rela-

| Basis        | Scaling | Time (loop) | Time (contr.) | Gain |
|--------------|---------|-------------|---------------|------|
| diagonal     | $N_b^5$ | 177         | 164           | 1.08 |
| symmetric    | $N_b^5$ | 1213        | 731           | 1.66 |
| general      | $N_b^5$ | 1527        | 1333          | 1.15 |

TABLE I. Comparison of serial runtimes (in seconds) of sample simulations of basis size $N_b = 10$ when calculating the self-energy by looping over the basis indices or employing tensor-contraction operations. The gain factor is defined as the ratio of the runtimes. (Note that different number of time steps is taken for the different lines for better comparison of the runtimes.)
tively small in the previous calculations, we expect increase in the gain factors when larger basis is used, due to profiting more from the optimized underlying numerical libraries. In Fig. 4 we show the execution runtime corresponding to Fig. 3(c) but with varying number of basis functions. With explicit looping over the basis indices we observe $\propto N_b^0$ behaviour. For smaller basis sizes the explicit looping is faster compared to the tensor-contraction operations done on the NumPy arrays. However, for larger basis sizes the runtimes using the tensor-contraction operations are significantly smaller, also following a power law behaviour $\propto N_b^\alpha$ for which we empirically find $\alpha \approx 4.34 \pm 0.17$, see Fig. 4. This exponent and its statistical errors were extracted by performing a nonlinear least squares fit to the flat part, $N_b \in [14, 24]$, using gnuplot. This reduced scaling could be related to the optimization of matrix multiplication using the Strassen algorithm, and to more advanced methods for tensor contraction algorithms which can scale faster than the naive looping scheme.

V. CONCLUSION

We presented an efficient way to compute the 2B self-energy diagrams, in the NEGF approach, by using tensor-contraction operations. The apparent attraction for efficient computation of the 2B self-energy, in particular, was due to the maximal speed-up in computational scaling when used together with the GKBA. The internal summations in the self-energy calculations were transformed into matrix and tensor operations to be performed by external low-level linear algebra libraries, speeding up the computation. We anticipate the speedup may be even more advantageous when the code is executed in parallel, taking full advantage of the optimized underlying numerical libraries. Instead of looping over the basis indices, utilizing efficiently optimized external numerical libraries for the tensor-contraction operations has the further advantage of speeding up the computation if/when future implementations of the the external libraries become faster and even more efficient.

There has been recent progress in reducing the computational bottleneck of constructing various self-energy approximations by using stochastic methods. Here we mention the work of Neuhauser, Baer, and Zgid who considered the 2B self-energy in an equilibrium setting and achieved a much more favourable quadratic scaling over the fifth power. While the reduced scaling with respect to the basis size using these stochastic methods goes beyond our findings, it is not straightforward to argue how the accuracy of such a stochastic-sampling approach may affect convergence or error propagation in an out-of-equilibrium setting. In this case one would have to sample not a single $\tau$-axis (Matsubara) self-energy but instead a new slice of ever-expanding self-energies $\Sigma_{\alpha}(t, t')$ in the two-time plane. However, it would be a promising venue to extend the stochastic methods also to real time in future studies.

The presented approach is not limited to the 2B self-energy only but could be readily used for other correlation self-energies, such as $GW$ or $T$-matrix. In addition, many other similar multi-index operations, such as evaluating the initial correlations collision integral in Ref., might become computationally more accessible by using the tensor-contraction representations. In the present work we considered only the GKBA with Hartree-Fock propagators, but extensions to correlated approximations to the propagator are also directly applicable in our approach. The presented simulations in selected molecular systems provided concrete evidence of the accuracy and applicability of the tensor-contraction operations. With reasonable and precise implementations or variations of the present study, we expect this procedure to allow for considerably larger basis sizes to be possible to address in forthcoming NEGF+first principles simulations.

ACKNOWLEDGMENTS

R.T. and M.A.S. acknowledge funding by the DFG (Grant No. SE 2558/2-1) through the Emmy Noether program. We wish to thank Damian Hofmann, Gianluca Stefanucci, Michael Bonitz, and Angel Rubio for productive discussions.

1G. Baym and L. P. Kadanoff, Phys. Rev. 124, 287 (1961).
2L. P. Kadanoff and G. Baym, Quantum Statistical Mechanics (W. A. Benjamin, New York, 1962).
3L. V. Keldysh, Zh. Eksp. Teor. Fiz. 47, 1515 (1964), [Sov. Phys. JETP 20, 1018 (1965)].
4P. Danielewicz, Ann. Phys. (N. Y.) 152, 239 (1984).
5G. Stefanucci and R. van Leeuwen, Nonequilibrium Many-Body Theory of Quantum systems: A Modern Introduction (Cambridge University Press, Cambridge, 2013).
D. G. A. Smith and J. Gray, J. Open Source Softw. 3, 753 (2018).

B. Kirtman and B. Champagne, Int. Rev. Phys. Chem. 16, 389 (1997).

M. van Faassen, P. L. de Boeij, R. van Leeuwen, J. A. Berger, and J. G. Snijders, Phys. Rev. Lett. 88, 186401 (2002).

M. van Faassen, P. L. de Boeij, R. van Leeuwen, J. A. Berger, and J. G. Snijders, J. Chem. Phys. 118, 1044 (2003).

K. Ohno, Theoret. Chim. Acta 2, 219 (1964).

D. Baeriswyl, D. K. Campbell, and S. Mazumdar, in Conjugated Conducting Polymers, edited by H. G. Giess (Springer, 1992).

M. A. Marques, A. Castro, G. F. Bertsch, and A. Rubio, Comput. Phys. Commun. 151, 60 (2003).

V. Strassen, Numer. Math. 13, 354 (1969).

Q. Ge, Y. Gao, R. Baer, E. Rabani, and D. Neuhauser, J. Phys. Chem. Lett. 5, 185 (2014).

D. Neuhauser, Y. Gao, C. Arntsen, C. Karshenas, E. Rabani, and R. Baer, Phys. Rev. Lett. 113, 076402 (2014).

Z. Ruan and R. Baer, Mol. Phys. 116, 2490 (2018).