Introduction into hierarchical matrix technique

Alexander Litvinenko
KAUST, SRI-UQ Center
http://sri-uq.kaust.edu.sa/

www.hlib.org
www.hlibpro.com
After defense, Prof. W. Hackbusch and scientific computing group, MPI für angewandte Mathematik, Leipzig
PostDoc 2007-2013, Prof. H.G. Matthies, Scientific computing group, TU Braunschweig
1. Motivation
2. Low-rank matrices
3. Cluster Tree, Block Cluster Tree and Admissibility condition
4. 1D BEM example
5. Hierarchical matrices: cost and storage
6. Two applications

\[ Ax = b \]

Iterative methods: Jacobi, Gauss-Seidel, SOR, ...
Direct solvers: Gaussian elimination, domain decompositions, LU,...

Cost of \( A^{-1} \) is \( \mathcal{O}(n^3) \), number of iteration is proportional to \( \sqrt{\text{cond}(A)} \).

If \( A \) is structured (diagonal, Toeplitz, circulant) then can apply e.g. FFT, but if not?

What if you need not only \( x = A^{-1}b \), but \( f(A) \) (e.g. \( A^{-1} \), \( \exp A \), \( \sin A \), \( \text{sign} A \), ...)?
**Figure:** The $\mathcal{H}$-matrix approximation of the stiffness matrix of the Poisson problem (left) and its inverse (right). The dark blocks are dense matrices. The light blocks are low-rank matrices with maximal rank $k_{max} = 5$. 
$M \in \mathbb{R}^{n \times m}, \ U \approx \tilde{U} \in \mathbb{R}^{n \times k}, \ V \approx \tilde{V} \in \mathbb{R}^{m \times k}, \ k \ll \min(n, m)$.

The storage $\tilde{M} = \tilde{U}\tilde{\Sigma}\tilde{V}^T$ is $k(n + m)$ instead of $n \cdot m$ for $M$ represented in the full matrix format.

**Figure:** Reduced SVD, only $k$ biggest singular values are taken.
1. Build cluster tree $T_I$ and block cluster tree $T_{I \times I}$.
2. For each \((t \times s) \in T_I \times I\), \(t, s \in T_I\), check admissibility condition
\[
\min\{\text{diam}(Q_t), \text{diam}(Q_s)\} \leq \eta \cdot \text{dist}(Q_t, Q_s).
\]

if(adm=\text{true}) then \(M|_{t \times s}\) is a rank-\(k\) matrix block
if(adm=\text{false}) then divide \(M|_{t \times s}\) further or define as a dense matrix block, if small enough.

**Resume:** Grid \(\rightarrow\) cluster tree \((T_I) +\) admissibility condition \(\rightarrow\) block cluster tree \((T_I \times I)\) \(\rightarrow\) \(\mathcal{H}\)-matrix \(\rightarrow\) \(\mathcal{H}\)-matrix arithmetics.
Where does the admissibility condition come from?

Let $B_1, B_2 \subset \mathbb{R}^d$ are compacts, and $\chi(x, y)$ is defined for $(x, y) \in B_1 \times B_2$ with $x \neq y$.

Let $\mathcal{K}$ be an integral operator with an asymptotic smooth kernel $\chi$ in the domain $B_1 \times B_2$:

$$(\mathcal{K} \nu)(x) = \int_{B_2} \chi(x, y) \nu(y) dy \quad (x \in B_1).$$

Suppose that $\chi^{(k)}(x, y)$ is an approximation of $\chi$ in $B_1 \times B_2$ of the separate form:

$$\chi^{(k)}(x, y) = \sum_{\nu=1}^{k} \varphi^{(k)}_{\nu}(x) \psi^{(k)}_{\nu}(y),$$

where $k$ is the rank of separation.

