HSS matrix, two levels

Hierarchically semi-separable (HSS) matrix, space padded around each matrix block, which are uniquely identified by dimensions and color.
Constructing the HSS factorization

Sometimes we can obtain $\mathcal{H}_l(A)$ implicitly (e.g. via multipole and Taylor expansions), but general methods can be applied to arbitrary matrices.

- We are most interested in cases when $A$ is sparse or structured, if a dense matrix has HSS structure one should try to not construct the dense matrix explicitly.
- A naive algorithm would construct $\mathcal{H}_l(A)$ from the leaves to the root.
  - Project off-diagonal blocks onto the row/column span of their children.
  - Do QR with column pivoting or truncated SVD.
- This would have cost $O(n^2k)$ and may be sensible for a dense matrix, but projections would not preserve sparsity.
- Randomized methods provide substantially more flexibility.
Interpolative decomposition

To define an efficient and stable randomized algorithm for HSS factorization we will leverage Interpolative decomposition (ID)
- the ID factorization of a matrix $A \in \mathbb{R}^{m \times n}$ has the form
  \[ A \approx W \cdot Z^T \]
  where $W \in \mathbb{R}^{m \times k}$ is a subset of columns of $A$, and $Z \in \mathbb{R}^{n \times k}$
- we can alternatively or additionally enforce that $Z^T$ is a subset of rows of $A$
- a good ID factorization can be found via the SVD $A = UDV^T$ by
  - calculating the statistical column leverage scores of $A$, given by the norms of the columns of $V^T$
  - picking $W$ to be the columns of $A$ that have largest leverage scores and solving linear systems or least squares to obtain $Z$
- the ID decomposition will allow us to construct nested basis matrices in the HSS factorization and can lead to better numerical stability
- Q: why is it reasonable to expect that a low-rank matrix can be approximated using only a subset of its rows and columns?
Randomized HSS factorization

We follow the approach of Martinsson 2011 “A fast randomized algorithm for computing a hierarchically semiseparable representation of a matrix”

- for simplicity, assume $A$ is symmetric, so its off-diagonal blocks are $U_1 U_2^T$ and $U_2 U_1^T$
- we start by picking a random matrix $\Psi \in \mathbb{R}^{n \times (k+10)}$ where 10 is some oversampling and compute $S = A \Psi$
  - the same step is done to compute a randomized low-rank factorization
  - if $A$ is dense it could make sense to make $\Psi$ a randomized DFT-transform (SRFT), while if $A$ is sparse or structured, we can just pick $\Psi$ to be Gaussian random and exploit fast multiplication by $A$
- we construct the HSS factorization for all levels from a single $S$
- we modify $S$ based on the previous (already factorized) levels

\[
\mathcal{R}_l(A, S) = S - \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix} S = \begin{bmatrix} \mathcal{R}_{l-1}(A_{11}, S_1) \\ \mathcal{R}_{l-1}(A_{22}, S_2) \end{bmatrix} - \begin{bmatrix} 0 & A_{12} & 0 & 0 \\ A_{21} & 0 & 0 & 0 \\ 0 & 0 & 0 & A_{34} \\ 0 & 0 & A_{43} & 0 \end{bmatrix} S
\]
HSS factorization via interpolative decomposition

At each recursive step, we can use ID and enforce that both factors are row/column samples

- the two children in the up-sweep recursive tree each provide $k$ sample rows and columns
- we can use the ID decomposition to subsample the best $k$ rows and columns out of $2k$
- in the symmetric case, the rows/columns are the same
- to get the ID, we perform a QR of a matrix of size $n \times (k + 10)$ at each level (where $n$ is the number of rows in $\bar{U}$ at that level)

- to compute $R_l(A, S)$, we need to multiply $S$ by $\bar{U}$ and $\bar{V}$, which requires $O(nk^2)$ work for all nodes at each level
- assuming $k < n/P$, and Cholesky-QR2 is used, the cost after computing $S$ is at least

$$O(nk^2 \log(n)/P \cdot \gamma + k^2 \log(P) \cdot \beta + \log(P)^2 \cdot \alpha)$$
Short pause
Electronic structure calculations

Models of chemical systems calculate the energy of a collection of atoms

- to understand a molecule or molecular system, we seek the configuration in which its energy is lowest
- this configuration is given by where the atom and electrons are
  - electrons occupy *orbitals* around each atom
- atoms are much heavier than electrons, so the configuration of atoms and electrons are almost always considered separately
  - this is the *Born-Oppenheimer approximation*
- the interactions of a system of $n$ electrons are encoded in a *Hamiltonian operator* $H$
- the *wavefunction* $\Psi(r_1, \ldots, r_n)$ and its energy $E$ is the eigenfunction of the Hamiltonian with the smallest eigenvalue

$$H\Psi(r_1, \ldots, r_n) = E \Psi(r_1, \ldots, r_n)$$

- $\Psi(r_1, \ldots, r_n)$ is a complex function of all electron coordinates
- $\psi^*(r_1, \ldots, r_n)\psi(r_1, \ldots, r_n)$ gives the probability of observing the electrons at $r_1, \ldots, r_n$
The Schrödinger equation describes electronic interactions

