Hierarchical low rank compression, with an application in quantum many-body physics

Jason Kaye

CCM/CCQ

October 19, 2020

On work with Denis Golež of CCQ

Matrix structures based on low rank decomposition

What structure does your matrix have?

What structure does your matrix have?

- Matrices found in scientific computing often have structures that can be exploited to reduce *asymptotic complexity* of storage, application, solves, other operations

What structure does your matrix have?

- Matrices found in scientific computing often have structures that can be exploited to reduce *asymptotic complexity* of storage, application, solves, other operations
- *Sparsity* is the simplest structure...

What structure does your matrix have?

- Matrices found in scientific computing often have structures that can be exploited to reduce *asymptotic complexity* of storage, application, solves, other operations
- *Sparsity* is the simplest structure...
  - $\mathcal{O}\left(N^2\right) \to \mathcal{O}(N)$ in application of second-order finite difference matrix:

$$\begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & \ddots & & \vdots \\ 0 & \ddots & \ddots & -1 & 0 \\ \vdots & & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix}$$

What structure does your matrix have?

- Matrices found in scientific computing often have structures that can be exploited to reduce *asymptotic complexity* of storage, application, solves, other operations
- *Sparsity* is the simplest structure...
  - $\mathcal{O}\left(N^2\right) \to \mathcal{O}\left(N\right)$ in application of second-order finite difference matrix:

$$\begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & \ddots & & \vdots \\ 0 & \ddots & \ddots & -1 & 0 \\ \vdots & & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix}$$

- ...but there are many others

What structure does your matrix have?

- Matrices found in scientific computing often have structures that can be exploited to reduce *asymptotic complexity* of storage, application, solves, other operations
- *Sparsity* is the simplest structure...
  - $\mathcal{O}\left(N^2\right) \to \mathcal{O}\left(N\right)$ in application of second-order finite difference matrix:

$$\begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & \ddots & & \vdots \\ 0 & \ddots & \ddots & -1 & 0 \\ \vdots & & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix}$$

- ...but there are many others
  - FFT uses *recursive* structure of discrete Fourier transform matrix to get fast matrix-vector product: $\mathcal{O}\left(N^2\right) \to \mathcal{O}\left(N \log N\right)$

What structure does your matrix have?

- Matrices found in scientific computing often have structures that can be exploited to reduce *asymptotic complexity* of storage, application, solves, other operations
- *Sparsity* is the simplest structure...
  - $\mathcal{O}\left(N^2\right) \to \mathcal{O}\left(N\right)$ in application of second-order finite difference matrix:

$$\begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & \ddots & & \vdots \\ 0 & \ddots & \ddots & -1 & 0 \\ \vdots & & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix}$$

- ...but there are many others
  - FFT uses *recursive* structure of discrete Fourier transform matrix to get fast matrix-vector product: $\mathcal{O}\left(N^2\right) \to \mathcal{O}\left(N \log N\right)$
  - *Low rank* and *hierarchical low rank* structure

Numerically low rank matrices

Numerically low rank matrices

- Rank of A: # linearly independent columns/rows

Numerically low rank matrices

- Rank of A: # linearly independent columns/rows
  - All columns of A can be written as linear combination of k (rank) columns

Numerically low rank matrices

- Rank of A: # linearly independent columns/rows
  - All columns of A can be written as linear combination of k (rank) columns
- ε-rank: # linearly independent columns *to accuracy* ε

Numerically low rank matrices

- Rank of A: # linearly independent columns/rows
  - All columns of A can be written as linear combination of k (rank) columns
- ε-rank: # linearly independent columns *to accuracy* ε
  - All columns of A can be written as linear combination of k columns *with error* ε

Numerically low rank matrices

- Rank of A: # linearly independent columns/rows
  - All columns of A can be written as linear combination of k (rank) columns
- ε-rank: # linearly independent columns *to accuracy* ε
  - All columns of A can be written as linear combination of k columns *with error* ε
- $N \times N$ matrix *numerically low rank* if ε-rank $k \ll N$

Numerically low rank matrices

- Rank of A: # linearly independent columns/rows
  - All columns of A can be written as linear combination of k (rank) columns
- ε-rank: # linearly independent columns *to accuracy* ε
  - All columns of A can be written as linear combination of k columns *with error* ε
- $N \times N$ matrix *numerically low rank* if ε-rank $k \ll N$
- Truncated SVD:

Numerically low rank matrices

- Rank of A: # linearly independent columns/rows
  - All columns of A can be written as linear combination of k (rank) columns
- ε-rank: # linearly independent columns *to accuracy* ε
  - All columns of A can be written as linear combination of k columns *with error* ε
- $N \times N$ matrix *numerically low rank* if ε-rank $k \ll N$
- Truncated SVD:

$$A = \sum_{i=1}^{N} \sigma_i u_i v_i^T \approx \sum_{i=1}^{k} \sigma_i u_i v_i^T$$

