Approximation of Fractional Advection Dispersion Equations by Hierarchical Matrix Methods

---

A Master's Project
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master's
Mathematical Science

---

by
Ricky Eugene Farr
August 2009

Accepted by:
Dr. Vince Ervin, Committee Chair
Dr. Chris Cox
Dr. Shuyu Sun

Dedicated to my father, who taught me how to think about the world, and to my mother, who taught me how to live in it.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Linear systems of equations with fully occupied coefficient matrices arise in a number of physical application, e.g. solution of integral equations, solution of fractional advection dispersion equations. There is many drawbacks to this as the operations of multiplication, addition, matrix-vector multiplication, are of $O(n^2)$ complexity. Also, finding the inverse of a dense square matrix is $O(n^3)$.

Hierarchical matrices arise naturally from integral equations. The idea of Hierarchical matrices is to replace a dense matrix with one that is data-sparse, meaning that a coefficient matrix from a linear system of equations can be represented by a small amount of data. This will reduce our memory storage and we hope that it will retain the approximation properties that one can achieve normally.

This paper is organized into three chapters. Chapter 1 begins with a motivational example. In this chapter, we use the hierarchical matrix structure to approximate a simple one-dimensional model problem. In Chapter 2, domain partitioning is discussed and an algorithm to partition a domain efficiently is discussed. In Chapter 3, the fractional advection dispersion equation (FADE) is introduced and using the ideas from Chapter 1 and 2, we approximate FADE and compare the results to the normal finite element approximation.

Also in Chapter 3, we experiment with the rank of submatrices in the resulting H-matrices.

1.1 Motivation Problem

Example 1. *Let* $F : [0,1] \to \mathbb{R}$ *be given. We seek a solution* $U : [0,1] \to \mathbb{R}$ *satisfying*

$$\int_0^1 \ln|x-y|U(y)dy = F(x), \qquad x \in [0,1] \qquad \textit{(1.1)}$$

Also, for convenience let $g(x,y) = ln|x-y|$ *be called the kernel function.*

First we observe that $g(x,y)$ is undefined for $x = y$. A graph of g is given below.

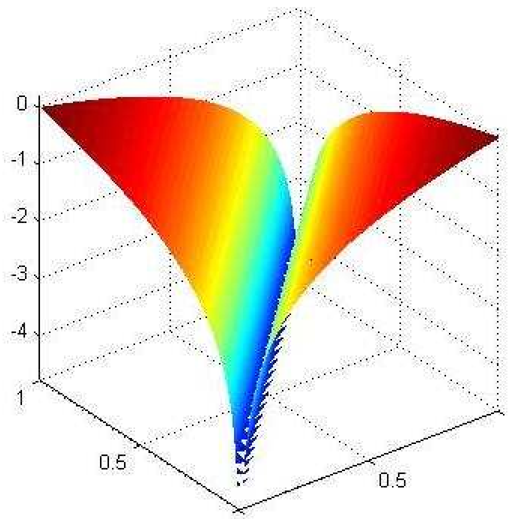

Figure 1.1: Graph of Kernel Function

We employ Galerkin's discretization for Example 1, that is we solve equation (1.1) projected onto the n-dimensional space $V_n := span\{\phi_1, \phi_2, ..., \phi_n\}$,

$$\int_0^1 \int_0^1 \phi_i(x) ln|x-y|U(y)dydx = \int_0^1 \phi_i(x)F(x)dx, \qquad 1 \le i \le n$$

and seek the discrete solution U_n in the space V_n, where $U_n = \sum_{j=1}^{n} c_j\phi_j$ for unknown coefficients c_j. Let $\mathbf{c}$ be the corresponding coefficient vector, then $\mathbf{c}$ is the solution to the linear system $G\mathbf{c} = f$, where

$$G_{ij} = \int_0^1 \int_0^1 \phi_i(x) ln|x-y| \phi_j(y) dy dx \qquad \text{and} \qquad f_i = \int_0^1 \phi_i(x) F(x) dx$$

In this introductory example, we use piece-wise constant basis functions

$$\phi_i(x) = \begin{cases} 1 & \frac{i}{n} \leq x \leq \frac{i+1}{n} \\ 0 & otherwise \end{cases}$$

on a uniform grid of $[0,1]$. Note that the matrix G is dense in the sense that all entries are nonzero. We will approximate G by a matrix $\hat{G}$ that can be stored in a data-sparse format. To do this we first replace the kernel by a degenerate kernel

$$\hat{g}(x,y) = \sum_{v=0}^{k-1} g_v(x) h_v(y)$$

so that the integration with respect to the x-variable is separated from the one with respect to the y-variable. Observe though, that the kernel function cannot be approximated on the entire domain $[0,1] \times [0,1]$. Instead we construct local approximations for subdomains of $[0,1] \times [0,1]$ where g is sufficiently smooth.

1.2 Taylor Expansion of the Kernel

Let $\tau = [a,b], \sigma = [c,d]$ with $\tau \times \sigma \subset [0,1] \times [0,1]$ be a subdomain with $b < c$ (and thus $\tau \cap \sigma = \emptyset$). It is easy to see that the kernel function is nonsingular in $\tau \times \sigma$. Let us now examine the partial derivatives of the kernel function.

Lemma 1. *The derivatives of $g(x,y) = ln|x-y|$ with $x \neq y$ and $v \in \mathbb{N}$ are given by*

$$\partial_x^v g(x,y) = (-1)^v (v-1)!(x-y)^{-v}$$
$$\partial_y^v g(x,y) = -(v-1)!(x-y)^{-v}$$

Proof. This is an elementary calculus proof, and is therefore omitted here. □

The Taylor series of $x \to g(x,y)$ about $\bar{x} = \frac{a+b}{2}$ converges in the whole interval τ, and the remainder of the truncated Taylor series can be estimated as follows.

Lemma 2. *Let $k \in \mathbb{N}$. The function $\hat{g}(x,y)$ defined by*

$$\hat{g}(x,y) = \textstyle\sum_{v=0}^{k-1} \frac{1}{v!} \partial_x^v g(\bar{x},y)(x-\bar{x})^v$$

approximates $g(x,y)$ with error given by

$$|g(x,y) - \hat{g}(x,y)| \leq \tfrac{|\bar{x}-a|}{|c-b|} \left(1 + \tfrac{|c-b|}{|\bar{x}-a|}\right)^{-k}$$

Proof. Let $k \in \mathbb{N}$, $x \in [a,b], y \in [c,d]$ with $a < b$ and $c < d$. Within the radius of convergence the Taylor series expansion of the kernel function about $\bar{x}$ as defined above is,

$$g(x,y) = \textstyle\sum_{v=0}^{\infty} \frac{1}{v!} \partial_x^v g(\bar{x},y)(x-\bar{x})^v.$$

Observe that,

$$\begin{aligned}
|g(x,y) - \hat{g}(x,y)| &= \left| \sum_{v=0}^{\infty} \frac{1}{v!} \partial_x^v g(\bar{x},y)(x-\bar{x})^v - \sum_{v=0}^{k-1} \frac{1}{v!} \partial_x^v g(\bar{x},y)(x-\bar{x})^v \right| \\
&= \left| \sum_{v=k}^{\infty} \frac{1}{v!} \partial_x^v g(\bar{x},y)(x-\bar{x})^v \right| \\
&= \left| \sum_{v=k}^{\infty} (-1)^v \frac{(v-1)!}{v!} \left(\frac{x-\bar{x}}{\bar{x}-y} \right)^v \right| \\
&\leq \sum_{v=k}^{\infty} \left| \frac{x-\bar{x}}{\bar{x}-y} \right|^v \\
&\leq \sum_{v=k}^{\infty} \left(\frac{|\bar{x}-a|}{|\bar{x}-a| + |c-b|} \right)^v \\
&= \left(1 + \frac{|\bar{x}-a|}{|c-b|}\right) \left(1 + \frac{|c-b|}{|\bar{x}-a|}\right)^{-k}
\end{aligned}$$

Since $1 + \frac{|c-b|}{|\bar{x}-a|} \geq 1$, we know that our approximation covers the entire interval $[a,b]$. $\square$

Note that if $c \to b$ then the error of approximation tends to infinity. Let us explore the condition $b < c$ (disjointness of the intervals).

