Supplemental material for:
Solving the Transient Dyson Equation with Quasilinear Complexity via Matrix Compression

Baptiste LAMIC

Univ. Grenoble Alpes, CEA, IRIG-Pheliqs, F-38000 Grenoble, France
(Dated: November 18, 2025)

EXPANDED DESCRIPTION OF THE KEY EQUATIONS

The objective of this section is to provide a more explicit and general formulation of the algorithm. For reader convenience, we reproduce some parts of the main text.

Problem Statement

We present here an algorithm to solve the Dyson equation in the transient regime within the Keldysh non-equilibrium field theory, for a given self-energy Σ. In its real-time formulation, the Dyson equation reads

$$G(t, t') = g(t, t') + \iint_{-\infty}^{\infty} g(t, t_1)\, \Sigma(t_1, t_2)\, G(t_2, t')\, dt_1\, dt_2\,. \tag{1}$$

We select the Keldysh basis such that the kernels can be written as

$$G \equiv \begin{pmatrix} 0 & G^{\mathrm{A}} \\ G^{\mathrm{R}} & G^{\mathrm{K}} \end{pmatrix}, \quad \Sigma \equiv \begin{pmatrix} \Sigma^{\mathrm{K}} & \Sigma^{\mathrm{R}} \\ \Sigma^{\mathrm{A}} & 0 \end{pmatrix}, \tag{2}$$

where the superscripts R, A, and K denote the retarded, advanced, and kinetic components, as detailed in [1]. Equation 1 can be rewritten as

$$G^{\mathrm{R}} = g^{\mathrm{R}} + g^{\mathrm{R}} \Sigma^{\mathrm{R}} G^{\mathrm{R}}, \tag{3}$$

$$G^{\mathrm{A}} = (G^{\mathrm{R}})^{\dagger}, \tag{4}$$

$$G^{\mathrm{K}} = \left(\mathbb{1} + G^{\mathrm{R}} \Sigma^{\mathrm{R}}\right) g^{\mathrm{K}} \left(\mathbb{1} + \Sigma^{\mathrm{A}} G^{\mathrm{A}}\right) + G^{\mathrm{R}} \Sigma^{\mathrm{K}} G^{\mathrm{A}}, \tag{5}$$

where all integrations are implicit. The orbital degrees of freedom are assumed to be discretized so that for any two kernels (F, R),

$$[F(t, t_1) R(t_1, t')]_{p,q} = \sum_{k=1}^{n_o} F_{p,k}(t, t_1) R_{k,q}(t_1, t'), \tag{6}$$

where p, q label orbitals and n_o is the number of orbital degrees of freedom. We further suppose the system is simulated from t_0 to t_{end} and that the self-energy is supported only inside this domain; i.e., for all $(t, t') \notin [t_0, t_{\mathrm{end}}]^2$, $\Sigma(t, t')^{R/A/K} = 0$. This restriction ensures that all integration domains lie inside $[t_0, t_{\mathrm{end}}]$. The classical time stepping algorithm requires $\mathcal{O}(N^3)$ operations and $\mathcal{O}(N^2)$ space, with N the number of time steps. By combining a compact discretization of the Dyson equation with matrix compression techniques, we reach $\mathcal{O}(N \log N)$ time and space complexity.

Algorithm

The algorithm relies on two main observations:

- Kernels can be efficiently compressed using hierarchical matrix formats such as HSS (Hierarchically Semi-Separable) or HODLR, greatly reducing memory and computational cost.

- Kernel products arising from discretized Dyson equations can be recast as matrix operations compatible with these compressed representations, preserving efficiency end-to-end.