Numerically low rank matrices

- Rank of A: # linearly independent columns/rows
  - All columns of A can be written as linear combination of k (rank) columns
- ε-rank: # linearly independent columns *to accuracy* ε
  - All columns of A can be written as linear combination of k columns *with error* ε
- $N \times N$ matrix *numerically low rank* if ε-rank $k \ll N$
- Truncated SVD:

$$A = \sum_{i=1}^{N} \sigma_i u_i v_i^T \approx \sum_{i=1}^{k} \sigma_i u_i v_i^T$$

  - Storage: $N^2 \to 2kN + k$

Numerically low rank matrices

- Rank of A: # linearly independent columns/rows
  - All columns of A can be written as linear combination of k (rank) columns
- ε-rank: # linearly independent columns *to accuracy* ε
  - All columns of A can be written as linear combination of k columns *with error* ε
- $N \times N$ matrix *numerically low rank* if ε-rank $k \ll N$
- Truncated SVD:

$$A = \sum_{i=1}^{N} \sigma_i u_i v_i^T \approx \sum_{i=1}^{k} \sigma_i u_i v_i^T$$

  - Storage: $N^2 \to 2kN + k$
  - Matvec: $\mathcal{O}\left(N^2\right) \to \mathcal{O}\left(kN\right)$

Physics example: well-separated interacting charges

Physics example: well-separated interacting charges

- Electrostatic potential at target point (x_i, y_i, z_i) due to N source charges of strength q_j at source points (x_j', y_j', z_j'):

Physics example: well-separated interacting charges

- Electrostatic potential at target point (x_i, y_i, z_i) due to N source charges of strength q_j at source points (x_j', y_j', z_j'):

$$\phi(x_i, y_i, z_i) = -\sum_{j=1}^{N} \frac{q_j'}{\sqrt{(x_i - x_j')^2 + (y_i - y_j')^2 + (z_i - z_j')^2}}$$

Physics example: well-separated interacting charges

- Electrostatic potential at target point (x_i, y_i, z_i) due to N source charges of strength q_j at source points (x_j', y_j', z_j'):

$$\phi(x_i, y_i, z_i) = -\sum_{j=1}^{N} \frac{q_j'}{\sqrt{(x_i - x_j')^2 + (y_i - y_j')^2 + (z_i - z_j')^2}}$$

- Computing potential at N target points due to all N source points is a matrix-vector product – $\mathcal{O}\left(N^2\right)$ work

Physics example: well-separated interacting charges

- Electrostatic potential at target point (x_i, y_i, z_i) due to N source charges of strength q_j at source points (x_j', y_j', z_j'):

$$\phi(x_i, y_i, z_i) = -\sum_{j=1}^{N} \frac{q_j'}{\sqrt{(x_i - x_j')^2 + (y_i - y_j')^2 + (z_i - z_j')^2}}$$

- Computing potential at N target points due to all N source points is a matrix-vector product – $\mathcal{O}\left(N^2\right)$ work
- If sources and targets fixed and well-separated, can we compute potential for some new charge strengths fast?

Physics example: well-separated interacting charges

- Electrostatic potential at target point (x_i, y_i, z_i) due to N source charges of strength q_j at source points (x_j', y_j', z_j'):

$$\phi(x_i, y_i, z_i) = -\sum_{j=1}^{N} \frac{q_j'}{\sqrt{(x_i - x_j')^2 + (y_i - y_j')^2 + (z_i - z_j')^2}}$$

- Computing potential at N target points due to all N source points is a matrix-vector product – $\mathcal{O}\left(N^2\right)$ work
- If sources and targets fixed and well-separated, can we compute potential for some new charge strengths fast?
- Yes – TSVD gives $\mathcal{O}(N)$ algorithm

Physics example: N-particle interaction problem

Physics example: N-particle interaction problem

- Targets = sources, with regularized interaction:

Physics example: N-particle interaction problem

- Targets = sources, with regularized interaction:

$$\phi(x_i, y_i, z_i) = -\beta \sum_{j=1}^{N} \frac{q'_j}{\sqrt{\beta^2 + (x_i - x'_j)^2 + (y_i - y'_j)^2 + (z_i - z'_j)^2}}$$

Physics example: *N*-particle interaction problem

- Targets = sources, with regularized interaction:

$$\phi(x_i, y_i, z_i) = -\beta \sum_{j=1}^{N} \frac{q_j'}{\sqrt{\beta^2 + (x_i - x_j')^2 + (y_i - y_j')^2 + (z_i - z_j')^2}}$$

- Charges no longer well-separated, so we use a standard fast algorithms trick – apply previous idea hierarchically

