Efficient Multikernel Hierarchical Compression for Boundary Element Matrices

Damiano Franzò, Simon B. Adrian, Adrien Merlini, and Francesco P. Andriulli

Abstract – We present a new scheme that allows for the compression of operators of the combined field integral equation in the low-frequency regime using a hierarchical decomposition that leverages a unified pseudoskeleton approximation of both the single- and the double-layer Green’s function at the quadrature points. Compared with a standard adaptive cross approximation, the numerical results show a reduced memory consumption without sacrificing computational run time.

1. Introduction

The boundary element method (BEM) gives rise to dense matrices, which are typically hierarchically compressed by so-called fast methods, such as the fast multipole method [1] or the adaptive cross approximation (ACA) [2, 3] to yield at most $\mathcal{O}(N)$ and $\mathcal{O}(N \log N)$ global complexity, respectively, for electrically small problems, where $N$ is the number of unknowns. What makes the ACA appealing is its purely algebraic nature, which allows for a straightforward combination with existing BEM codes. A less commonly used algorithm, which is also of purely algebraic nature, is the pseudoskeleton approximation, also known as CUR [4, 5].

Regardless of whether the ACA or the CUR is used, when the underlying integral equation formulation consists of several operators, one has to either store and compress a single-system matrix of the linear combination of the operators or to separately discretize, store, and compress the system matrix of each operator. Although the first approach is more memory efficient, it makes it difficult, if not impossible, to allow for a combination with advanced preconditioning strategies [6, 7]. The second approach, on the other hand, results in an increased memory consumption, a slower execution of matrix vector products, and an increased compression time. Moreover, if material parameters are changed, the entire matrix must be rebuilt.

In this article, we present a new scheme that is compatible with advanced preconditioning strategies, that is, where each operator has its own discretization, however, by using a CUR-based single hierarchical decomposition leveraging a pseudoskeleton approximation of the single- and double-layer Green’s function kernel for electrically small problems at the quadrature points, the memory cost is significantly reduced compared with traditional approaches. We achieve this by compressing the Green’s function sampled at the quadrature points instead of compressing the weakly tested operators (e.g., as in a standard ACA approach) together with using a CUR-based decomposition. The latter allows to efficiently compute rows and columns of the compressed submatrices on the fly, so only a small portion of the system matrices needs to be stored, thus reducing the memory consumption. The weak discretization of the operators are then reconstructed through the usage of sparse mappings, which contain the Gaussian quadrature weights and the evaluated basis functions. The hybrid cross approximation (HCA), introduced in [8], leverages the compression of an interpolated kernel-generating function to efficiently handle single- and double-layer matrices. Although the objectives of the method we propose and those of the HCA exhibit some similarities in philosophy, compressing the Green’s function matrix at the quadrature points as we propose allows for straightforward integration into existing $\mathcal{O}$-matrix-enabled frameworks and remains a purely matrix-based technique. Although the method we present targets electrically small problems, extensions to handle electrically larger problems are foreseen, by properly leveraging advanced preconditioning and compression techniques, and will be the subject of future contributions.

2. Background and Notation

The Helmholtz equation

$$\nabla^2 \varphi(x) + k^2 \varphi(x) = 0 \quad x \in \Omega^+ \subset \mathbb{R}^d$$

with $d \in \{2, 3\}$ is used to model two-dimensional (2D) and three-dimensional acoustic and 2D electromagnetic scattering and radiation problems, where $k$ is the wavenumber, $\Omega$ denotes the scatterer, $\Omega^+ = \mathbb{R}^d \backslash \Omega$, and in the context of acoustics, $\varphi$ denotes the total pressure field. For the impenetrable scatterer, we have the sound soft boundary condition $\varphi = \varphi^{\text{scat}} + \varphi^{\text{inc}} = 0$ on the boundary $\Gamma = \partial \Omega$, where $\varphi^{\text{inc}}$ is the incident and $\varphi^{\text{scat}}$ is the scattered pressure field. To solve the sound soft scattering...

Manuscript received 11 July 2023. This work was supported, in part, by the European Research Council through the European Union’s Horizon 2020 research and innovation program (grant 724846, project CEREBRO).