Then $\|\chi - \chi^{(k)}\|_{\infty, B_1 \times B_2} \leq c_1 \left[ \frac{c_2 \min\{\text{diam}(B_1), \text{diam}(B_2)\}}{\text{dist}(B_1, B_2)} \right]^{k}$. 
Consider the following integral equation

\[ \int_{0}^{1} \log |x - y| U(y) dy = F(x), \quad x \in (0, 1). \]

After discretisation by Galerkin’s method we obtain

\[ \int_{0}^{1} \int_{0}^{1} \phi_i(x) \log |x - y| U(y) dy dx = \int_{0}^{1} \phi_i(x) F(x) dx, \quad 0 \leq i < n, \]

in the space \( V_n := \text{span}\{\phi_0, \ldots, \phi_{n-1}\} \), where \( \phi_i, \ i = 1, \ldots, n - 1 \), are some basis functions in BEM. The discrete solution \( U_n \) in the space \( V_n \) is \( U_n := \sum_{j=0}^{n-1} u_j \phi_j \) with \( u_j \) being the solution of the linear system.
\[ Gu = f, \quad G_{ij} := \int_0^1 \int_0^1 \phi_i(x) \log |x - y| \phi_j(y) dy \, dx, \quad f_i := \int_0^1 \phi_i(x) F(x) \, dx. \] (1)

We replace the kernel function \( g(x, y) = \log |x - y| \) by a degenerate kernel

\[
\tilde{g}(x, y) = \sum_{\nu=0}^{k-1} g_{\nu}(x) h_{\nu}(y).
\] (2)

Then we substitute \( g(x, y) = \log |x - y| \) in (1) for \( \tilde{g}(x, y) \)

\[
\tilde{G}_{ij} := \int_0^1 \int_0^1 \phi_i(x) \sum_{\nu=0}^{k-1} g_{\nu}(x) h_{\nu}(y) \phi_j(y) dy \, dx.
\]

After easy transformations

\[
\tilde{G}_{ij} := \sum_{\nu=0}^{k-1} \left( \int_0^1 \phi_i(x) g_{\nu}(x) dx \right) \left( \int_0^1 h_{\nu}(y) \phi_j(y) dy \right).
\]
Now, all admissible blocks $G|_{(t,s)}$ can be represented in the form

$$G|_{(t,s)} = AB^T, \quad A \in \mathbb{R}^{t \times k}, \quad B \in \mathbb{R}^{s \times k},$$

where the entries of the factors $A$ and $B$ are

$$A_{i\nu} := \int_0^1 \phi_i(x) g_\nu(x) dx, \quad B_{j\nu} := \int_0^1 \phi_j(y) h_\nu(y) dy.$$

We use the fact that the basis functions are local and obtain for all inadmissible blocks:

$$\tilde{G}_{ij} := \int_{i/n}^{(i+1)/n} \int_{j/n}^{(j+1)/n} \log |x - y| dy dx.$$
Let $\mathcal{H}(T_{I \times J}, k) := \{ M \in \mathbb{R}^{I \times J} \mid \text{rank}(M |_{t \times s}) \leq k \text{ for all admissible leaves } t \times s \text{ of } T_{I \times J} \}$, $n := \max(|I|, |J|, |K|)$.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Sequential Compl.</th>
<th>Parallel Complexity (R.Kriemann 2005)</th>
</tr>
</thead>
<tbody>
<tr>
<td>building$(M)$</td>
<td>$N = \mathcal{O}(n \log n)$</td>
<td>$\frac{N}{p} + \mathcal{O}(</td>
</tr>
<tr>
<td>storage$(M)$</td>
<td>$N = \mathcal{O}(kn \log n)$</td>
<td>$N$</td>
</tr>
<tr>
<td>$M \chi$</td>
<td>$N = \mathcal{O}(kn \log n)$</td>
<td>$\frac{N}{p} + \frac{n}{\sqrt{p}}$</td>
</tr>
<tr>
<td>$\alpha M' \oplus \beta M''$</td>
<td>$N = \mathcal{O}(k^2 n \log n)$</td>
<td>$\frac{N}{p}$</td>
</tr>
<tr>
<td>$\alpha M' \odot M'' \oplus \beta M$</td>
<td>$N = \mathcal{O}(k^2 n \log^2 n)$</td>
<td>$\frac{N}{p} + \mathcal{O}(C_{sp}(T)</td>
</tr>
<tr>
<td>$M^{-1}$</td>
<td>$N = \mathcal{O}(k^2 n \log^2 n)$</td>
<td>$\frac{N}{p} + \mathcal{O}(n n_{\text{min}}^2)$</td>
</tr>
<tr>
<td>LU</td>
<td>$N = \mathcal{O}(k^2 n \log^2 n)$</td>
<td>$N$</td>
</tr>
<tr>
<td>$\mathcal{H}$-LU</td>
<td>$N = \mathcal{O}(k^2 n \log^2 n)$</td>
<td>$\frac{N}{p} + \mathcal{O}(\frac{k^2 n \log^2 n}{n^{1/d}})$</td>
</tr>
</tbody>
</table>
Example: $\mathcal{H}$-matrix approximation of covariance matrices