Physics example: N-particle interaction problem

- Targets = sources, with regularized interaction:

$$\phi(x_i, y_i, z_i) = -\beta \sum_{j=1}^{N} \frac{q'_j}{\sqrt{\beta^2 + (x_i - x'_j)^2 + (y_i - y'_j)^2 + (z_i - z'_j)^2}}$$

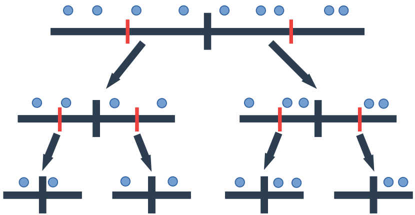
- Charges no longer well-separated, so we use a standard fast algorithms trick – apply previous idea hierarchically
- Consider particles on a line:

Physics example: *N*-particle interaction problem

- Targets = sources, with regularized interaction:

$$\phi(x_i, y_i, z_i) = -\beta \sum_{j=1}^{N} \frac{q_j'}{\sqrt{\beta^2 + (x_i - x_j')^2 + (y_i - y_j')^2 + (z_i - z_j')^2}}$$

- Charges no longer well-separated, so we use a standard fast algorithms trick – apply previous idea hierarchically
- Consider particles on a line:

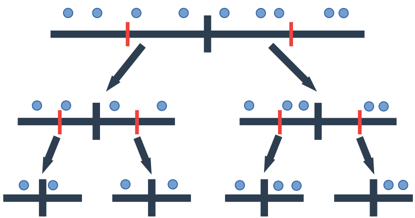


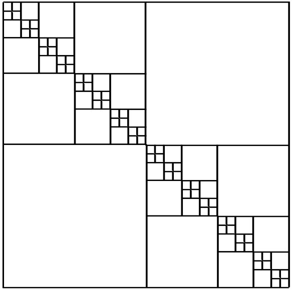
Physics example: N-particle interaction problem

- Targets = sources, with regularized interaction:

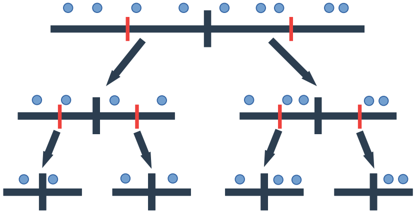
$$\phi(x_i, y_i, z_i) = -\beta \sum_{j=1}^{N} \frac{q_j'}{\sqrt{\beta^2 + (x_i - x_j')^2 + (y_i - y_j')^2 + (z_i - z_j')^2}}$$

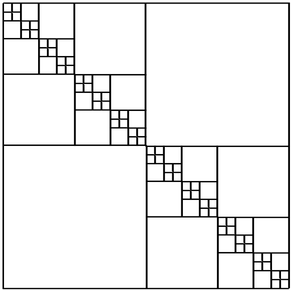
- Charges no longer well-separated, so we use a standard fast algorithms trick – apply previous idea hierarchically
- Consider particles on a line:





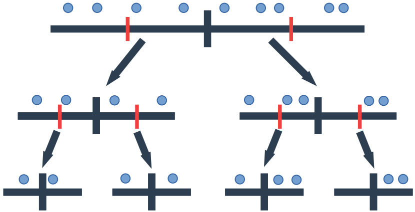
Hierarchical off-diagonal low rank (HODLR) matrices

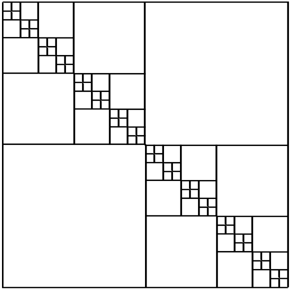




Hierarchical off-diagonal low rank (HODLR) matrices

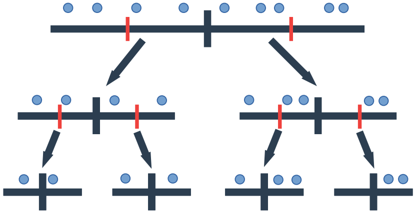
- HODLR matrix - blocks in partition have ε-rank at most $k \ll N$

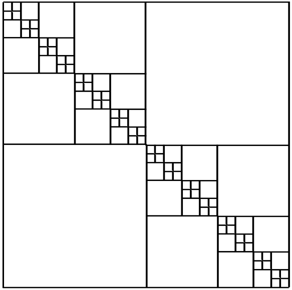




Hierarchical off-diagonal low rank (HODLR) matrices

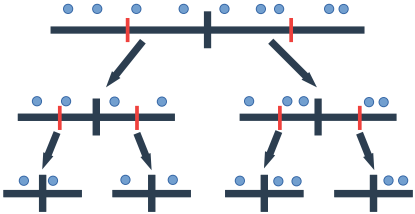
- HODLR matrix - blocks in partition have ε-rank at most $k \ll N$
  - Compute TSVD of all blocks