Definition 1. Let $\tau = [a, b], \sigma = [c, d]$ with $\tau \times \sigma \subset [0, 1] \times [0, 1]$. We say $\tau \times \sigma$ is an admissible subdomain of $[0, 1] \times [0, 1]$ if $diam(\tau) \leq dist(\tau, \sigma)$.

The importance of this definition is given in the next theorem.

Theorem 3. *Let $\tau = [a, b], \sigma = [c, d]$ with $\tau \times \sigma \subset [0, 1] \times [0, 1]$. If $\tau \times \sigma$ is an admissible subdomain of $[0, 1] \times [0, 1]$, then $|g(x, y) - \hat{g}(x, y)| \leq \frac{3^{-k+1}}{2}$.*

Proof. Let $\tau = [a, b], \sigma = [c, d]$ with $\tau \times \sigma \subset [0, 1] \times [0, 1]$, assume that $\tau \times \sigma$ is an admissible subdomain of $[0, 1] \times [0, 1]$. That is, assume $diam(\tau) \leq dist(\tau, \sigma)$. That is $b - a \leq c - b$. By lemma 2, we know $|g(x, y) - \hat{g}(x, y)| \leq \left(1 + \frac{|\bar{x}-a|}{|c-b|}\right)\left(1 + \frac{|c-b|}{|\bar{x}-a|}\right)^{-k}$. Observe that

$$\begin{aligned}
|g(x, y) - \hat{g}(x, y)| &\leq \left(1 + \frac{|\bar{x} - a|}{|c - b|}\right)\left(1 + \frac{|c - b|}{|\bar{x} - a|}\right)^{-k} \\
&= \left(1 + \frac{\left|\frac{a+b}{2} - a\right|}{|c - b|}\right)\left(1 + \frac{|c - b|}{\left|\frac{a+b}{2} - a\right|}\right)^{-k} \\
&= \left(1 + \frac{|b - a|}{2|c - b|}\right)\left(1 + \frac{2|c - b|}{|b - a|}\right)^{-k} \\
&\leq \left(\frac{3}{2}\right)(3)^{-k} \\
&= \frac{3^{-k+1}}{2}
\end{aligned}$$

Hence, $|g(x, y) - \hat{g}(x, y)| \leq \frac{3^{-k+1}}{2}$. □

So, on a uniform grid of $[0, 1]$ we know that the approximation error decays exponentially with respect to the order of k, as long as the admissibility condition is satisfied.

1.3 Low Rank Approximation of Admissible Blocks

Let $I = \{0, 1, ..., n-1\}$ be an index set that contains the indices of the basis functions ϕ_i used in the Galerkin discretisation. Let t and s be two subsets of I and define their corresponding domains as

$$\tau = \bigcup_{i\in t} \text{supp } \phi_i \qquad \text{and} \qquad \sigma = \bigcup_{i\in s} \text{supp } \phi_i.$$

If $\tau \times \sigma$ is an admissible subdomain of $[0,1]\times[0,1]$ then we can approximate the kernel g in $\tau \times \sigma$ by a truncated Taylor series of g and replace the matrix entries

$$G_{ij} = \int_0^1 \int_0^1 \phi_i(x) g(x,y) \phi_j(y) dy dx$$

with the use of the degenerate kernel $\hat{g}(x,y) = \sum_{v=0}^{k-1} g_v(x) h_v(y)$, so for $(i,j) \in t \times s$ we have

$$\begin{aligned} G_{ij} &= \int_0^1 \int_0^1 \phi_i(x) \hat{g}(x,y) \phi_j(y) dy dx \\ &= \int_0^1 \int_0^1 \phi_i(x) \sum_{v=0}^{k-1} g_v(x) h_v(y) \phi_j(y) dy dx \\ &= \sum_{v=0}^{k-1} \left(\int_0^1 \phi_i(x) g_v(x) dx \right) \left(\int_0^1 \phi_j(y) h_v(y) dy \right) \end{aligned}$$

Hence, the benefit of doing this is two fold, since the double integral is separated into two single integrals, and the submatrix $G|_{t\times s}$ can be represented in factorized form as

$$G|_{t\times s} = AB^T, A \in \mathbb{R}^{t\times\{0,1,\ldots,k-1\}}, B \in \mathbb{R}^{s\times\{0,1,\ldots,k-1\}}$$

where A and B are defined as

$$A_{iv} = \int_0^1 \phi_i(x) g_v(x) dx \qquad \text{and} \qquad B_{jv} = \int_0^1 \phi_j(y) h_v(y) dy$$

We should point out, that the rank of AB^T is at most k and also independent of the cardinality of t and s. This motivates the next two definitions. The approximation error in approximating the submatrix is given in the next lemma.

Definition 2. Let A be an $m\times n$ matrix, if A can be written as $A = UV^T$, for some matrices U and V, we call this representation of A the outer-product form.

Definition 3. Let $I = \{0, 1, ..., n-1\}$ be an index set that contains the indices of the basis functions ϕ_i used in the Galerkin discretization. A block $t \times s \subset I \times I$ is said to be admissible if $\tau \times \sigma$ is an admissible subdomain of $[0,1] \times [0,1]$ where $\tau = \bigcup_{i \in t} supp\phi_i$ and $\sigma = \bigcup_{i \in s} supp\phi_i$. If $\tau \times \sigma$ is not an admissible subdomain of $[0,1] \times [0,1]$ then we say $t \times s$ is inadmissible.

Lemma 4. *The elementwise error for the matrix entries G_{ij} approximated by the truncated Taylor Series of g in the admissible block $t \times s$ (and g in the other blocks) is given by*

$$|G_{ij} - \hat{G}_{ij}| \leq \tfrac{3^{-k+1}}{2n^2}$$

Proof. Let G and $\hat{G}$ be matrices defined as $G_{ij} = \int_0^1 \int_0^1 \phi_i(x) g(x,y) \phi_j(y) dy dx$ and $\hat{G}_{ij} = \int_0^1 \int_0^1 \phi_i(x) \hat{g}(x,y) \phi_j(y) dy dx$ (in the admissible block $t \times s$ and g in the other blocks). Observe that we have

$$\begin{aligned}
|G_{ij} - \hat{G}_{ij}| &= \left| \int_0^1 \int_0^1 \phi_i(x)(g(x,y) - \hat{g}(x,y))\phi_j(y) dy dx \right| \\
&\leq \int_0^1 \int_0^1 |\phi_i(x)| \left(\frac{3^{-k+1}}{2} \right) |\phi_j(y)| dy dx \\
&= \frac{3^{-k+1}}{2} \int_0^1 \phi_i(x) dx \int_0^1 \phi_j(y) dy \\
&= \frac{3^{-k+1}}{2n^2}
\end{aligned}$$

$\square$

Now let us assume that we have partitioned the index set $I \times I$ for the matrix G into admissible blocks, where the low rank approximation is applicable, and inadmissible blocks, where we use the matrix entries from G. (We will present a method for doing this in the next chapter.) With this assumption, we can put a bound on the global approximation error in the Frobenius norm, where the Frobenius norm is defined for a matrix M of size $m \times n$, to be $||M||_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} m_{ij}^2$.