Damiano Franzò and Francesco Andriulli are with Dipartimento di Elettronica e Telecomunicazioni (DET), Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy; e-mail: damiano.franzo@polito.it, francesco.andriulli@polito.it.

Simon Adrian is with Fakultät für Informatik und Elektrotechnik, Universität Rostock, 18059 Albert-Einstein-Str. 2, Rostock, Germany; e-mail: simon.adrian@uni-rostock.de.

Adrien Merlini is with Microwave department, IMT Atlantique, Technopôle Brest-Îroise, 29238 Brest cedex 03, France; e-mail: adrien.merlini@imt-atlantique.fr.
problem, we use the Brakhage–Werner equation combined field integral equation (CFIE) formulation \[9, 10]\n
\[(\mathcal{G}_{\mathcal{R}} f - j \eta \mathcal{G}_{\mathcal{R}}) \psi = -\varphi^{\text{inc}} \text{ for } r \in \Gamma \]  \tag{2}

where \(\psi\) is a surface scalar field, \(\mathcal{R}\) is the imaginary unity, \((\mathcal{G}_{\mathcal{R}} f)(x) = f(x)\) is the identity operator, and \(\eta\) is the coupling parameter of the single-layer \((\mathcal{G}_{\mathcal{R}} f)(x) = \int_{\Gamma} G(x, y) f(y) \, ds(y)\) and the double-layer operator \((\mathcal{G}_{\mathcal{R}} f)(x) = \int_{\Gamma} \partial_n G(x, y) f(y) \, ds(y)\), where \(G(x, y) = 1/(4\pi |x - y|)\) is the free-space Green’s function associated with (1). We will refer to \(G(x, y)\) as kernel of \(\mathcal{G}\) and to \(\partial_n G(x, y) = \nabla_y G(x, y) \cdot \mathbf{n}'\) as the kernel of \(\partial_n\). Once \(\psi\) is found, the scattered field \(\varphi^{\text{sc}}\) can be computed via \(\varphi^{\text{sc}} = (\mathcal{G}_{\mathcal{R}} - j \eta \mathcal{G}_{\mathcal{R}}) \psi\).

Using the BEM, we approximate \(\psi \approx \sum_{i=1}^{N} v_n \lambda_n\) in terms of piecewise linear basis functions \(\{\lambda_n\}_{n=1}^{N}\), where \(v_n\) is the unknown expansion coefficient and

\[
\lambda_n(x) = \begin{cases} 
1 & \text{for } x = v_n \\
0 & \text{for } x = v_{p \neq n} 
\end{cases} \tag{3}
\]

where \(v_n\) is the \(n\)th vertex of the mesh. Leveraging a Galerkin approach, the discretized Brakhage–Werner equation becomes

\[
\left(1/2 [I] + [D] - j \eta [S]\right) [v] = -[-\varphi^{\text{inc}}] \tag{4}
\]

where \([D]_{\text{nn}} = (\lambda_m, \varphi^{\text{inc}})_{L^2}, [S]_{\text{nn}} = (\lambda_m, \lambda^\prime_m)_{L^2}, [I]_{\text{nn}} = (\lambda_m, \lambda^\prime_m)_{L^2}, [\varphi^{\text{inc}}]_{\text{mm}} = (\lambda_m, \varphi^{\text{inc}})_{L^2}, \) and \([v]_{\text{nn}} = v_n\) are the unknown expansion coefficients.

The matrices \([S]\) and \([D]\) are fully populated, resulting in \(O(N^2)\) complexity for memory storage and as complexity for the cost of a single matrix-vector product (MVP). The goal of any fast method is, at most, to obtain a quasilinear complexity, that is, to reduce the storage and the MVP time complexity to \(O(N \log N)\).