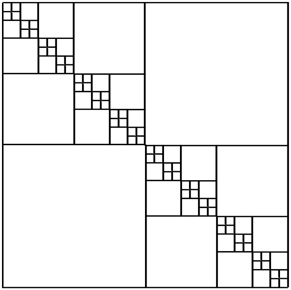




Hierarchical off-diagonal low rank (HODLR) matrices

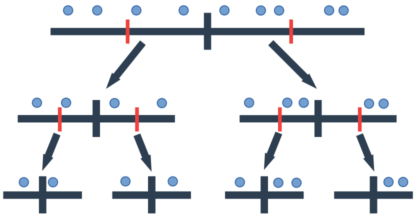
- HODLR matrix - blocks in partition have ε-rank at most $k \ll N$
  - Compute TSVD of all blocks
  - Storage: $N^2 \to \mathcal{O}(kN \log N)$

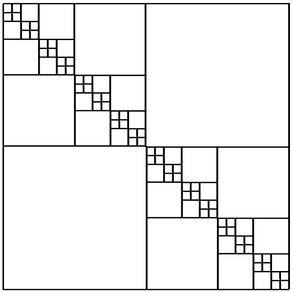




Hierarchical off-diagonal low rank (HODLR) matrices

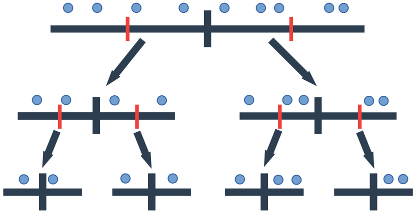
- HODLR matrix - blocks in partition have ε-rank at most $k \ll N$
  - Compute TSVD of all blocks
  - Storage: $N^2 \to \mathcal{O}(kN \log N)$
  - Matvec: $\mathcal{O}\left(N^2\right) \to \mathcal{O}(kN \log N)$

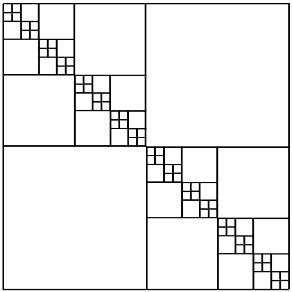




Hierarchical off-diagonal low rank (HODLR) matrices

- HODLR matrix - blocks in partition have ε-rank at most $k \ll N$
  - Compute TSVD of all blocks
  - Storage: $N^2 \to \mathcal{O}(kN \log N)$
  - Matvec: $\mathcal{O}\left(N^2\right) \to \mathcal{O}(kN \log N)$
- Want k fixed (or *slowly growing*) as $N \to \infty$





Application: nonequilibrium Dyson equation from quantum many-body physics

Joint work with Denis Golež of CCQ

Simulating nonequilibrium many-body physics

Simulating nonequilibrium many-body physics

- Simulate *nonequilibrium* dynamics of quantum many-body system

Simulating nonequilibrium many-body physics

- Simulate *nonequilibrium* dynamics of quantum many-body system
  - Electric field driving lattice of hydrogen atoms, with interacting electrons allowed to hop from one site to another

Simulating nonequilibrium many-body physics

- Simulate *nonequilibrium* dynamics of quantum many-body system
  - Electric field driving lattice of hydrogen atoms, with interacting electrons allowed to hop from one site to another
- Solving quantum many-body problem is exponentially hard

Simulating nonequilibrium many-body physics

- Simulate *nonequilibrium* dynamics of quantum many-body system
  - Electric field driving lattice of hydrogen atoms, with interacting electrons allowed to hop from one site to another
- Solving quantum many-body problem is exponentially hard
- Keldysh formalism: write down equation for observables, the single particle two-time Green's functions

Simulating nonequilibrium many-body physics

- Simulate *nonequilibrium* dynamics of quantum many-body system
  - Electric field driving lattice of hydrogen atoms, with interacting electrons allowed to hop from one site to another
- Solving quantum many-body problem is exponentially hard
- Keldysh formalism: write down equation for observables, the single particle two-time Green's functions

$$G_{jk}(t,t') = -i\langle T_C \hat{c}_j(t)\hat{c}_k^\dagger(t')\rangle$$

Simulating nonequilibrium many-body physics

- Simulate *nonequilibrium* dynamics of quantum many-body system
  - Electric field driving lattice of hydrogen atoms, with interacting electrons allowed to hop from one site to another
- Solving quantum many-body problem is exponentially hard
- Keldysh formalism: write down equation for observables, the single particle two-time Green's functions

$$G_{jk}(t,t') = -i\langle T_{\mathcal{C}} \hat{c}_j(t) \hat{c}_k^\dagger(t')\rangle$$

  - Correlation between particle appearing at site k at time t' and particle disappearing from site j at time t