Equation discretization

The time axis is discretized into $N + 1$ points on a uniform grid with step $\delta_t = (t_{\text{end}} - t_0)/N$. The i-th point is $t_i = t_0 + i\,\delta_t$. For integration, we introduce quadrature weights $w_{i,l,j}$ well-defined for $|i - j| > d$, with $d \in \mathbb{N}$, and, upon suitable redefinition, $w_{i,l,j} = 0$ whenever $l \notin \{i, \ldots, j\}$. We require that the weights can be factorized as

$$w_{i,l,j} = a_{i,l}\, b_{l,j} + c_{i,l,j}, \tag{7}$$

subject to the existence of integer n_w such that

$$|i - j| > 2n_w \implies \begin{cases} a_{i,j} = 1 \\ b_{i,j} = 1 \\ c_{i,l,j} = 0 \end{cases}. \tag{8}$$

This property is satisfied by any quadrature whose weights become constant far from the domain boundaries, for example, Gregory quadrature, including the trapezoidal rule with $d = 0$, $n_w = 1$. Therefore, the product of two retarded kernels A^R and B^R can be approximated by a Nyström discretization for $i > j + d$:

$$\left[A^R B^R\right]_{p,q}(t_i, t_j) \equiv \sum_{k=1}^{n_o} \int_{t_j}^{t_i} A_{p,k}^R(t_i, t_1) B_{k,q}^R(t_1, t_j) dt_1 \tag{9}$$

$$\approx \delta_t \sum_{k=1}^{n_o} \sum_{l=0}^{N} w_{i,l,j}\, A_{p,k}^R(t_i, t_l) B_{k,q}^R(t_l, t_j), \tag{10}$$

and the entries for $i \leq j + d$ are computed by other means. Since there are only $\mathcal{O}(N)$ such elements, such elements and they are located near the diagonal, all of these entries can be evaluated in $\mathcal{O}(N)$ time. Inserting the quadrature decomposition, we almost rewrite the kernel product as a matrix product

$$\left[A^R B^R\right]_{p,q}(t_i, t_j) \approx \delta_t \sum_{k=1}^{n_o} \sum_{l=0}^{N} a_{i,l}\, A_{p,k}^R(t_i, t_l)\, b_{l,j} B_{k,q}^R(t_l, t_j) \tag{11}$$

$$+ \delta_t \sum_{k=1}^{n_o} \sum_{l=j}^{i} c_{i,l,j} A_{p,k}^R(t_i, t_l) B_{k,q}^R(t_l, t_j). \tag{12}$$

This suggests defining the correction term

$$\mathcal{C}[A^R, B^R]_{p,q}(t_i, t_j) \equiv \delta_t \sum_{k=1}^{n_o} \sum_{l=j}^{i} c_{i,l,j} A_{p,k}^R(t_i, t_l) B_{k,q}^R(t_l, t_j) + \Pi_d(i - j) \left[A^R B^R\right]_{p,q}(t_i, t_j), \tag{13}$$

where $\Pi_d(i - j) = 1$ if $|i - j| \leq d$ and 0 otherwise. Only $\mathcal{O}(N)$ coefficients $c_{i,l,j}$, all located close to the diagonal, are nonzero, so $\mathcal{C}$ can be computed in $\mathcal{O}(N)$ time. Thus for all $i, j \in \{1, \ldots, N\}$, the kernel product can be expressed as

$$\left[A^R B^R\right]_{p,q}(t_i, t_j) \approx \delta_t \sum_{k=1}^{n_o} \sum_{l=0}^{N} a_{i,l} A_{p,k}^R(t_i, t_l)\, b_{l,j} B_{k,q}^R(t_l, t_j) + \mathcal{C}[A^R, B^R]_{p,q}(t_i, t_j). \tag{14}$$

Equipped with this expression, we can reformulate the kernel product as a matrix product.

Matrix Notation.