To obtain such a complexity, \(\mathcal{R}\) matrices can be used \([11–14]\), where the fact is leveraged that the matrices \([S]\) and \([D]\) have submatrices \([B] \in \mathbb{C}^{1 \times 1}\) formed by rows and columns \(\subset \{1, \ldots, N\}\), admitting a low-rank form (LRF), where the rank is \(O(1)\) in the low-frequency regime, and we used MATLAB notation to denote the submatrices. In this article, we leverage the pseudoskeleton form \([B](t, s) \approx [C][U][R]\), where \([C] \in \mathbb{R}^{n \times r}, [U] \in \mathbb{R}^{r \times r}, [R] \in \mathbb{R}^{r \times m}\) and \(r\) is the \(\varepsilon\) rank of the decompositions satisfying an error bound \(\|B(t, s) - [C][U][R]\|_2 \leq \varepsilon\), lowering the cost of the MVP and storage complexity if \(r \times (n + m) < n \times m\). This form arises typically from sampling rows \([R]\) and columns \([C]\) according to a certain pattern. The matrix \([U]\) is usually the inverse or pseudoinverse of the intersection between the sampled rows and columns. To identify suitable submatrices, a partitioning algorithm such as an octree must be used, which creates a hierarchical partition of row and column indices (for details, see \([12]\)). In the following, we refer to a cluster as a set of either row or column indices; thus, a row and a column cluster together correspond to a specific submatrix.

For compressible \([B](t, s)\), the associated integrands are nonsingular because the support of the expansion and of the testing functions are well separated. Because the integrand is nonsingular, a Gauss quadrature can be used for both the integrals over the domain of the testing and of the expansion function. Let \(\mathcal{G}\) be an integral operator with kernel \(q(x, y)\). We indicate with \([Q_{\text{ns}}]\) and \([Q_{\text{s}}]\) the nonsingular and the singular interactions so that \([Q_{\text{ns}}] + [Q_{\text{s}}] = [Q]\). We note that

\[
[Q_{\text{ns}}]_{ij} = \langle \lambda_i, \lambda_j \rangle \approx \sum_{l} w_l \lambda(p_l) \sum_{j} w_j \lambda(p_j) q(p_l - p_j) \tag{5}
\]

where \(p_l\) and \(p_j\) are the quadrature points associated with the integration over supp \((\lambda_i)\) and supp \((\lambda_j)\). By leveraging (5), we can rewrite \([Q_{\text{ns}}]\) as a composition of matrix multiplications, leading to the matrix decomposition

\[
[Q] = [W]^T [Q] [W] + [Q_{\text{s}}] + [Q_{\text{c}}] \tag{6}
\]

where \([Q_{\text{s}}], [W]\), and \([Q_{\text{c}}]\) are sparse matrices whose values are zeros except \([W]_{ij} = \delta(p_i, p_j)\), if \(p_i \subset \text{supp}(\lambda_j)\), \([Q_{\text{c}}]_{ij} = -[W]^T [Q] [W]_{ij}\) if \((\lambda_i, \lambda_j)\) is singular, \([Q_{\text{s}}]_{ij} = (\lambda_i, \lambda_j)\) is singular and \([Q_{\text{c}}]_{ij} = (1 - \delta_{ij}) q(p_i - p_j)\), where \(\delta_{ij} = 1\), if \(i = j = 0\), otherwise, is a dense matrix containing the kernel evaluation on the quadrature points. This approach has been referred to as point based \([15]\), and it is equivalent to replacing the traditional basis-to-basis interactions with point-to-point interactions. The correction matrix \([Q_{\text{c}}]\) subtracts contributions that are considered twice in \([Q_{\text{c}}]\) and \([Q]\). All the matrices above are sparse except \([Q]\), as it contains the kernel evaluations for quadrature points of nonsingular interactions.

3. A Multikernel Compression for the CFIE

We now consider \([Q] = [S]\) and \([Q] = [D]\), where we show that \([D]\) can be computed by reusing \([S]\), which we directly obtain, based on (5). Defining \(\tilde{G}(x, y) = 4\pi G(x, y)\), we find that

\[
\partial_n G(x, y) = G(x, y) \left((\|\tilde{G}(x, y)\|^2 - k^2)\|\tilde{G}(x, y)\|(x \cdot n' - y \cdot n')\right) \tag{7}
\]

By leveraging the decomposition (6), we obtain

\[
[D] = [H_{1x}][\tilde{G}][H_{2x}] + [H_{1y}][\tilde{G}][H_{2y}] + [H_{1x}][\tilde{G}][H_{2x}] - [\tilde{G}][M_1] - [\tilde{G}][M_2] - [\tilde{G}][M_2] \tag{8}
\]

where \([\tilde{G}]_{xy} = \langle S_{xy}, (16\pi^2 \|S_{xy}\|^2 - 4k^2) \|S_{xy}\|\rangle, [M_1] = \text{diag}(y_{\rho n'} p_i), [H_{1y}] = \text{diag}(\rho_n p_i), \) and \([H_{2y}] = \text{diag}(x_{\rho p_i})\).
with \( p \in \{1, 2, 3\} \), \( n_{i,p} \) are the \( p \)th components of the normal of the \( i \)th triangle, \( x_{i,p} \) and \( y_{i,p} \) are the \( p \)th components of the quadrature points \( x_i \) and \( y_i \) of the \( i \)th triangle, respectively.