Simulating nonequilibrium many-body physics

- Simulate *nonequilibrium* dynamics of quantum many-body system
  - Electric field driving lattice of hydrogen atoms, with interacting electrons allowed to hop from one site to another
- Solving quantum many-body problem is exponentially hard
- Keldysh formalism: write down equation for observables, the single particle two-time Green's functions

$$G_{jk}(t,t') = -i\langle T_C \hat{c}_j(t)\hat{c}_k^\dagger(t')\rangle$$

  - Correlation between particle appearing at site k at time t' and particle disappearing from site j at time t
- Exponential complexity goes into interaction kernel (self-energy Σ) between G_{jk}, which is truncated at some order

Nonequilibrium Dyson equations

$$(-\partial_\tau - h(0))\, G^{\mathrm{M}}(\tau) - \int_0^\beta d\bar\tau\, \Sigma^{\mathrm{M}}(\tau-\bar\tau) G^{\mathrm{M}}(\bar\tau) = 0 \tag{1}$$

$$(-\mathrm{i}\partial_{t'} - h(t'))\, G^R(t,t') - \int_{t'}^t d\bar t\, G^R(t,\bar t)\Sigma^R(\bar t,t') = 0 \tag{2}$$

$$(\mathrm{i}\partial_t - h(t))\, G^{\rceil}(t,\tau) - \int_0^t d\bar t\, \Sigma^{\mathrm{R}}(t,\bar t) G^{\rceil}(\bar t,\tau) = \int_0^\beta d\bar\tau\, \Sigma^{\rceil}(t,\bar\tau) G^M(\bar\tau-\tau) \tag{3}$$

$$(\mathrm{i}\partial_t - h(t))\, G^<(t,t') - \int_0^t d\bar t\, \Sigma^{\mathrm{R}}(t,\bar t) G^<(\bar t,t') = \int_0^{t'} d\bar t\, \Sigma^<(t,\bar t) G^A(\bar t,t') - \mathrm{i}\int_0^\beta d\bar\tau\, \Sigma^{\rceil}(t,\bar\tau) G_1^{\lceil}(\bar\tau,t') \tag{4}$$

$$G^{\mathrm{M}}(-\tau) = \xi G^{\mathrm{M}}(\beta-\tau) \tag{5}$$

$$G^R(t,t) = -i \tag{6}$$

$$G^{\rceil}(0,\tau) = iG^{\mathrm{M}}(-\tau) = i\xi G^{\mathrm{M}}(\beta-\tau) \tag{7}$$

$$G^<(0,t') = -\overline{G^{\rceil}(t',0)} \tag{8}$$

$$G^{\lceil}(\tau,t) = -\xi\overline{G^{\rceil}(t,\beta-\tau)} \tag{9}$$

$$G^A(t,t') = \overline{G^R(t',t)} \tag{10}$$

$$G^<(t,t') = -\overline{G^<(t',t)}. \tag{11}$$

Nonequilibrium Dyson equations

$$(-\partial_\tau - h(0))\, G^{\mathrm{M}}(\tau) - \int_0^\beta d\bar\tau\, \Sigma^{\mathrm{M}}(\tau-\bar\tau) G^{\mathrm{M}}(\bar\tau) = 0 \quad (1)$$

Focus on this ⟶

$$(-\mathrm{i}\partial_{t'} - h(t'))\, G^R(t,t') - \int_{t'}^t d\bar t\, G^R(t,\bar t)\Sigma^R(\bar t, t') = 0 \quad (2)$$

$$(\mathrm{i}\partial_t - h(t))\, G^{\rceil}(t,\tau) - \int_0^t d\bar t\, \Sigma^{\mathrm{R}}(t,\bar t) G^{\rceil}(\bar t,\tau) = \int_0^\beta d\bar\tau\, \Sigma^{\rceil}(t,\bar\tau) G^M(\bar\tau - \tau) \quad (3)$$

$$(\mathrm{i}\partial_t - h(t))\, G^<(t,t') - \int_0^t d\bar t\, \Sigma^{\mathrm{R}}(t,\bar t) G^<(\bar t,t') = \int_0^{t'} d\bar t\, \Sigma^<(t,\bar t) G^A(\bar t, t') - \mathrm{i}\int_0^\beta d\bar\tau\, \Sigma^{\rceil}(t,\bar\tau) G_1^{\lceil}(\bar\tau, t') \quad (4)$$

$$G^{\mathrm{M}}(-\tau) = \xi G^{\mathrm{M}}(\beta-\tau) \quad (5)$$

⟶

$$G^R(t,t) = -i \quad (6)$$

$$G^{\rceil}(0,\tau) = iG^{\mathrm{M}}(-\tau) = i\xi G^{\mathrm{M}}(\beta-\tau) \quad (7)$$