Lemma 5. *The approximation error $||G - \hat{G}||_F$ in the Frobenious norm for the matrix $\hat{G}$*

$(n \times n)$ built by the degenerate kernel $\hat{g}$ in the admissible blocks $t \times s \in P$ (P is a partition here) and by g in the inadmissible blocks is bounded by

$$||G - \hat{G}||_F \leq \tfrac{1}{2(3^{k-1})n}$$

The question that remains to be answered is how we want to partition the index set $I \times I$ into admissible and inadmissible blocks. A trivial partition would be $P = \{(i, j)|i \in I, j \in I\}$ where only 1×1 blocks of rank 1 appear. In this case, $\hat{G}$ would be identical to G, but this does not exploit the option to approximate the matrix in large subblocks of matrices of low rank.

Chapter 2

Building Partitions of a Domain

In this chapter, for I an index set of the basis elements used in the Galerkin discretization scheme, we discuss how to partition $I \times I$ into admissible and inadmissible blocks. To do this, we first define a cluster tree. For the admissibility condition we use Definition 1.

2.1 Cluster Tree T_I

Let I be an index set after Galerkin discretization used in the approximation of problem (1.1). The candidates $t, s \subset I$ for the construction of the partition $I \times I$ will be stored in a so-called cluster tree, T_I. The root of the tree, T_I, is the index set $I_1^{(0)} = \{0, 1, ..., n-1\}$. For ease of presentation we assume that the number of basis functions, n, is a power of two, i.e. $n = 2^p$.

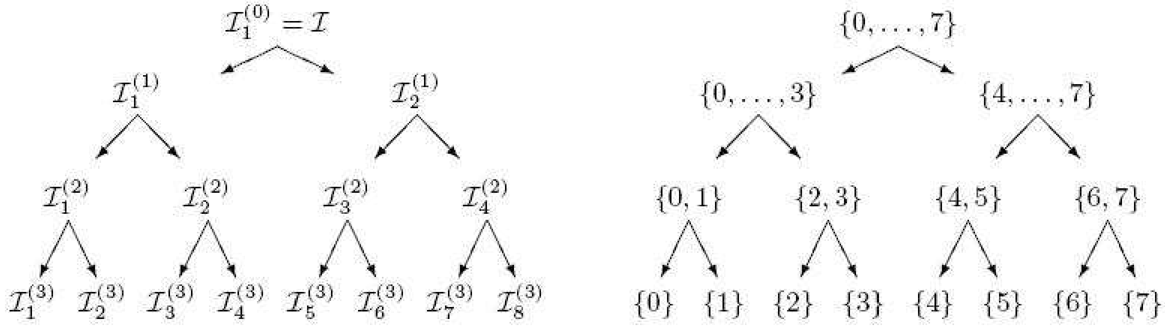

Figure 2.1: Illustration of T_I

As one can see from Figure 2.1, the two successors of $I_1^{(0)}$ are $I_1^{(0)} = \{0, 1, ..., \frac{n}{2} - 1\}$ and $I_2^{(1)} = \{\frac{n}{2}, ..., n-2, n-1\}$. Also, the two successors of $I_1^{(1)}$ are $I_1^{(2)} = \{0, 1, ..., \frac{n}{4} - 1\}$ and $I_2^{(2)} = \{\frac{n}{4}, ..., \frac{n}{2} - 2, \frac{n}{2} - 1\}$. This scheme is illustrated in Figure 2.1. Let n_{min} denote the desired minimal size of a node, where by node we mean an entry in the tree T_I. Each subsequent node, t, with more that n_{min} indices has exactly two successors: The first node contains the first half of its indices, the second node contains the second half. Nodes with n_{min} or less indices are leaves. So, parameter n_{min} controls the depth of the tree. For $n_{min} = 1$ we get the maximal depth. However, for practical purposes we might want to set $n_{min} = 2k$ or $n_{min} = 16$, to preserve computation time in creating the cluster tree, T_I.

Note that for $n_{min} = 1$ the tree T_I is a binary tree of depth p, see Figure 2.1. It contains subsets of the index set I of different size. The first level consists of the root $I = \{0, 1, ..., n-1\}$ with n indices, the second level contains two nodes with $n/2$ indices and so fourth. So, we know that the tree is balanced with respect to cardinality, i.e. nodes on the same level have the same number of elements. Also, the number of nodes in the tree T_I is $2n - 1$, since the tree is binary.

2.2 Block Cluster Tree $T_{I\times I}$

The number of possible blocks $t \times s$ with nodes t, s from the the tree T_I is $(2n-1)^2 = O(n^2)$. So, for n large, it is computationally infeasible to test all possible combinations $I \times I$. In this section our aim is to reduce the quadratic cost for the assembly of the matrix.

One approach to reduce this cost is to test nodes starting with the root I of the tree T_I and desending in the tree. The tested blocks are stored in a so-called block cluster tree $T_{I\times I}$ whose leaves form a partition of the index set $I \times I$. The algorithm is given as follows and called with parameters $BuildBlockClusterTree(I, I)$. Let $S(cluster\ t, cluster\ s)$ be the set of sons of t.

Algorithm 1. BuildBlockClusterTree(cluster t,cluster s)

if(t,s) is admissible or $|t| = |s| = 1$

S($t \times s$)=$\emptyset$

else

S($t \times s$)=$\{t^{'} \times s^{'} | t^{'} \in S(t), s^{'} \in S(s)\}$

for $t' \in S(t)$ and $s' \in S(s)$

BuildBlockClusterTree(t', s')

end

end

Note that the tree $T_{I\times I}$ is a quadtree, but there are leaves on different levels of the tree which is not the case for the binary tree T_I.

Example 2. We consider the example tree from the above Figure 2.1. The root of the tree is $\{0, 1, ..., 7\} \times \{0, 1, ..., 7\}$.

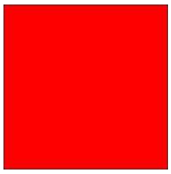

Figure 2.2: Root of Cluster Tree

which is not admissible, in terms of Definition 1, because the corresponding domain to the index set $\{0, 1, ..., 7\}$ is the interval $[0, 1]$ and $diam([0, 1]) = 1 \nleq 0 = dist([0, 1], [0, 1])$.

The four successors of the root in the tree $T_{I \times I}$ are $\{0, 1, 2, 3\} \times \{0, 1, 2, 3\}$, $\{0, 1, 2, 3\} \times \{4, 5, 6, 7\}$, $\{4, 5, 6, 7\} \times \{0, 1, 2, 3\}$, and $\{4, 5, 6, 7\} \times \{4, 5, 6, 7\}$.

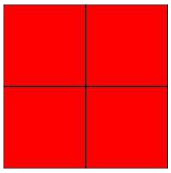

Figure 2.3: Structure when p=1

Again, none of these are admissible, in terms of Definition 1, and they are further subdivided into $\{0, 1\} \times \{0, 1\}$, $\{0, 1\} \times \{2, 3\}$, $\{0, 1\} \times \{4, 5\}$, $\{0, 1\} \times \{6, 7\}$, $\{2, 3\} \times \{0, 1\}$, $\{2, 3\} \times \{2, 3\}$, $\{2, 3\} \times \{4, 5\}$, $\{2, 3\} \times \{6, 7\}$, $\{4, 5\} \times \{0, 1\}$, $\{4, 5\} \times \{2, 3\}$, $\{4, 5\} \times \{4, 5\}$, $\{4, 5\} \times \{6, 7\}$, $\{6, 7\} \times \{0, 1\}$, $\{6, 7\} \times \{2, 3\}$, $\{6, 7\} \times \{4, 5\}$, $\{6, 7\} \times \{6, 7\}$.

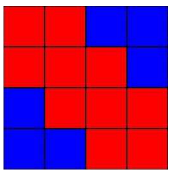

Figure 2.4: Structure when p=2

Now some of the nodes are admissible, in terms of Definition 1, e.g. the node $\{0, 1\} \times \{4, 5\}$ because the corresponding domain is $[0, \frac{1}{4}] \times [\frac{1}{2}, \frac{3}{4}]$, and $diam([0, \frac{1}{4}]) = \frac{1}{4} = dist([0, \frac{1}{4}], [\frac{1}{2}, \frac{3}{4}])$.