Assume that \([B_i]\) and \([B_{G_i}]\) are rank-deficient submatrices of \([S]\) and \([G_d]\) that are based on the same subset of rows and columns. Further assume the LRF of 
\[ [B_i] \approx [C_i][U_i][R_i] \] 
with \([C_i] = [B_i](i,:)\), \([R_i] = [B_i](i,:)\), 
\[ [U_i] = [[B_i](i,J)]^\dagger \], where \( I \) and \( J \) are index sets containing the row and column indices used in the factorization. The \( \dagger \) denotes the Moore-Penrose pseudoinverse. With the same definitions, we denote the decomposition 
\[ [B_{G_i}] \approx [C_{G_i}][U_{G_i}][R_{G_i}] \] 
where we use the same index sets \( I \) and \( J \) for \([B_i]\). The matrix decomposition of \([B_{G_i}]\) can be related to \([B_i]\) by 
\[ [C_{G_i}][U_{G_i}][R_{G_i}] = [C_i][U_i][R_i] \]  
(9)

where, as in [16], \( \dagger \) is defined as 
\[ [A]^\dagger x = 4\pi \sum_{j=1}^{N} (4\pi |[A]_{jj}|^2 - kj|[A]_{jj}|)[A]_{jj}y_j \]  
(10)

Also, (9) shows that once a decomposition of \([B_i]\) has been obtained, the only additional information needed to approximate the submatrix \([B_{G_i}]\) is the matrix \([U_{G_i}]\), which can be computed in constant time. To obtain a memory-efficient compression, we note that the computation of \([C_i]\) and \([R_i]\) is extremely fast. Unlike in a traditional ACA scheme, where the actual system matrix is sampled, the direct evaluation of the Green’s function at the quadrature points is much faster in terms of CPU time. For this reason, we will compute the matrix vector products with \([C_i]\) and \([R_i]\) on the fly. Due to (9), the same approach carries over to \([G_{G_i}]\).

Thus, in our approach, we store only the intersections \([B_i](i,J)\) and \([B_{G_i}](i,J)\) for each admissible cluster, which significantly reduces the stored memory. During each MVP \([C][U][R][x]\), rather than computing the pseudoinverse, which is subject to instabilities and depends on a truncation threshold, we use an lower-upper solver. Because the rank \( r \) is not known beforehand, we use the heuristic algorithm described in [5] to sample rows and columns on each iteration, the indices \( I \) and \( J \) are chosen randomly, and the error is computed with respect to the projection into a random subspace of the previous iteration. If the error exceeds the demanded threshold, the sample size is doubled, and the algorithm continues with the next iteration. For each cluster, we have to compress the two submatrices \([B_i]\) and \([B_{G_i}]\).

For stability reasons, we first start factorizing \([B_i]\), and the obtained \( I \) and \( J \) are used as a starting point for factorizing \([B_{G_i}]\).

It is possible to optimize storage by only saving rows and columns, where the single layer is evaluated for \( I \) and \( J \) indices. One can use the subset of rows \( I_i \subset I \) and columns \( J_i \subset J \) to compute the single-layer MVP. Likewise, the double-layer MVP can be computed with the stored rows \( I \) and columns \( J \) by using (9) and (8). Our approach is based on two assumptions: 1) the \( \varepsilon \)-rank of \([B_{G_i}]\) is larger than that of \([B_i]\); and 2) the same rows and columns can be used for the factorization. In the next paragraph, we will provide some considerations that support these assumptions.