Let $\sigma : (p, i) \mapsto p + i\,n_o$ be the map from orbital and time indices to matrix indices. we define $\mathbf{A} \in \mathbb{C}^{n_o N \times n_o N}$ as

$$\mathbf{A}_{\sigma(p,i),\,\sigma(q,j)} \equiv A_{p,q}(t_i, t_j). \tag{15}$$

We define the matrix form of a and b as

$$\mathbf{a}_{\sigma(p,i),\sigma(q,j)} = a_{i,j}, \tag{16}$$

$$\mathbf{b}_{\sigma(p,i),\sigma(q,j)} = b_{i,j}. \tag{17}$$

And finally, we introduce the blockwise Hadamard product $\odot$ defined as

$$[\mathbf{A} \odot \mathbf{B}]_{\sigma(p,i),\,\sigma(q,j)} \equiv \sum_{k=1}^{n_o} \mathbf{A}_{\sigma(p,i),\,\sigma(k,j)} \, \mathbf{B}_{\sigma(k,i),\,\sigma(q,j)}. \tag{18}$$

Combining all our ingredients, we obtain the compact expression

$$[\mathbf{A}^R \mathbf{B}^R] = (\mathbf{a} \odot \mathbf{A}^R)(\mathbf{b} \odot \mathbf{B}^R) + \mathbf{C}[A^R, B^R]. \tag{19}$$

The matrix $\mathbf{C}[A^R, B^R]$ is extremely sparse and can be evaluated cheaply using standard methods in $\mathcal{O}(N)$ operations, which is trivial when using trapezoidal quadrature. The blockwise Hadamard product is computed efficiently by exploiting that $a_{i,j}$ and $b_{i,j}$ deviate from unity only near the diagonal. Since matrices $\mathbf{A}$ and $\mathbf{B}$ are already in HSS format, the evaluation of eq. (19) achieves overall $\mathcal{O}(N)$ complexity. We can now turn to solving the Dyson equation.

Application to the Dyson Equation.

Let's use this discretization of the Kernel product to discretize the retarded Dyson equation. First we define $F^R \equiv g^R \Sigma^R$,

$$\mathbf{G}^R = \mathbf{g}^R + (\mathbf{a} \odot \mathbf{F}^R)(\mathbf{b} \odot \mathbf{G}^R) + \mathbf{C}[F^R, G^R]. \tag{20}$$

We regroup the terms,

$$(\mathbf{b} \odot \mathbf{G}^R) = \mathbf{g}^R + (\mathbf{a} \odot \mathbf{F}^R)(\mathbf{b} \odot \mathbf{G}^R) + \mathbf{C}[F^R, G^R] - ((\mathbf{1} - \mathbf{b}) \odot \mathbf{G}^R), \tag{21}$$

and define

$$\mathbf{S} \equiv \mathbf{C}[F^R, G^R] - ((\mathbf{1} - \mathbf{b}) \odot \mathbf{G}^R). \tag{22}$$

By design, $\mathbf{S}$ is only nonzero near the diagonal, i.e., $\mathbf{S}_{\sigma(p,i),\,\sigma(q,j)} \neq 0 \implies |i - j| \leq \max(2n_w, d)$. Consequently, $\mathbf{S}$ is efficiently computable in $\mathcal{O}(N)$ time. The Dyson equation thus becomes

$$(\mathbf{b} \odot \mathbf{G}^R) = [\mathrm{Id} - (\mathbf{a} \odot \mathbf{F}^R)]^{-1} \left(\mathbf{g}^R + \mathbf{S} \right), \tag{23}$$

with Id the identity. This equation can be solved efficiently in compressed format, yielding the retarded Green function throughout most of the domain and enabling evaluation of the full Green function over the complete domain in $\mathcal{O}(N)$ operations when starting from already compressed matrices. As discussed in the main text, the compression bottleneck is bypassed using stochastic compression methods, which preserves the quasi-linearity of the entire approach. For trapezoidal quadrature, these expressions reduce to those presented in the main text.

_____

[1] J. Rammer, *Quantum Field Theory of Non-Equilibrium States*, Vol. 9780521874 (Cambridge University Press, Cambridge, 2007) pp. 1–536.