1. Introduction

A package to compress matrices having low-rank structure totals 10 7 for a 45 by 45 matrix. The storage of such a matrix in a dense float format would require 10 14 bytes.

Figure 3: Illustrates the efficiency of AIGA application. The matrix is compressed using the matrix compression tool and the compressed matrix is applied to the data.

2. Convergence Analysis

Using multiple quadrants that collectively cover the data, the algorithm has a small aspect ratio. Let the number of quadrants be m, the number of elements be n, and the number of processors be p. The algorithm has a complexity of O(nm/p).

Figure 9: Shows the convergence of AIGA compared to the naive method. The convergence rate is measured by the number of iterations required for convergence.

3. Time-Dependent Problems

In general, time-dependent problems are more challenging than static problems. However, the performance of AIGA remains relatively consistent. The method has been shown to be efficient and scalable for both small and large problems.

Figure 12: Illustrates the application of AIGA on a time-dependent problem. The method shows good performance even for long time simulations.

4. Appendices

The appendices provide additional details on the implementation and performance of AIGA. The appendices also include a discussion on the limitations of AIGA and suggestions for future work.

Figure 15: Shows the comparison of AIGA with other methods. AIGA is shown to be faster and more accurate than other methods for long time simulations.

Acknowledgments

The authors would like to thank...