Remark 1. The nodes on the diagonal are not admissible as the distance of the corresponding domain to itself is zero, and also some of the nodes off the diagonal, e.g., $\{0, 1\} \times \{2, 3\}$, are not admissible. The successors of these nodes are the singletons $\{(i, j)\}$ for indices i, j. The final structure of the partition looks as follows

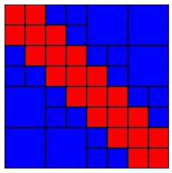

Figure 2.5: Final Structure

For $p = 4$ and $p = 5$ the structure of the partition is similar

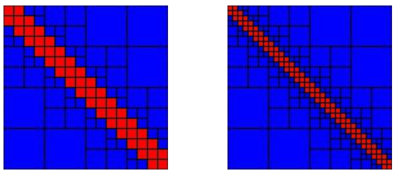

Figure 2.6: *Final Structure when p=4 and p=5*

2.3 Assembly and Storage

The product index set $I \times I$ decomposes into admissible and inadmissible nodes of the tree $T_{I\times I}$. The assembly and storage differs for the corresponding two classes of submatrices. Likewise, for the two corresponding classes of submatrices their arithmetic will differ too; this will be discussed later. First, we introduce some notation to aid in the discussions.

Definition 4. Let $m, n \in \mathbb{N}, k \in \mathbb{N}_0$. Define the set of $m \times n$ matrices of rank at most k by

$$R(k, m, n) = \{M \in \mathbb{R}^{m\times n} | rank(M) \leq k\}$$

A suitable representation for the submatrix $\hat{G}|_{t\times s}$ is the rk-matrix format defined next.

Definition 5. An $m \times n$ matrix M of rank at most k is said to be stored in rk-matrix representation if it is stored in factorized form $M = AB^T$ where the two matrices $A \in \mathbb{R}^{m\times k}$ and $B \in \mathbb{R}^{n\times k}$ are both stored as two-dimensional arrays.

The next definition is a way of storing the inadmissible blocks.

Definition 6. An $m \times n$ matrix $M = [m_{ij}]$ is said to be stored in full-matrix representation if the entries m_{ij} are stored as real numbers (double) in an array of length $m * n$ in the order

$$m_{1,1}, m_{1,2}, ..., m_{1,m}, m_{2,1}, ..., m_{2,m}, ..., m_{n,1}, ..., m_{n,m}$$

column-wise.

We use the definitions of full-matrix and rk-matrix formats to formally define a Hierarchical matrix representation of a matrix.

Definition 7. Let $T_{I\times I}$ be a block cluster tree for the index set I. We define the set of $H - matrices$ as

$$H(T_{I\times I}, k) = \{M \in \mathbb{R}^{I\times I} | rank(M|_{t\times s} \leq k \text{ for all admissible nodes } t \times s \ \in T_{I\times I}\}$$

Definition 8. Let $T_{I\times I}$ be a block cluster tree for the index set I. A matrix $M \in H(T_{I\times I}, k)$ is said to be stored in $H - matrix$ representation if the submatrices corresponding to inadmissible blocks are stored in full-matrix representation and those corresponding to admissible blocks are stored in rk-matrix representation.

Having formally defined the set of $H - matrices$ and the $H - matrix$ representation, we next discuss the assembly of inadmissible and admissible blocks.

2.3.1 Inadmissible Leaves

For the inadmissible blocks $t \times s \subset I \times I$ we compute the entries (i, j) of $G|_{t\times s}$ as usual. Namely,

$$\begin{aligned}
\hat{G}_{ij} &= \int_0^1 \int_0^1 \phi_i(x) ln|x-y| \phi_j(y) dy dx \\
&= \int_{\frac{i}{n}}^{\frac{i+1}{n}} \int_{\frac{j}{n}}^{\frac{j+1}{n}} ln|x-y| dy dx
\end{aligned}$$

2.3.2 Admissible Leaves

For the admissible blocks $t \times s \subset I \times I$ with corresponding domains $[a, b] \times [c, d]$ and $\bar{x} = \frac{a+b}{2}$ we compute the submatrix in factorized form

$$\begin{aligned}
\hat{G}|_{t\times s} &= AB^T \\
A_{iv} &= \int_{\frac{i}{n}}^{\frac{i+1}{n}} (x-\bar{x}) dx \\
B_{jv} &= \begin{cases} (-1)^{v+1} v^{-1} \int_{\frac{j}{n}}^{\frac{j+1}{n}} (\bar{x}-y)^{-v} dy, v > 0 \\ \int_{\frac{j}{n}}^{\frac{j+1}{n}} ln|\bar{x}-y| dy, v = 0 \end{cases}
\end{aligned}$$

2.4 Numerical Results

So far, we have only discussed how to assemble and store H-matrices. Before we get to results, we discuss how to solve the approximate system of linear equations. With the next definition, we define how to multiply an H-matrix by a vector. After defining this operation, we use GMRES (Generalized Minimum Residual Method) to solve the linear system of equations.

Definition 9. Let A be a matrix, n_A the number of admissible blocks in a partition, A_{IA} be a sparse matrix with only inadmissible blocks represented. Also, let A_i be an enumeration of matrices that are sparse, and non-zero in the admissible portion of the matrix. We can write, $A \approx A_{IA} + \sum_{k=1}^{n_A} A_k$. Note that $A_{IA} + \sum_{k=1}^{n_A} A_k$ is the H-matrix representation of A. We define H-matrix vector multiplication as, $Ax \approx A_{IA}x + \sum_{k=1}^{n_A} A_k x$

Adapting GMRES for H-matrix multiplication, we get the following results (no preconditioner) for Example 1. Also, note that we truncate the series in Lemma 2 so that the number of terms is no more than the size of the admissible block being approximated.

h	$\|\|u - u_h\|\|_{L^2(\Omega)}$	cvge. rate
1/4	8.508752×10^{-2}	
1/8	4.025139×10^{-2}	1.08
1/16	1.887749×10^{-2}	1.09
1/32	9.234988×10^{-3}	1.03
1/64	4.56487×10^{-3}	1.02

Table 2.1: H-Matrix Technique for Example 1

These results are comparable to the results obtained by using Galerkin's Method without an H-matrix technique.

h	$\|\|u - u_h\|\|_{L^2(\Omega)}$	cvge. rate
1/4	8.424849×10^{-2}	
1/8	4.008098×10^{-2}	1.07
1/16	1.885375×10^{-2}	1.09
1/32	9.231173×10^{-3}	1.03
1/64	4.564444×10^{-3}	1.02

Table 2.2: Galerkin FEM Estimates for Example 1

Chapter 3

Fractional Advection Diffusion Equations

The fractional advection dispersion equation (FADE) is useful in modeling anomalous diffusion, i.e. diffusion not accurately modeled by the usual advection dispersion equation. For example, FADE has been used to simulate solute transport in groundwater [2]. In order to discuss this equation, we define left and right fractional integrals.

Definition 10. Let u be a function defined on the interval (a, b) and let $\sigma > 0$. Then the left Riemann-Liouville fractional integral of order σ is defined to be

$$ {}_aD_x^{-\sigma}u(x) = \tfrac{1}{\Gamma(\sigma)} \int_a^x (x-s)^{\sigma-1}u(s)ds. $$

Definition 11. Let u be a function defined on the interval (a, b) and let $\sigma > 0$. Then the right Riemann-Liouville fractional integral of order σ is defined to be

$$ {}_xD_b^{-\sigma}u(x) = \tfrac{1}{\Gamma(\sigma)} \int_x^b (s-x)^{\sigma-1}u(s)ds. $$

For properties of the Riemann-Liouville fractional integral operators, see [3]. We investigate the approximation of the steady state FADE given by

$$- Da(p_0 D_x^{-\beta} + q_x D_1^{-\beta})Du + b(x)Du + c(x)u = f \tag{3.1}$$

where D represents a single spatial derivative, and ${}_0D_x^{-\beta}$, ${}_xD_1^{-\beta}$ represent left and right fractional integral operators, respectively, with $0 \leq \beta < 1$, and $0 \leq p, q \leq 1$, satisfying $p + q = 1$.