Can be applied for efficient computing the Karhunen-Loeve Expansion (need to solve a large eigenvalue problem), and for generating large random fields (Cholesky and MV product)
Examples of $\mathcal{H}$-matrix approximates of $\text{cov}(x, y) = e^{-2|x-y|}$

**Figure:** $\mathcal{H}$-matrix approximations $\in \mathbb{R}^{n \times n}$, $n = 32^2$, with standard (left) and weak (right) admissibility block partitionings. The biggest dense (dark) blocks $\in \mathbb{R}^{n \times n}$, max. rank $k = 4$ left and $k = 13$ right.
Dependence of the computational time and storage requirements of $\mathbf{C}_H$ on the rank $k$, $n = 32^2$.

<table>
<thead>
<tr>
<th>$k$</th>
<th>time (sec.)</th>
<th>memory</th>
<th>$\frac{|\mathbf{C} - \mathbf{C}_H|_2}{|\mathbf{C}|_2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.04</td>
<td>$2e + 6$</td>
<td>$3.5e - 5$</td>
</tr>
<tr>
<td>6</td>
<td>0.1</td>
<td>$4e + 6$</td>
<td>$1.4e - 5$</td>
</tr>
<tr>
<td>9</td>
<td>0.14</td>
<td>$5.4e + 6$</td>
<td>$1.4e - 5$</td>
</tr>
<tr>
<td>12</td>
<td>0.17</td>
<td>$6.8e + 6$</td>
<td>$3.1e - 7$</td>
</tr>
<tr>
<td>17</td>
<td>0.23</td>
<td>$9.3e + 6$</td>
<td>$6.3e - 8$</td>
</tr>
</tbody>
</table>

The time for dense matrix $\mathbf{C}$ is 3.3 sec. and the storage $1.4e + 8$ (Bytes).
Examples of covariance functions

The random field requires to specify its spatial correlation structure

\[
\text{cov}_f(x, y) = \mathbb{E}[(f(x, \cdot) - \mu_f(x))(f(y, \cdot) - \mu_f(y))],
\]

where \(\mathbb{E}\) is the expectation and \(\mu_f(x) := \mathbb{E}[f(x, \cdot)]\).

Let \(h = \sqrt{\sum_{i=1}^{3} h_i^2 / \ell_i^2}\), where \(h_i := x_i - y_i\), \(\ell_i\) are correlation lengths.

Gaussian \(\text{cov}(h) = \sigma^2 \cdot \exp(-h^2)\),

exponential \(\text{cov}(h) = \sigma^2 \cdot \exp(-h)\),

spherical

\[
\text{cov}(h) = \begin{cases} 
\sigma^2 \cdot \left(1 - \frac{3}{2} \frac{h}{h_r} + \frac{1}{2} \frac{h^3}{h_r^3}\right) & \text{for } 0 \leq h \leq h_r, \\
0 & \text{for } h > h_r.
\end{cases}
\]
1. **Matrix exponential allows us to solve ODEs**

\[
\dot{x}(t) = Ax(t), \quad x(0) = x_0, \quad \rightarrow x(t) = \exp(tA)x_0
\]