$$G^<(0,t') = -\overline{G^{\rceil}(t',0)} \quad (8)$$

$$G^{\lceil}(\tau,t) = -\xi\overline{G^{\rceil}(t,\beta-\tau)} \quad (9)$$

$$G^A(t,t') = \overline{G^R(t',t)} \quad (10)$$

$$G^<(t,t') = -\overline{G^<(t',t)}. \quad (11)$$

Volterra integro-differential equation

Volterra integro-differential equation

$$\left(-\mathrm{i}\partial_t - h(t)\right) G^R(t',t) - \int_t^{t'} d\bar{t}\, G^R(t',\bar{t})\Sigma^R(\bar{t},t) = 0$$

$$G^R(t',t') = -\mathrm{i}$$

Volterra integro-differential equation

$$\left(-\mathrm{i}\partial_t - h(t)\right) G^R(t',t) - \int_t^{t'} d\bar{t}\, G^R(t',\bar{t})\Sigma^R(\bar{t},t) = 0$$

$$G^R(t',t') = -\mathrm{i}$$

For each fixed t', after some manipulations...

Volterra integro-differential equation

$$(-\mathrm{i}\partial_t - h(t))\, G^R(t',t) - \int_t^{t'} d\bar{t}\, G^R(t',\bar{t})\Sigma^R(\bar{t},t) = 0$$

$$G^R(t',t') = -\mathrm{i}$$

For each fixed t', after some manipulations...

$$y'(t) + p(t)y(t) + \int_0^t k(t,s)y(s)\, ds = 0$$

$$y(0) = y_0$$

$$t \in [0,T]$$

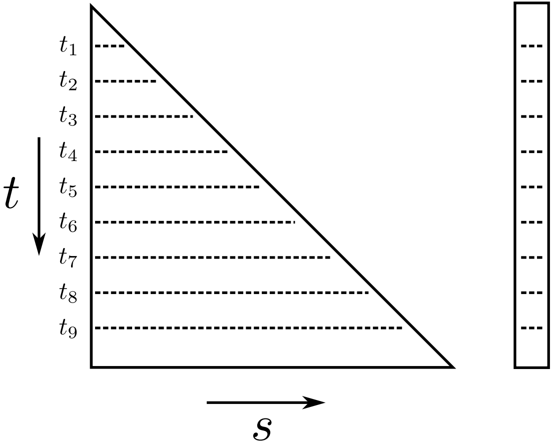
The history integral

The history integral

$$\int_0^{t_n} k(t_n, s) y(s) ds \approx \sum_{m=0}^{n} k(t_n, t_m) \tilde{y}(t_m)$$

The history integral

$$\int_0^{t_n} k(t_n, s) y(s) ds \approx \sum_{m=0}^{n} k(t_n, t_m) \tilde{y}(t_m)$$



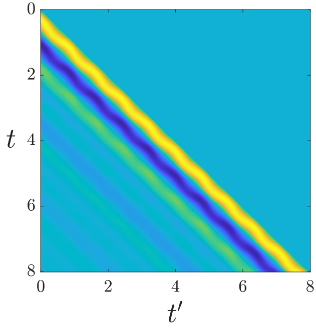
Structure in the kernel Σ^R

Structure in the kernel Σ^R

- Empirical observation: for many physical systems, kernel $\Sigma^R(t, t')$ (which becomes $k(t, s)$) has low rank off-diagonal blocks, as do the solutions $G^R(t, t')$

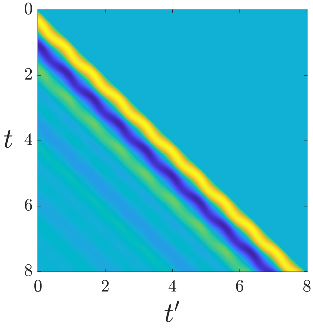
Structure in the kernel Σ^R

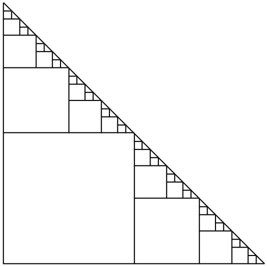
- Empirical observation: for many physical systems, kernel $\Sigma^R(t, t')$ (which becomes $k(t, s)$) has low rank off-diagonal blocks, as do the solutions $G^R(t, t')$



Structure in the kernel Σ^R

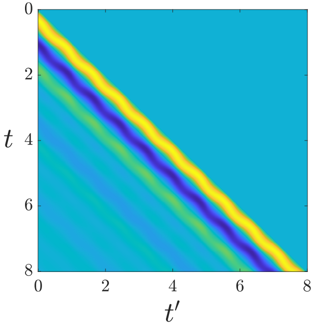
- Empirical observation: for many physical systems, kernel $\Sigma^R(t, t')$ (which becomes $k(t, s)$) has low rank off-diagonal blocks, as do the solutions $G^R(t, t')$