We now discuss the Galerkin Finite Element method approximation to FADE. Then we will discuss using H-matrix schemes.

3.1 Galerkin Finite Element Method for Concrete Examples

In this section we formulate the Galerkin Finite Element Method for (3.1). Numerical results will be given for two examples. This will be useful in the following sections, as we will compare the Galerkin Finite Element Method approximation with that obtained using an H-matrix scheme.

Example 3. Let $p = 1, q = 0, \beta = 1/2, a = 1$, and $b = c = 0$. Then, $u(x) = x^2$ is the exact solution to the boundary value problem

$$-D_0 D_x^{-1/2} Du = \frac{-2\sqrt{x}}{\Gamma(3/2)} \tag{3.2}$$

$$u(0) = 0, u(1) = 1$$

We approximate the solution of (3.2) by projecting the solution into the n-dimensional space $V_n = span\{\phi_0, \phi_1, ..., \phi_{n-1}\}$, where ϕ_i, $i = 0, ..., n-1$ forms a basis for V_n. Multiplying (3.2) by ϕ_i, integrating over $(0, 1)$, then integrating by parts we obtain

$$\frac{1}{\Gamma(1/2)}\int_0^1\int_0^x (x-s)^{-1/2}Du(s)ds\phi_i^{'}(x)dx = \int_0^1 f(x)\phi_i(x)dx, 1\le i<n$$

where $f(x) = \frac{-2\sqrt{x}}{\Gamma(3/2)}$. We seek the solution u_n in the space V_n, where $u_n = \sum_{j=0}^{n-1} c_j\phi_j$ for unknown coefficients c_j. Let $\mathbf{c}$ denote the corresponding coefficient vector. Then $\mathbf{c}$ is the solution of the linear system, $G\mathbf{c} = \mathbf{f}$, where

$$G_{ij} = \frac{1}{\Gamma(1/2)}\int_0^1\int_0^x (x-s)^{-1/2}\phi_j^{'}(s)ds\phi_i^{'}(x)dx \quad \text{and} \quad f_i = \int_0^1 \phi_i(x)f(x)dx$$

Let $x_j, j = 0, 1, ..., n-1$, denote a uniform partition of $[0, 1]$, with $h = 1/n$. In this example, we use piece-wise linear, continuous basis functions (hat functions) defined by:

$$\phi_i(x) = \begin{cases} \frac{1}{h}(x - x_{i-1}), & x_{i-1}\le x\le x_i \\ \frac{-1}{h}(x - x_{i+1}), & x_i\le x\le x_{i+1} \\ 0, & otherwise \end{cases}$$

We simplify the expression for G_{ij} given above. Observe,

$$\begin{aligned}
G_{ij} &= \frac{1}{\Gamma(1/2)}\int_0^1\int_0^x (x-s)^{-1/2}\phi_j^{'}(s)ds\phi_i^{'}(x)dx \\
&= \frac{1}{\Gamma(1/2)}\int_{x_{i-1}}^{x_{i+1}}\int_0^x (x-s)^{-1/2}\phi_j^{'}(s)ds\phi_i^{'}(x)dx \\
&= \frac{1}{\Gamma(1/2)}\left(\int_{x_{i-1}}^{x_i}\int_0^x (x-s)^{-1/2}\phi_j^{'}(s)ds\left(\frac{1}{h}\right)dx + \int_{x_i}^{x_{i+1}}\int_0^x (x-s)^{-1/2}\phi_j^{'}(s)ds\left(\frac{-1}{h}\right)dx\right) \\
&= \frac{1}{h\Gamma(1/2)}\left(\int_{x_{i-1}}^{x_i}\int_0^x (x-s)^{-1/2}\phi_j^{'}(s)dsdx - \int_{x_i}^{x_{i+1}}\int_0^x (x-s)^{-1/2}\phi_j^{'}(s)dsdx\right) \\
&= \frac{1}{h\Gamma(1/2)}\left(\int_{x_{i-1}}^{x_i}\sum_{r=0}^{i-2}\int_{x_r}^{x_{r+1}} (x-s)^{-1/2}\phi_j^{'}(s)dsdx + \int_{x_{i-1}}^{x_i}\int_{x_{i-1}}^{x} (x-s)^{-1/2}\phi_j^{'}(s)dsdx\right. \\
&\quad \left. - \int_{x_{i-1}}^{x_i}\sum_{r=0}^{i-1}\int_{x_r}^{x_{r+1}} (x-s)^{-1/2}\phi_j^{'}(s)dsdx - \int_{x_i}^{x_{i+1}}\int_{x_i}^{x} (x-s)^{-1/2}\phi_j^{'}(s)dsdx\right)
\end{aligned}$$

Since $\phi_j'(s)$ is piecewise constant, integrating we get

$$\begin{aligned} G_{ij} &= \frac{4}{3h\Gamma(1/2)}\left(\sum_{r=0}^{i-2}\phi_j'\left(\frac{x_{r+1}+x_r}{2}\right)\left((x_i-x_r)^{3/2}\right.\right. \\ &\quad \left.\left. -(x_i-x_{r+1})^{3/2}+(x_{i-1}+x_{r+1})^{3/2}-(x_{i-1}-x_r)^{3/2}\right)\right) \\ &\quad +\frac{4}{3h^{1/2}\Gamma(1/2)}\phi_j'\left(\frac{x_{i-1}+x_i}{2}\right)-\frac{4}{3h^{1/2}}\Gamma(1/2)\phi_j'\left(\frac{x_i+x_{i+1}}{2}\right) \\ &\quad -\frac{4}{3h\Gamma(1/2)}\left(\sum_{r=0}^{i-1}\phi_j'\left(\frac{x_{r+1}+x_r}{2}\right)\left((x_{i+1}-x_r)^{3/2}-(x_{i+1}-x_{r+1})^{3/2}\right.\right. \\ &\quad \left.\left. +(x_i+x_{r+1})^{3/2}-(x_i-x_r)^{3/2}\right)\right) \end{aligned}$$

Using MATLAB, coding this system of equations, we obtain the following numerical calculations.

h	$\|\|u-u_h\|\|_{L^2(\Omega)}$	cvge. rate
1/4	5.054464×10^{-3}	
1/8	1.644609×10^{-3}	1.62
1/16	4.742156×10^{-4}	1.79
1/32	1.275353×10^{-4}	1.89
1/64	3.305235×10^{-5}	1.95

Table 3.1: Galerkin FEM Estimates for Example 3

Clearly, the experimental convergence rate is converging to two, as h goes to zero. This rate of convergence is discussed in more detail in [3]. Now, we turn our attention to a more general example.