- most often, a time-independent, nonrelativistic form is used
- in this simplified case the $N$-particle Hamiltonian has the form

$$H = -\frac{1}{2m} \sum_{i=1}^{N} \nabla_i^2 + \sum_{i=1}^{N} V(r_i) + \sum_{i=1}^{N} \sum_{j<i} U(r_i, r_j)$$

- the one-particle component $V(r_i)$ encodes interactions between electrons and atoms
- the two-particle component $U(r_i, r_j)$ encodes electron–electron interactions
- $\Psi$ is generally a function of all electrons, to obtain an approximate solution a simpler ansatz is often used
Density Function Theory (DFT)

DFT is a common method for electronic structure calculations

- it uses a wavefunction ansatz of a *Hartree product* of $N$ single-particle wavefunctions

$$\Psi(r_1, \ldots, r_n) \approx \Psi_1(r_1) \cdots \Psi_N(r_n)$$

- the electron (probability) density given this ansatz is

$$n(r) = \sum_{i=1}^{n} \int \cdots \int (\Psi^* \Psi)(r_1, \ldots r_{i-1}, r, r_{i+1}, \ldots r_n) \, dr_1 \cdots dr_{i-1} dr_{i+1} \cdots dr_n \approx \sum_{i=1}^{n} \Psi_i^*(r) \Psi_i(r)$$

- *Hohenberg–Kohn theorem*: there exists a unique ‘functional’ $F$ with

$$E = F(n(r))$$
Kohn–Sham equations

DFT finds the single-electron wavefunctions using *Kohn–Sham equations*

\[
\left[ -\frac{1}{2m} \nabla^2 + V(r) + V_H(r) + V_{XC}(r) \right] \psi_i(r) = \varepsilon_i \psi_i(r)
\]

- these equations replace the many-body Hamiltonian operator
- the two-electron interactions are now approximated by a potential between each single-electron wavefunction and the density \( n(r) \)
- \( V_H(r) \) is the Hartree potential holding Coulomb repulsion
- \( V_{XC}(r) \) is an approximation to the exchange-correlation potential (incl. Pauli exclusion)
- the key part of the aforementioned functional is \( V_{XC}(r) \)
- there is no known general representation for exchange-correlation and much reason to believe there is no simple general form
- various approximations make sense in different application contexts
Solving the Kohn–Sham equations

The Kohn–Sham equations depend on the density and define the single particle wavefunctions

- the electron density is a function of the single particle wavefunctions
- so DFT is solved iteratively

1. define an initial guess for the density $n^{(0)}(\mathbf{r})$
2. solve the Kohn–Sham equations defined by $n^{(j)}(\mathbf{r})$ to get $\Psi^{(j)}(\mathbf{r})$
3. calculate a new Kohn–Sham electron density

$$ n^{(j+1)}(\mathbf{r}) = \sum_{i=1}^{n} \Psi^{(j)}(\mathbf{r})^* \Psi^{(j)}(\mathbf{r}) $$

- a basis must be defined for the spatial domain to get a numerical representation of $n(\mathbf{r})$
  - Gaussian basis functions are often used and have numerical advantages
  - basis functions can also be localized around orbitals, enabling sparse representations of the density and lower-order scaling
Computing DFT

Typically $m = O(n)$ basis functions are selected

- each single-electron wavefunction is then a vector of dimension $m$
- there are many ways to formulate and solve the Kohn-Sham equations numerically
- with a Gaussian basis, they are often solved using FFT and involve
  - matrix multiplication and QR with matrices of dimension $m \times n$
  - FFT on each wavefunction vector
  - eigendecomposition of $n \times n$ symmetric matrix
- they can also be done in real-space using an $m \times m$ eigendecomposition
- recent work on methods (see CP2K application) that achieve $O(n)$ scaling by computing the sign function of $m \times m$ sparse matrix (using localized basis-functions)
  - sign function can be found by repeated squaring like $D_{j+1} = 3D_j^2 - 2D_j^3$
- the communication cost of most DFT methods is easily derived from the complexity of dense linear algebra routine
Hartree–Fock method

The Hartree–Fock (HF) method provides a more accurate representation of electron exchange.