2. **Other matrix function:** use representation by the Cauchy integral

\[
f(A) = \frac{1}{2\pi i} \oint_{\Gamma} f(t)(A - tl)^{-1} dt
\]

and exponentially convergent quadrature rule [Hackbusch, TR 4/2005, MPI]

\[
f(A) \approx \sum_{j=1}^{k} w_j f(t_j)(A - t_j l)^{-1}
\]

to obtain an approximation.
Matrix equations

(Arises in the context of optimal control problems)

(Lyapunov) \[ AX + XA^T + C = 0, \] \[ (3) \]

(Sylvester) \[ AX - XB + C = 0, \] \[ (4) \]

(Riccati) \[ AX + XA^T - XFX + C = 0. \] \[ (5) \]

where \( A, B, C, F \) are given matrices and \( X \) is the solution (a matrix).
Theorem

Let $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{m \times m}$, and let $\sigma(A) < \sigma(B)$, then

$$X := \int_{0}^{\infty} \exp(tA)C \exp(-tB)dt$$

(6)

solves the Sylvester equation.

And with some special quadrature formulas [Hackbusch, TR 4/2005, MPI]

$$X_k := \sum_{j=0}^{k} w_j \exp(t_j A)C \exp(-t_j B).$$

(7)

fulfills $\|X - X_k\| < C_{A,B}\|C\|_2 \exp(-\sqrt{k})$. 
Lemma

If \( C \in \mathcal{H}(T, k_C) \), \( \exp(t_j A) \approx A_j \in \mathcal{H}(T, k_A) \), 
\( \exp(-t_j B) \approx B_j \in \mathcal{H}(T, k_B) \), for all \( t_j \) (see more in [Gavrilyuk et al 2002, and Grasedyck et al 2003]) then

\[
X_k \in \mathcal{H}(T, k_X), \quad k_X = O(kp^2 \max\{k_A, k_B, k_C\}). \tag{8}
\]

Numerically \( X_k \) can be found by a) matrix exponential b) multigrid c) meta method (ADI); d) matrix sign function (see more \( \mathcal{H} \)-matrix lecture notes, Ch. 10).
Future work

1. Parallel $\mathcal{H}$-matrix implementation on multicore systems and on GPUs
2. $\mathcal{H}$-matrix approximation of large covariance matrices (Matérn)
3. Determinant and trace of an $\mathcal{H}$-matrix

Kullback-Leibler divergence (KLD) $D_{KL}(P\|Q)$ is measure of the information lost when distribution $Q$ is used to approximate $P$:

$$D_{KL}(P\|Q) = \int_{-\infty}^{\infty} p(x) \ln \frac{p(x)}{q(x)} dx,$$

where $p, q$ densities of $P$ and $Q$. For multivariate normal distributions $(\mu_0, \Sigma_0)$ and $(\mu_1, \Sigma_1)$

$$2D_{KL}(\mathcal{N}_0\|\mathcal{N}_1) = \left( tr(\Sigma_1^{-1}\Sigma_0) + (\mu_1 - \mu_0)^T \Sigma_1^{-1}(\mu_1 - \mu_0) - k - \ln \left( \frac{\det \Sigma_0}{\det \Sigma_1} \right) \right)$$
Conclusion

+ Complexity and storage is $\mathcal{O}(k^r n \log^q n)$, $r = 1, 2, 3$, $q = 1, 2$
+ Allow to compute $f(A)$ efficiently for some class of functions $f$
+ Many examples: FEM 1D, 2D and 3D, BEM 1D, 2D and 3D, Lyapunov, Sylvester, Riccati matrix equations
+ Well appropriate to be used as a preconditioner (for iterative methods)
+ There are sequential (www.hlib.org) and parallel (www.hlibpro.com) libraries
+ There are A LOT of implementation details!
  - Not so easy implementation
  - Can be large constants in 3D