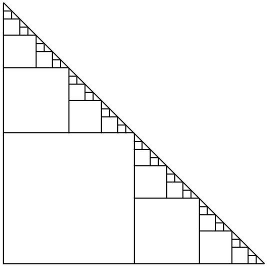




Structure in the kernel Σ^R

- Empirical observation: for many physical systems, kernel $\Sigma^R(t, t')$ (which becomes $k(t, s)$) has low rank off-diagonal blocks, as do the solutions $G^R(t, t')$





- Have efficient method of building HODLR representation of $\Sigma^R(t, t')$

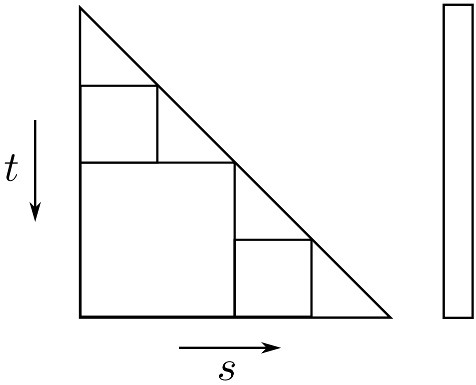
Fast evaluation of history integrals on the fly

Fast evaluation of history integrals on the fly

$$\int_0^{t_n} k(t_n, s) y(s) ds \approx \sum_{m=0}^{n} k(t_n, t_m) \tilde{y}(t_m)$$

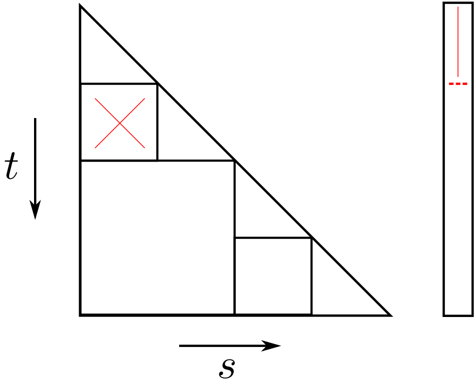
Fast evaluation of history integrals on the fly

$$\int_0^{t_n} k(t_n, s) y(s) ds \approx \sum_{m=0}^{n} k(t_n, t_m) \tilde{y}(t_m)$$



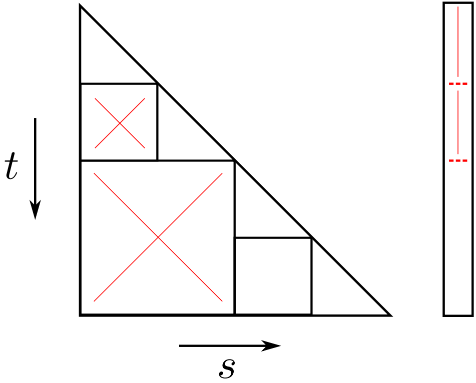
Fast evaluation of history integrals on the fly

$$\int_0^{t_n} k(t_n, s) y(s) ds \approx \sum_{m=0}^{n} k(t_n, t_m) \tilde{y}(t_m)$$



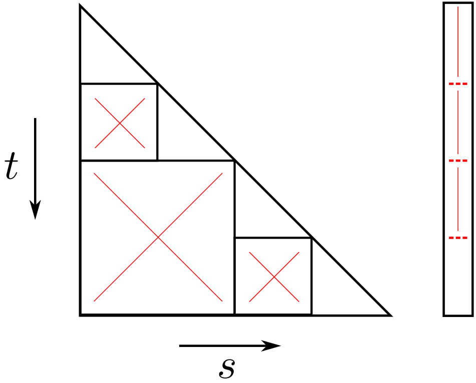
Fast evaluation of history integrals on the fly

$$\int_0^{t_n} k(t_n, s) y(s) ds \approx \sum_{m=0}^{n} k(t_n, t_m) \tilde{y}(t_m)$$



Fast evaluation of history integrals on the fly

$$\int_0^{t_n} k(t_n, s) y(s) ds \approx \sum_{m=0}^{n} k(t_n, t_m) \tilde{y}(t_m)$$



Hierarchical low rank compression for the nonequilibrium Dyson equation

Hierarchical low rank compression for the nonequilibrium Dyson equation

- For systems with HODLR structure, we can increase propagation times substantially:

Hierarchical low rank compression for the nonequilibrium Dyson equation

- For systems with HODLR structure, we can increase propagation times substantially:
  - With N time steps, computational complexity: $\mathcal{O}\left(N^3\right) \to \mathcal{O}\left(N^2 \log N\right)$