- HF is still a mean-field treatment that does not treat electron–electron interactions explicitly.
- HF uses a Slater determinant as a wavefunction ansatz:

\[
\psi(r_1, \ldots, r_n) \approx \det \begin{pmatrix}
\psi_1(r_1) & \cdots & \psi_1(r_2) \\
\vdots & \ddots & \vdots \\
\psi_n(r_1) & \cdots & \psi_n(r_n)
\end{pmatrix}
\]

- This is the same as taking all possible coordinate permutations of the Hartree product and assigning signs to each term so that the result is antisymmetric to any permutation of a coordinate pair.
- The antisymmetry allows the wavefunction to satisfy the Pauli exclusion principle.
Self Consistent Field iteration

HF is solved by the Self Consistent Field (SCF) iteration, similar to DFT

- an \( m \)-dimensional spatial basis again needs to be introduced
- an \( \times m \) density matrix \( \mathbf{D} \) is computed iteratively
- the *Fock matrix* is given by

\[
F_{ij} = H_{ij}^{\text{core}} + \sum_{kl} D_{kl}(2(ij|kl) - (ik|jl))
\]

where \( H_{ij}^{\text{core}} \) is the core-Hamiltonian and \((ij|kl)\) are the electron–repulsion integrals

- the matrix \( \mathbf{C} \) of \( k \) eigenvectors of \( \mathbf{F} \) with the smallest eigenvalues defines \( \mathbf{D} \)

\[
\mathbf{D} = \mathbf{C} \mathbf{C}^T
\]

- SCF iteratively computes \( \mathbf{F} \) from \( \mathbf{D} \) then a new \( \mathbf{D} \) from \( \mathbf{F} \)
Electron-repulsion integral computation

A key difference between DFT and HF is the need to compute the electron–repulsion integrals

- these are generally screened and only a subset is computed
- an integral \((ij|kl)\) is derived from \(D_{ab}\) where \(\{a, b\} \in \{i, j, k, l\}\) and contributes to each \(F_{ab}\)
- both \(F\) and \(D\) are symmetric so we only care about \(\binom{4}{2} = 6\) permutations
- Q: if we compute a 4D block of \((ij|kl)\) of size \(O(M)\) how many entries of \(F\) and \(D\) do we touch?
- A: the projections from the 4D block onto 2D subspaces are of size \(\Theta(\sqrt{M})\)
- thus computing \(O(n^4)\) ERI integrals with \(P\) processors can be done with \(O(n^2/\sqrt{P})\) communication
- to compute a screened subset of \(\Theta(M)\) integrals, a generalized Loomis-Whitney theorem can be used to show that \(\Omega(\sqrt{M/P})\) communication is generally required
Configuration interaction

Hartree-Fock represents the wavefunction as a single Slater determinant

- given a basis set of $m > n$ functions (orbitals), we can define $\binom{m}{n}$ Slater determinants of $n$-electrons, which ‘occupy’ orbitals
- full configuration-interaction (full CI) works with a maximal set of determinants
- by computing the eigendecomposition of the resulting matrix, an exact solution can be obtained for the electronic Schrödinger equation given the basis set
- thus full-CI method is exact in the ‘basis set limit’, but has exponential cost
- other CI methods select a subset of determinants by using knowledge of the electronic system
Moller-Plesset perturbation theory

Moller-Plesset perturbation theory, modifies the Hamiltonian slightly to take into account some ‘excited-states’ configurations

- considering single-electron excitations has no effect on the energy, as shown by the *Brillouin theorem*
- thus, MP2 and MP3, which consider second and third order perturbations are the primary methods employed
- MP2 can be computed directly from the ERI tensor as a correction, requiring $O(n^4)$ cost
- MP3 requires a tensor contraction between two order four tensors, requiring $O(n^6)$ cost
- the dominant part of the cost in MP3 is the tensor contraction, which can be done by matrix-matrix multiplication
Coupled-cluster

A more computationally robust alternative to CI is presented by Coupled-cluster (CC) methods

- CC methods try to take into account electron correlation, by taking into account all possible excitations of $k$ electrons
- there is a hierarchy of coupled-cluster methods
  - CCSD: (singles and doubles) $k = 2$, $O(n^6)$ cost
  - CCSDT: (singles, doubles, and triples) $k = 3$, $O(n^8)$ cost
  - CCSDTQ: (singles, double, triples, and quadruples) $k = 4$, $O(n^{10})$ cost
- coupled cluster methods use an wavefunction ansatz of the form $\Psi \approx e^{T_1 + T_2 + \cdots} \Psi_0$ where $\Psi_0$ is the HF slater determinant
- the exponential is expanded in polynomial form and truncated, resulting in a set of tensor contractions that define possible electron state transitions
- the methods are dominated by matrix-multiplication (tensor contractions)
- the tensors have a lot of symmetry and sometimes are sparse or low-rank