Assume \( g(x_i, y_i) \) to be the kernel function associated to the matrix \([B_g]\). We have 
\[ [B_g]_{ij} = g(x_i, y_j) = g_p(x_i, y_j) + r^{g_p}_{xy}(x_i) \]  
(11)

where \( g_p \) is the polynomial (Lagrange) interpolation of \( g \) of order \( p \) and \( r^{g_p}_{xy} \) is the associated remainder. Also, assume a pseudoskeleton approximation of rank \( K_p = \sum_{j=0}^{p-1} \frac{(l+2)!}{l!} \) 
\[ [B_g] = [C_g][U_g][R_g] + [E_g] \]  
(12)

where \([E_g]_{ij} = r^{g_p}_{K_p}(x_i, y_j)\) is the associated remainder. This definition can also be written as 
\[ [B_g]_{ij} = [C_g](i,:)[U_g][R_g](,:j) + r^{g_p}_{K_p}(x_i, y_j) \]  
(13)

It can be shown that \( r^{g_p}_{K_p} \) is related to \( r^{g_p}_{xy} \) [17, eq. (6)]
\[ |r^{g_p}_{K_p}(x_i, y_j)| \leq (1 + 2^{K_p}) \sup_{x} |r^{g_p}_{xy}(x)| \]  
(14)

that later will allow us to shift the results on the bound of the polynomial remainder to the one arising from the pseudoskeleton approximation. As a consequence, we exploit this connection to justify our method by showing that given a function \( g_s(x, y) = \|x - y\|^{-s} \), for \( s = 1 \), the relative error of an associated polynomial interpolation will decrease faster to zero than for \( s > 1 \) and \( p \to \infty \). To show this, we first recall that the kernels \( g_s : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R} \) are asymptotically smooth, so \( C_{as1,g_s} < C_{as2,g_s} > 0 \) and \( s \in \mathbb{N}^\ast \) exist and we have [18]
\[ |\partial^n g_s(x, y)| \leq C_{as1,g_s}(C_{as2,g_s})^{-n-s} n! \]  
(15)

where \( g_s(x, y) = \|x - y\|^{-s} \) and \( \partial^n \) is the \( n \)th derivative with respect to \( x_i \). The constants \( C_{as1} \) and \( C_{as2} \) are indicated with a subscript \( g_s \) to indicate the kernel dependency. For \( s = 1 \), we have \( C_{as1,g_s} = C_{as2,g_s} = 1 \) [8, section 2.1]. Furthermore, to show the faster convergence, we need \( C_{as2,g_s} < 1 \) for \( s > 1 \), which is guaranteed by the following lemma.

### 3.1 Lemma 1

Let \( g_s(x, y) = \|x - y\|^{-s} \) with \( x, y \in \mathbb{R}^3, x \neq y \) and \( C_{as1,g_s} \in \mathbb{R}^+ \). For \( \mathbb{N} \ni s > 1 \), we have \( C_{as2,g_s} < 1 \).
3.1.1 Proof: Let $i \in 1, 2, 3$. For proving the lemma, it is sufficient to consider the special case $x_j = y_j$ for $j \neq i$. Given the kernel $g_s(x, y) = \|x - y\|^{-s}$, with $s \geq 1$, we have

$$\|g_s^n(x, y)\| = \|x - y\|^{n-s}(n - s + 1)!$$  \hspace{1cm} (16)

that can be substituted in the asymptotic condition (15), leading to

$$\|x - y\|^{n-s}(n - s + 1)! \leq C_{as1, s}$$

$$C_{as2, s} \|x - y\|^{n-s}n!$$  \hspace{1cm} (17)

that simplifies to

$$\prod_{i=n+1}^{n+s-1} i \leq C_{as1, s} C_{as2, s}$$  \hspace{1cm} (18)

that leads to a contradiction if $C_{as2, s} \geq 1$. □

Let $x \in B_r \subset \mathbb{R}^3$ and $y \in B_r \subset \mathbb{R}^3$, where $B_r$ and $B_s$ are boxes as an octree would generate them. We introduce the auxiliary constants defined in [18] $\gamma_s := 1/(C_{as2, s} \text{dist}(B_r, B_s))$, $C_s := C_{as1, s}/(C_{as2, s} \text{dist}(B_r, B_s))^s$, and $C_{diam} := \min\{\text{diam}(B_r), \text{diam}(B_s)\}$. Let $g_s$ be an approximation of $g_s$ obtained via an interpolation scheme. Assuming that $\text{dist}(B_r, B_s) > 0$, the interpolation (absolute) error is bounded by [18, eq. (12)]

$$\|g_s - g_s\|_{\infty, B_r \times B_s} \leq 24\Lambda \gamma_s C_g$$

$$(1 + \gamma_s C_{diam}) (p + 1) (1 + 2/(\gamma_s C_{diam}))^{-p+1}$$  \hspace{1cm} (19)

where $\Lambda$ is the Lebesgue constant, which depends on the specific interpolation scheme used.