Example 4. Let $u = x^2 - x^3$, $\beta = \frac{1}{2}$, $a = 2$, $b = c = 1$, and consider the boundary value problem

$$
\begin{aligned}
-2D(p_0 D_x^{-1/2} + q_x D_1^{-1/2})Du + Du + u &= f \qquad (3.3)\\
u(0) &= 0\\
u(1) &= 0
\end{aligned}
$$

We again use the Galerkin method to approximate the solution in the n-dimensional space $V_n = span\{\phi_0, \phi_1, ..., \phi_{n-1}\}$, with the basis functions as defined in Example 3. Observe that

$$
-2D(p_0 D_x^{-1/2} + q_x D_1^{-1/2})Du + Du + u = -2pD_0 D_x^{-1/2} Du - 2qD_x D_1^{-1/2} Du + Du + u
$$

Multiplying (3.3) through by $\phi_i(x)$ and integrating we have

$$
\begin{aligned}
\frac{2p}{\Gamma(\frac{1}{2})} \int_0^1 \int_0^x (x-s)^{-1/2} Du(s) ds \phi_i^{'}(x) dx + \frac{2q}{\Gamma(\frac{1}{2})} \int_0^1 \int_x^1 (s-x)^{-1/2} Du(s) ds \phi_i^{'}(x) dx\\
+ \int_0^1 Du(x)\phi_i(x) dx + \int_0^1 u(x)\phi_i(x) dx = \int_0^1 f(x)\phi_i(x) dx \quad (3.4)
\end{aligned}
$$

Again, we approximate $u(x)$ as

$$
u_n(x) = \sum_{j=0}^{n-1} c_j \phi_j(x). \qquad (3.5)
$$

Substituting (3.5) into (3.4) yields

$$\frac{2p}{\Gamma(\frac{1}{2})}\int_0^1 (x-s)^{-1/2}\left(\sum_{j=0}^{n-1} c_j\phi_j(s)\right)' ds\phi_i'(x)dx$$
$$+\frac{2q}{\Gamma(\frac{1}{2})}\int_0^1\int_x^1 (s-x)^{-1/2}\left(\sum_{j=0}^{n-1} c_j\phi_j(s)\right)' ds\phi_i' dx$$
$$+\int_0^1\left(\sum_{j=0}^{n-1} c_j\phi_j(s)\right)'\phi_i(x)dx+\int_0^1\sum_{j=0}^{n-1} c_j\phi_i(s)\phi_j(x)dx$$
$$=\int_0^1 f(x)\phi_i(x)dx$$

Simplifying we have

$$\sum_{j=0}^{n-1} c_j\left(\frac{2p}{\Gamma(\frac{1}{2})}\int_0^1\int_0^x (x-s)^{-1/2}\phi_j'(s)ds\phi_i'(x)dx+\frac{2q}{\Gamma(\frac{1}{2})}\int_0^1\int_x^1 (s-x)^{-1/2}\phi_j'(s)ds\phi_i'(x)dx\right.$$
$$\left.+\int_0^1\phi_i(x)\phi_j'(x)dx+\int_0^1\phi_i(x)\phi_j(x)dx\right)$$
$$=\int_0^1 f(x)\phi_i(x)dx$$

Let $\mathbf{c}$ be the corresponding coefficient vector, then $\mathbf{c}$ is the solution to the linear system $G\mathbf{c}=\mathbf{f}$ where

$$\begin{aligned} G_{ij} = \quad & \frac{2p}{\Gamma(\frac{1}{2})}\int_0^1\int_x^1 (x-s)^{-1/2}\phi_j'(s)ds\phi_i'(x)dx+\frac{2q}{\Gamma(\frac{1}{2})}\int_0^1\int_x^1 (s-x)^{-1/2}\phi_j'(s)ds\phi_i'(x)dx \\ & +\int_0^1\phi_i(x)\phi_j'(x)dx+\int_0^1\phi_i(x)\phi_j(x)dx \end{aligned} \tag{3.6}$$

and

$$f_i=\int_0^1 f(x)\phi_i(x)dx$$

We simplify the expression for G_{ij}. We do this by considering each term individually. From

(3.1) it follows that

$$
\begin{aligned}
&\frac{2p}{\Gamma(\frac{1}{2})}\int_0^1\int_0^x (x-s)^{-1/2}\phi_j^{'}(s)\phi_i^{'}(x)dx = \\
&\frac{8p}{3h\Gamma(\frac{1}{2})}\left(\sum_{r=0}^{i-2}\phi_j^{'}\left(\frac{x_{r+1}+x_r}{2}\right)((x_i-x_r)^{3/2}-(x_i-x_{r+1})^{3/2}\right. \\
&\left.+(x_{i-1}+x_{r+1})^{3/2}-(x_{i-1}-x_r)^{3/2})\right) \\
&+\frac{8p}{3h^{1/2}\Gamma(\frac{1}{2})}\phi_j^{'}\left(\frac{x_{i-1}+x_i}{2}\right)-\frac{8p}{3h^{1/2}\Gamma(\frac{1}{2})}\phi_j^{'}\left(\frac{x_i+x_{i+1}}{2}\right) \\
&-\frac{8p}{3h\Gamma(\frac{1}{2})}\left(\sum_{r=0}^{i-1}\phi_j^{'}\left(\frac{x_{r+1}+x_r}{2}\right)((x_{i+1}-x_r)^{3/2}-(x_{i+1}-x_{r+1})^{3/2}\right. \\
&\left.+(x_i+x_{r+1})^{3/2}-(x_i-x_r)^{3/2})\right) \qquad (3.7)
\end{aligned}
$$

For the second term in (3.6) we have

$$
\begin{aligned}
&\frac{2q}{\Gamma(\frac{1}{2})}\int_0^1\int_x^1 (s-x)^{-1/2}\phi_j^{'}(s)ds\phi_i^{'}(x)dx \\
= &\frac{2q}{\Gamma(\frac{1}{2})}\int_{x_{i-1}}^{x_{i+1}}\int_x^1 (s-x)^{-1/2}\phi_j^{'}(s)ds\phi_i^{'}(x)dx \\
= &\frac{2q}{\Gamma(\frac{1}{2})}\left(\int_{x_{i-1}}^{x_i}\int_x^1 (s-x)^{-1/2}\phi_j^{'}(s)ds\left(\frac{1}{h}\right)+\int_{x_i}^{x_{i+1}}\int_x^1 (s-x)^{-1/2}\phi_j^{'}(s)ds\left(\frac{-1}{h}\right)\right) \\
= &\frac{2q}{h\Gamma(\frac{1}{2})}\left(\int_{x_{i-1}}^{x_i}\int_x^1 (s-x)^{-1/2}\phi_j^{'}(s)dsdx-\int_{x_i}^{x_{i+1}}\int_x^1 (s-x)^{-1/2}\phi_j^{'}(s)dsdx\right) \\
= &\frac{2q}{h\Gamma(\frac{1}{2})}\left(\int_{x_{i-1}}^{x_i}\int_x^{x_i} (s-x)^{-1/2}\phi_j^{'}(s)dsdx+\int_{x_{i-1}}^{x_i}\sum_{r=i}^{n-1}\int_{x_r}^{x_{r+1}} (s-x)^{-1/2}\phi_j^{'}(s)dsdx\right. \\
&\left.-\int_{x_i}^{x_{i+1}}\int_x^{x_{i+1}} (s-x)^{-1/2}\phi_j^{'}(s)dsdx-\int_{x_i}^{x_{i+1}}\sum_{r=i+1}^{n-1}\int_{x_r}^{x_{r+1}} (s-x)^{-1/2}\phi_j^{'}(s)dsdx\right)
\end{aligned}
$$

$$
\begin{aligned}
= \quad & \frac{2q}{h\Gamma(\frac{1}{2})}\left(\phi_j^{'}\left(\frac{x_{i-1}+x_i}{2}\right)\int_{x_{i-1}}^{x_i}\int_{x}^{x_i}(s-x)^{-1/2}dsdx\right.\\
& +\sum_{r=i}^{n-1}\phi_j^{'}\left(\frac{x_r+x_{r+1}}{2}\right)\int_{x_{i-1}}^{x_i}\int_{x_r}^{x_{r+1}}(s-x)^{-1/2}dsdx\\
& -\phi_j^{'}\left(\frac{x_i+x_{i+1}}{2}\right)\int_{x_i}^{x_{i+1}}\int_{x}^{x_{i+1}}(s-x)^{-1/2}dsdx\\
& \left.-\sum_{r=i+1}^{n-1}\phi_j^{'}\left(\frac{x_r+x_{r+1}}{2}\right)\int_{x_i}^{x_{i+1}}\int_{x_r}^{x_r+1}(s-x)^{-1/2}dsdx\right)
\end{aligned}
$$

