MAP55672 Case Study 2

Krylov Subspace Methods: GMRES Implementation

Student Name: James Naughton
Student No: 24357517

March 2025

1. Introduction

This report presents the implementation and evaluation of the GMRES algorithm as part of Case Study 2: Krylov Subspace Methods. The GMRES method is implemented in both serial and parallel versions to solve large sparse linear systems of the form $Ax = b$. The matrix A is tridiagonal with constant diagonals, and the vector b is constructed based on a linearly increasing sequence.

2. Arnoldi Iteration

The Arnoldi iteration constructs an orthonormal basis for the Krylov subspace $K_m(A, u)$ and yields an upper Hessenberg matrix H. This forms the basis of the GMRES algorithm.

3. Serial Implementation of GMRES

The file `Q2.c` contains a serial implementation of GMRES. This algorithm begins by computing an inital residual $r = b - Ax$, and normalising it to form the first Krylov basis vector. An Arnoldi process is then used to build an orthonormal basis V and upper Hessenberg matrix H representing the projection of A in the Krylov subspace. The matrix A is mainly 0s so in order to optimise memory the elements are stored as arrays. The matrix-vector multiplication function is defined appropriately. This GMRES method minimises the residual by solving the least squares problem $Hy = \beta e_1$ using backward substitution. The approximate solution x is updated by $x = Vy$. This implementation uses classical Gram-Schmidt orthogonalisation and terminates early if the residual is sufficiently small.

Results

The normalized residuals $\|r_k\|_2/\|b\|_2$ for various dimensions n were recorded and used to assess convergence. These are plotted in below using a semi-log scale. The convergence rate of GMRES appears to deteriorate as n increases. This is due to the fact that as the problem size increases so does the condition number, making the Krylov subspace less effective unless preconditioners are used.

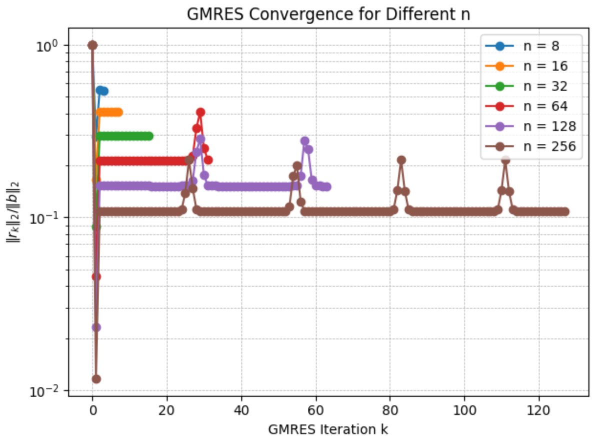

Figure 1: Convergence of GMRES for various n

4. Parallel Implementation of GMRES

The parallel implementation of GMRES uses OpenMP to accelerate computational steps of the algorithm. This version parallelises the initialisation of the b vector, the computation of residuals and normalised vectors, the matrix vector multiplication, the vector update steps and the orthogonalisation of the Arnoldi iterations. The dot product uses OpenMP reduction in order to ensure thread safe accumulation. The backward substitution for solving the least-squares problem remains serial due to the dependencies between iterations. In order to test for correctness, one could use known systems to check that the approximation is the true answer.

5. Conclusion

In this case study, the GMRES algorithm was successfully implemented in both serial and parallel forms to solve large sparse linear systems. The serial implementation focused on memory efficiency and algorithmic clarity, while the parallel implementation leveraged OpenMP to accelerate key computational steps. Results from the serial version demonstrated the impact of matrix size on convergence, highlighting the increasing difficulty of solving poorly conditioned systems without preconditioning.