To simplify the notation, we introduce the constant $\beta = C_{diam}/(2 \text{dist}(B_r, B_s))$. We note that $(1 + \gamma_s C_{diam}) = (C_{as2, s} + 2\beta)/C_{as2, s}$, and $C_{diam}$ is equal to $\min\{\text{diam}(B_r), \text{diam}(B_s)\}$. Let $g_s$ be an approximation of $g_s$ obtained via an interpolation scheme. Assuming that $\text{dist}(B_r, B_s) > 0$, the interpolation (absolute) error is bounded by [18, eq. (12)]

$$\|g_s - g_s\|_{\infty, B_r \times B_s} \leq 24\Lambda \gamma_s C_g$$

$$\left(\frac{C_{as2, s} + 2\beta}{C_{as2, s}}\right) (p + 1) \left(\frac{\beta}{\beta + C_{as2, s}}\right)^{-p+1}$$  \hspace{1cm} (20)

We divide (20) by $\|g_s\|_{\infty, B_r \times B_s} = \text{dist}(B_r, B_s)^{-s}$, yielding

$$\tilde{E}_s \|_{p} := \left\|\frac{g_s}{\|g_s\|_{\infty, B_r \times B_s}}\|_{p} \leq 24\Lambda \gamma_s C_g$$

$$\left(\frac{C_{as2, s} + 2\beta}{C_{as2, s}}\right) (p + 1) \left(\frac{\beta}{\beta + C_{as2, s}}\right)^{-p+1}$$  \hspace{1cm} (21)

4. Numerical Results

Consider as a canonical example the scattering from a sphere with radius 1 m, sound soft boundary condition, and a plane wave excitation with $\hat{k} = \hat{z}$, and frequency $20 \text{ Hz}$ corresponding to $k \approx 0.36 \text{ m}^{-1}$. We study the asymptotic memory consumption and the time. As denoted in Section 2, we apply our algorithm to the CFIE (4). We compare the performance of our scheme with a standard ACA. Table 1 compares the timing between the two methods, while Figure 1 shows

<table>
<thead>
<tr>
<th>Time (min)</th>
<th>N</th>
<th>2562</th>
<th>10,242</th>
<th>40,962</th>
<th>163,842</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACA</td>
<td>Assembly</td>
<td>0.0675</td>
<td>0.3538</td>
<td>1.8326</td>
<td>9.1448</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>0.0690</td>
<td>0.3600</td>
<td>1.8605</td>
<td>9.2567</td>
</tr>
<tr>
<td>New</td>
<td>Assembly</td>
<td>0.0408</td>
<td>0.2171</td>
<td>1.0742</td>
<td>5.2420</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>0.1130</td>
<td>0.5204</td>
<td>2.4237</td>
<td>11.0048</td>
</tr>
</tbody>
</table>

Table 1. The timing performance of a standard ACA and our new formulation (multikernel)
memory consumption and the relative error. First, we note that the relative error of the solution displayed in the plot is with respect to the result provided by the analytical solution for the far field. Our formulation yields a similar accuracy as the standard ACA, while yielding memory savings. We note that for larger problems, the relative memory savings increase without jeopardizing the total run time. Our experiments have been performed on a cluster with a CPU Intel Xeon Gold 6238R and 1.5 TB of RAM. We used GMRES [19] as the numerical solver of the linear system, with a target accuracy of $10^{-5}$. The target accuracy for the standard and new compression schemes is set to $10^{-5}$. However, given the differences between compression algorithms, an identical target accuracy will lead to a lower level of compression for the reference ACA-based scheme. In practice, this means that a lower target accuracy can be set for the new scheme to obtain the same field accuracy, which leads to time and memory savings. The discussion on the choice of optimal target accuracy is omitted here because of space constraints.

5. References