Integrating we obtain

$$
\begin{aligned}
& \frac{8q}{3h\Gamma(\frac{1}{2})}\left(\phi_j^{'}\left(\frac{x_{i-1}+x_i}{2}\right)(x_i-x_{i-1})^{3/2}\right.\\
& +\sum_{r=i}^{n-1}\phi_j^{'}\left(\frac{x_r+x_{r+1}}{2}\right)(-(x_{r+1}-x_i)^{3/2}+(x_{r+1}-x_{i-1})^{3/2}\\
& +(x_r-x_i)^{3/2}-(x_r-x_{i-1})^{3/2})\\
& -\phi_j^{'}\left(\frac{x_i+x_{i+1}}{2}\right)((x_{i+1}-x_i)^{3/2})\\
& -\sum_{r=i+1}^{n-1}\phi_j^{'}\left(\frac{x_r+x_{r+1}}{2}\right)(-(x_{r+1}-x_{i+1})^{3/2}+(x_{r+1}-x_i)^{3/2}\\
& \left.+(x_r+x_{i+1})^{3/2}-(x_r-x_i)^{3/2})\right) \qquad (3.8)
\end{aligned}
$$

Also, the third and fourth terms of (3.6) are given by

$$
\int_0^1 \phi_j^{'}(x)\phi_i(x)dx = \begin{cases} \frac{1}{2}, & i-j=-1 \\ -\frac{1}{2}, & i-j=1 \\ 0, & otherwise \end{cases} \qquad (3.9)
$$

and

$$\int_0^1 \phi_i(x)\phi_j(x)dx = \begin{cases} \frac{2}{3}h, & i = j \\ \frac{1}{6}h, & |i-j| = 1 \\ 0, & otherwise \end{cases} \tag{3.10}$$

The derivation of these integrals can be found in [6]. Combining (3.7), (3.8), (3.9), and (3.10) we find a suitable representation for G_{ij}. We use MATLAB to find the numerical results given in Table 3.2 for different p and q values.

h	$\|\|u - u_h\|\|_{L^2(\Omega)}$ $p = 1, q = 0$	cvge. rate	$\|\|u - u_h\|\|_{L^2(\Omega)}$ $p = 0, q = 1$	cvge. rate	$\|\|u - u_h\|\|_{L^2(\Omega)}$ $p = 1/2, q = 1/2$	cvge. rate
1/4	7.624559×10^{-3}		5.098315×10^{-3}		2.959363×10^{-3}	
1/8	2.064648×10^{-3}	1.88	1.59486×10^{-3}	1.68	9.948354×10^{-3}	1.57
1/16	5.34346×10^{-4}	1.95	4.529012×10^{-4}	1.82	2.966672×10^{-4}	1.75
1/32	1.35577×10^{-4}	1.98	1.216846×10^{-4}	1.90	8.148339×10^{-4}	1.86
1/64	3.40909×10^{-5}	1.99	3.162439×10^{-5}	1.94	2.15481×10^{-5}	1.92

Table 3.2: Galerkin FEM Estimates for Example 4 for different p and q values

3.2 Hierarchical Matrix Construction for the Steady State FADE

Unlike Example 1, we do not use a Taylor Series expansion to find an outer-product form for the approximation of admissible blocks. Instead, we will use a Singular Value Decomposition (SVD).

Definition 12. Let A be an $m \times n$ matrix. A singular value decomposition (SVD) of A is a factorization $A = U\Sigma V^T$ where U is a $m \times m$ unitary matrix, V is a $n \times n$ unitary matrix, and Σ is a diagonal matrix of size $m \times n$. Furthermore, it is assumed that the diagonal entries, denoted by σ_j of Σ are nonnegative and in nondecreasing order; that is $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_p$, where $p = min(m, n)$.

With the preceding definition, we have the following well known theorem.

Theorem 6. *Every $m \times n$ matrix A has a singular value decomposition.*

We will use the fact that every matrix has a singular value decomposition in combination with the following theorem called the Eckart-Young Theorem.

Theorem 7. *Let A be a $m \times n$ matrix, with SVD $A = U\Sigma V^T$, with $m \geq n$. For $k \in \mathbb{N}$ satisfying $k \leq n$, we have that*

$$min_{B,rank(B)\leq k}||A - B|| = ||A - A_k|| = ||\Sigma - \Sigma_k||$$

where $A_k = U\Sigma_k V^T \in \mathbb{R}^{m\times n}$ and $\Sigma_k = diag(\sigma_1, \sigma_2, ..., \sigma_k, 0, 0, ..., 0) \in \mathbb{R}^{m\times n}$

The Eckhart-Young theorem, gives us a way of controlling the rank of the approximation used for admissible blocks. Also it gives us a way of approximating admissible blocks with the best possible matrix of a chosen rank. Before we discuss numerical results, we describe how to use the SVD and Eckhart-Young Theorem to assemble a given matrix into outer product form.

Let A be a given $m \times n$ matrix. Also, let $A = U\Sigma V^T$ be its SVD and it's best rank k approximation $A_k = U\Sigma_k V^T$. Let $Q = U\Sigma_k$. Note that Q will have $n - k$ zero columns. Let R be the matrix formed from the first k columns of Q. Now, let S be the matrix formed from the first k columns of V. Then an approximation for A in outer-product form, $A \approx RS^T$, where $R \in \mathbb{R}^{m\times k}$ and $S \in \mathbb{R}^{n\times k}$.

Now, we are in position to discuss the construction of Hierarchical matrices that will approximate the coefficient matrix for the two previous examples. Note that the functions $f(s, x) = (x - s)^{-1/2}$ and $h(s, x) = (s - x)^{-1/2}$ are undefined when $x = s$, so the admissibility condition given in Chapter 1 gives a suitable partitioning of $[0, 1] \times [0, 1]$.

3.3 Inadmissible Blocks

For the inadmissible blocks $t \times s \subset I \times I$ we compute the entries as usual

$$\begin{aligned} G_{ij} &= \frac{2p}{\Gamma(\frac{1}{2})} \int_0^1 \int_x^1 (x-s)^{-1/2} \phi_j^{'}(s) ds \phi_i^{'}(x) dx + \frac{2q}{\Gamma(\frac{1}{2})} \int_0^1 \int_x^1 (s-x)^{-1/2} \phi_j^{'}(s) ds \phi_i^{'}(x) dx \\ &\quad + \int_0^1 \phi_i(x) \phi_j^{'}(x) dx + \int_0^1 \phi_i(x) \phi_j(x) dx \end{aligned}$$

and we store the inadmissible blocks as discussed in Definition 4.

3.4 Admissible Blocks

For the admissible blocks $t \times s \subset I \times I$ we compute the exact values with

$$\begin{aligned} G_{ij} &= \frac{2p}{\Gamma(\frac{1}{2})} \int_0^1 \int_x^1 (x-s)^{-1/2} \phi_j^{'}(s) ds \phi_i^{'}(x) dx + \frac{2q}{\Gamma(\frac{1}{2})} \int_0^1 \int_x^1 (s-x)^{-1/2} \phi_j^{'}(s) ds \phi_i^{'}(x) dx \\ &\quad + \int_0^1 \phi_i(x) \phi_j^{'}(x) dx + \int_0^1 \phi_i(x) \phi_j(x) dx \end{aligned}$$

then, for a specified maximum rank, k, we approximate $G|_{t \times s} \approx RS^T$ as discussed previously in Section 3.2.

3.5 Numerical Results for the FADE

Using the GMRES algorithm to solve the approximate linear systems, we obtain the following results (full rank) for Examples 3 and 4, respectively.