Hierarchical low rank compression for the nonequilibrium Dyson equation

- For systems with HODLR structure, we can increase propagation times substantially:
  - With N time steps, computational complexity: $\mathcal{O}\left(N^3\right) \to \mathcal{O}\left(N^2 \log N\right)$
  - Memory complexity: $\mathcal{O}\left(N^2\right) \to \mathcal{O}\left(N \log N\right)$

Hierarchical low rank compression for the nonequilibrium Dyson equation

- For systems with HODLR structure, we can increase propagation times substantially:
  - With N time steps, computational complexity: $\mathcal{O}\left(N^3\right) \to \mathcal{O}\left(N^2 \log N\right)$
  - Memory complexity: $\mathcal{O}\left(N^2\right) \to \mathcal{O}\left(N \log N\right)$
- Falicov-Kimball model: conducting electrons interacting with fixed non-conducting electrons

Hierarchical low rank compression for the nonequilibrium Dyson equation

- For systems with HODLR structure, we can increase propagation times substantially:
  - With N time steps, computational complexity: $\mathcal{O}\left(N^3\right) \to \mathcal{O}\left(N^2 \log N\right)$
  - Memory complexity: $\mathcal{O}\left(N^2\right) \to \mathcal{O}\left(N \log N\right)$
- Falicov-Kimball model: conducting electrons interacting with fixed non-conducting electrons
  - Simulation on a laptop with very long propagation time

Hierarchical low rank compression for the nonequilibrium Dyson equation

- For systems with HODLR structure, we can increase propagation times substantially:
  - With N time steps, computational complexity: $\mathcal{O}\left(N^3\right) \to \mathcal{O}\left(N^2 \log N\right)$
  - Memory complexity: $\mathcal{O}\left(N^2\right) \to \mathcal{O}\left(N \log N\right)$
- Falicov-Kimball model: conducting electrons interacting with fixed non-conducting electrons
  - Simulation on a laptop with very long propagation time
    - 5 months $\to$ 1 day

Hierarchical low rank compression for the nonequilibrium Dyson equation

- For systems with HODLR structure, we can increase propagation times substantially:
  - With N time steps, computational complexity: $\mathcal{O}\left(N^3\right) \to \mathcal{O}\left(N^2 \log N\right)$
  - Memory complexity: $\mathcal{O}\left(N^2\right) \to \mathcal{O}\left(N \log N\right)$
- Falicov-Kimball model: conducting electrons interacting with fixed non-conducting electrons
  - Simulation on a laptop with very long propagation time
    - 5 months $\to$ 1 day
    - 2.2 TB $\to$ 4 GB

Hierarchical low rank compression for the nonequilibrium Dyson equation

- For systems with HODLR structure, we can increase propagation times substantially:
  - With N time steps, computational complexity: $\mathcal{O}\left(N^3\right) \rightarrow \mathcal{O}\left(N^2 \log N\right)$
  - Memory complexity: $\mathcal{O}\left(N^2\right) \rightarrow \mathcal{O}\left(N \log N\right)$
- Falicov-Kimball model: conducting electrons interacting with fixed non-conducting electrons
  - Simulation on a laptop with very long propagation time
    - 5 months → 1 day
    - 2.2 TB → 4 GB
- Confirmed HODLR structure for several other systems; Hubbard model in weak, strong coupling regimes with various driving protocols

Hierarchical low rank compression for the nonequilibrium Dyson equation

- For systems with HODLR structure, we can increase propagation times substantially:
  - With N time steps, computational complexity: $\mathcal{O}\left(N^3\right) \to \mathcal{O}\left(N^2 \log N\right)$
  - Memory complexity: $\mathcal{O}\left(N^2\right) \to \mathcal{O}\left(N \log N\right)$
- Falicov-Kimball model: conducting electrons interacting with fixed non-conducting electrons
  - Simulation on a laptop with very long propagation time
    - 5 months $\to$ 1 day
    - 2.2 TB $\to$ 4 GB
- Confirmed HODLR structure for several other systems; Hubbard model in weak, strong coupling regimes with various driving protocols
- Implementation in NESSi (Nonequilibrium Systems Simulation Library) on the way

What does it mean for you?

What does it mean for you?

- Sparsity is not the only useful matrix structure

What does it mean for you?

- Sparsity is not the only useful matrix structure
- Relatively simple compressibility structures can enable fast algorithms giving cost and memory gains of orders of magnitude

What does it mean for you?

- Sparsity is not the only useful matrix structure
- Relatively simple compressibility structures can enable fast algorithms giving cost and memory gains of orders of magnitude
- It doesn't stop at low rank/hierarchical low rank (DFT matrix is full rank)

What does it mean for you?

- Sparsity is not the only useful matrix structure
- Relatively simple compressibility structures can enable fast algorithms giving cost and memory gains of orders of magnitude
- It doesn't stop at low rank/hierarchical low rank (DFT matrix is full rank)
- It doesn't stop at storage and matrix-vector products