Observe that these numbers are in agreement with the results previously given in Table 3.1 and Table 3.2.

h	$\|\|u-u_h\|\|_{L^2(\Omega)}$	cvge. rate
1/4	5.893497×10^{-3}	
1/8	1.815028×10^{-3}	1.70
1/16	4.979494×10^{-4}	1.87
1/32	1.30793×10^{-4}	1.93
1/64	3.348411×10^{-5}	1.97

Table 3.3: H-Matrix Approximation for Example 3

h	$\|\|u-u_h\|\|_{L^2(\Omega)}$ $p=1, q=0$	cvge. rate	$\|\|u-u_h\|\|_{L^2(\Omega)}$ $p=0, q=1$	cvge. rate	$\|\|u-u_h\|\|_{L^2(\Omega)}$ $p=1/2, q=1/2$	cvge. rate
1/4	8.186032×10^{-3}		5.822182×10^{-3}		4.999207×10^{-3}	
1/8	2.542884×10^{-3}	1.69	1.89936×10^{-3}	1.62	1.491161×10^{-3}	1.75
1/16	7.093784×10^{-4}	1.84	4.918516×10^{-4}	1.95	3.786135×10^{-4}	1.98
1/32	1.883940×10^{-4}	1.91	1.276019×10^{-4}	1.95	9.42999×10^{-5}	2.00
1/64	4.86371×10^{-5}	1.95	3.257662×10^{-5}	1.97	2.355234×10^{-5}	2.00

Table 3.4: H-Matrix Approximation for Example 4 for different values of p and q

3.6 Rank Experiments

As our goal is to approximate our derived stiffness matrix with submatrices of low rank, in the sufficiently smooth portions of $[0,1] \times [0,1]$, we have investigated the effect of using different rank matrix approximations to the admissible blocks. Given in Tables 3.5, we present results for rank one approximations for each admissible block in Example 4.

h	$\|\|u-u_h\|\|_{L^2(\Omega)}$ $p=1, q=0$	cvge. rate	$\|\|u-u_h\|\|_{L^2(\Omega)}$ $p=0, q=1$	cvge. rate	$\|\|u-u_h\|\|_{L^2(\Omega)}$ $p=1/2, q=1/2$	cvge. rate
1/4	8.186032×10^{-3}		5.822182×10^{-3}		4.999207×10^{-3}	
1/8	2.546919×10^{-3}	1.68	1.859935×10^{-3}	1.65	1.486843×10^{-3}	1.75
1/16	8.178375×10^{-4}	1.64	3.600902×10^{-4}	2.37	4.105771×10^{-4}	1.86
1/32	8.495415×10^{-4}	-0.05	7.888035×10^{-4}	-1.13	8.274934×10^{-5}	-1.01
1/64	2.660558×10^{-3}	-1.65	2.538944×10^{-3}	-1.69	2.918238×10^{-3}	-1.82

Table 3.5: All Admissible Blocks being Rank One in Example 4

As one can see from Table 3.5, the rate of convergence is not preserved as h tends toward zero. Our question now becomes how to preserve the rate of convergence. In one of our experiments, we used the norm and a given tolerence (as a percentage) to find the rank of the admissible blocks. Let τ be a given tolerance. For each admissible block, we first find the two-norm of the admissible block, call this quanity η, and multiply η by τ and use A_k to approximate the block, where k is the smallest whole number such that $\tau * \eta \leq ||A_k||_2$. The following table shows results for given tolerences for $p = 1$, $q = 0$ in Example 4.

h	$\|\|u - u_h\|\|_{L^2(\Omega)}$ 50%	cvge. rate	$\|\|u - u_h\|\|_{L^2(\Omega)}$ 99%	cvge. rate	$\|\|u - u_h\|\|_{L^2(\Omega)}$ 99.9%	cvge. rate
1/4	8.186032×10^{-3}		8.186032×10^{-3}		8.186032×10^{-3}	
1/8	2.546919×10^{-3}	1.68	2.546919×10^{-3}	1.68	2.546919×10^{-3}	1.68
1/16	8.178375×10^{-4}	1.64	8.178375×10^{-4}	1.64	8.135502×10^{-4}	1.65
1/32	8.495415×10^{-4}	-0.05	8.495415×10^{-4}	-0.05	6.812155×10^{-4}	0.26
1/64	2.660558×10^{-3}	-1.65	2.660558×10^{-3}	-1.65	1.727734×10^{-3}	-1.34

Table 3.6: Adaptive Rank for Admissible Blocks using the Norm for Example 4

As one can see from Table 3.6 the tolerence doesn't influence the approximation until the tolerence is sufficiently close to 100%. This is because the largest singular value in most blocks is several orders greater than the next largest singular value. Since this method does not suffice, we move on to a size-tolerence (as a percentage) algorithm. Let S be the size of the admissible block. We multiply S by the given tolerence τ, and round up to the nearest whole number, this whole number will be the rank of the admissible block. Below is the results of this experiment with $p = 1$, $q = 0$ in Example 4.

h	$\|\|u - u_h\|\|_{L^2(\Omega)}$ 50%	cvge. rate	$\|\|u - u_h\|\|_{L^2(\Omega)}$ 51%	cvge. rate	$\|\|u - u_h\|\|_{L^2(\Omega)}$ 90%	cvge. rate
1/4	8.186032×10^{-3}		8.186032×10^{-3}		8.186032×10^{-3}	
1/8	2.546919×10^{-3}	1.68	2.542884×10^{-3}	1.69	2.542884×10^{-3}	1.69
1/16	8.138958×10^{-4}	1.65	7.093784×10^{-4}	1.84	7.093784×10^{-4}	1.84
1/32	6.804931×10^{-4}	0.26	1.883940×10^{-4}	1.91	1.883940×10^{-4}	1.91
1/64	1.712048×10^{-3}	-1.33	4.863716×10^{-5}	1.95	4.86371×10^{-5}	1.95

Table 3.7: Adaptive Rank for Admissible blocks using the Size of the Block

From Table 3.7, one can see that when the tolerence given is greater than 50%, we obtain the full-rank approximation. This is still problematic, as the amount of storage in such cases is then $O(n^2)$.

3.7 Future Work

In future work, it would be interesting to study the relationship between the rank of admissible blocks and the rate of convergence. Some questions to investigate are: What rank of admissible blocks is optimal in terms of memory and approximation? How does the FADE and a Hierarchical Matrix method behave in higher dimensions, $\mathbb{R}^n, n > 1$?, Can we rigorously bound the error of using the discussed H-matrix schemes? For future work, in addition to various rank approximation options, an investigation into different schemes of domain partitioning for generating inadmissible and admissible blocks would be interesting. This may strongly affect the approximation property of the resulting hierarchical matrices.

3.8 Conclusion

We have presented an introduction to H-matrices and approximations of FADE using H-matrices. We have introduced notation and several algorithms to aid in this discussion. We have also shown that using the norm to decide the rank of admissible blocks may not be a suitable approach.

Bibliography

[1] Borm, S., Grasedyck,L., and Hackbusch, W., Hierarchical Matrices, June 2006, Max-Planck-Institut fur Mathematik in den Naturwissenschaften Leipzig, (available at www.mis.mpg.de/preprints/ln/lecturenote-2103.pdf), 2006.

[2] Huang, Q., A Finite Element Solution for the Fractional Advection- Dispersion Equation, Advances in water resources, 12, 1578-1589, (2008)

[3] Ervin, V.J., and Roop, J.P., Variational Formulation for the Stationary Fractional Advection Dispersion Equation, Numer. Methods for Partial Differential Equations, 22, 558-576, (2006)

[4] Brenner, S. and Scott, L.R., *The Mathematical Theory of Finite Element Methods*, Springer-Verlag, New York, 1994.

[5] Bebendorf, M., *Hierarchical Matrices A Means to Efficiently Solve Elliptic Boundary Value Problems Series: Lecture Notes in Computational Science and Engineering*, Springer-Verlag, Berlin, Heidelberg, 2008.