MAP55672 Case Studies 3

Conjugate Gradient (CG) Solver

Student Name: James Naughton
Student No: 24357517

April 2025

1 Introduction

This report presents the setup and implementation of the Conjugate Gradient algorithm for the solution of a square linear system $Ax = b$ for positive definite, symmetric regular matrices A and $b \neq 0$. The Poisson problem on the unit square is first investigated using symmetric finite difference approximations to express the problem as a symmetric linear system. A serial implementation of the CG algorithm is then built. The convergence of the CG algorithm then was examined.

2 Discretisation of the Poisson Problem

We consider the Poisson equation:

$$-\Delta u(x) = f(x), \quad x \in \Omega = (0,1)^2, \quad u|_{\partial\Omega} = 0,$$

with $f(x) = 2\pi^2 \sin(\pi x_1)\sin(\pi x_2)$. Using central finite differences on a uniform $N \times N$ grid, the Laplacian operator Δ is approximated by a five-point stencil, resulting in a sparse matrix A of size $(N-1)^2 \times (N-1)^2$.

The grid points are ordered lexicographically to map the 2D problem to a 1D system $Ay = b$, where y contains the solution values $u(x)$ at interior grid points. The right-hand side vector b is computed by evaluating $f(x)$ at these points and applying the discretisation.

3 Serial Implementation of CG Algorithm

The CG method was implemented in C (`Question2.c`) and applied to the Poisson problem for increasing values of N. The algorithm iteratively minimizes the quadratic form associated with A, using:

- Initial guess x_0
- Residual $r_0 = b - Ax_0$
- Search directions updated recursively

- Termination when $\|r_k\| \leq 10^{-8}$

Efficient memory handling and avoidance of redundant calculations improved performance.

3.1 Results

Table 1 presents the number of iterations and runtime for varying grid sizes:

Grid Size (N)	Iterations	Time (s)
8	30	0.0004
16	63	0.0005
32	127	0.0054
64	254	0.0303
128	511	0.2148
256	1033	1.7161

Table 1: CG performance for the Poisson problem

Figure 1 shows the solution $u(x)$ for $N = 256$.

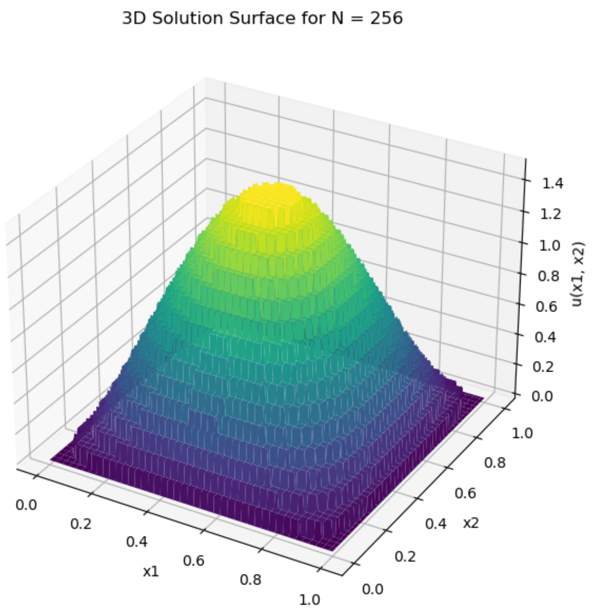

Figure 1: Surface plot of solution $u(x)$ for $N = 256$

Figure 2 shows the relationship between time and N.

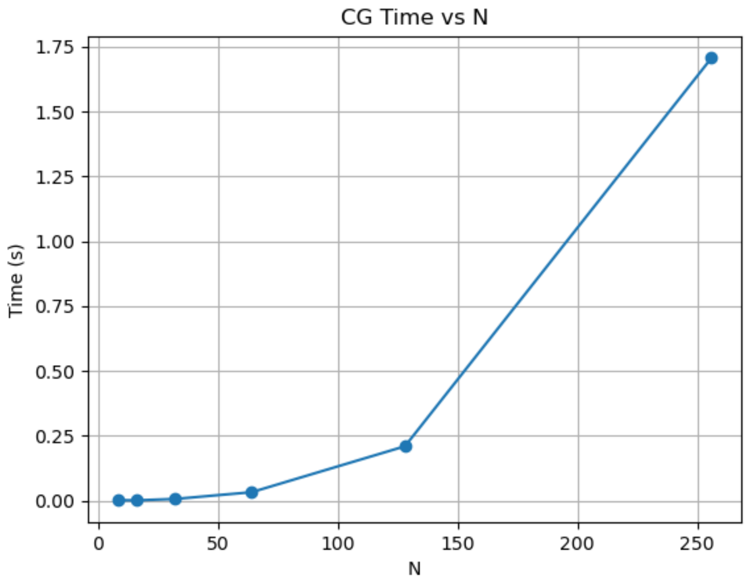

Figure 2: Relationship between t and N

Figure 3 shows the relationship between iterations and N.

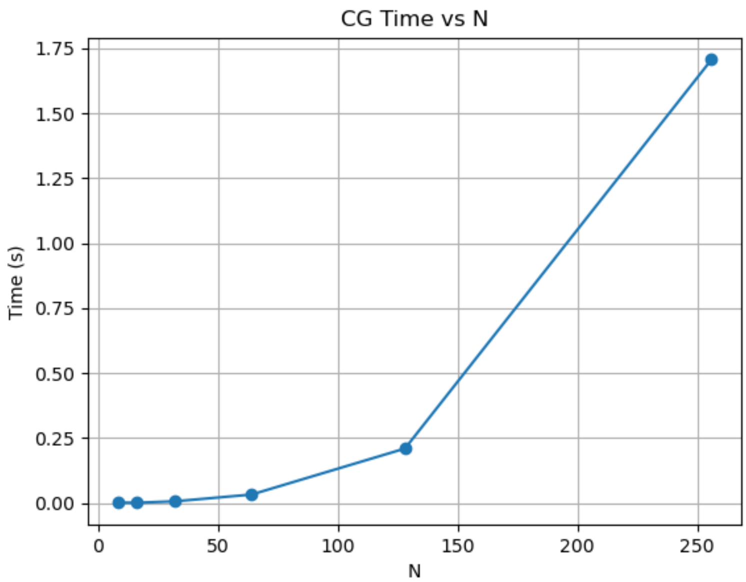

Figure 3: Relationship between iterations and N

4 Convergence of CG on Dense System

A separate implementation (`Question3.c`) was developed to study the convergence of CG applied to a dense matrix:

$$A_{ij} = \frac{N - |i - j|}{N}, \quad b_j = 1$$

for $N \in \{100, 1000, 10000\}$. The stopping criterion was:

$$\|r_k\| \leq \max(\texttt{reltol} \cdot \|r_0\|, 0), \quad \texttt{reltol} = \sqrt{\epsilon_{\text{machine}}}$$

At each iteration, the residual norm was recorded and compared to the theoretical bound:

$$\|e_k\| \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \|e_0\|,$$

where $\kappa = \frac{\lambda_{\max}}{\lambda_{\min}}$ is the spectral condition number of A.

Results

Figure 4 plots the convergence history of the residual norm for $N = 10000$.

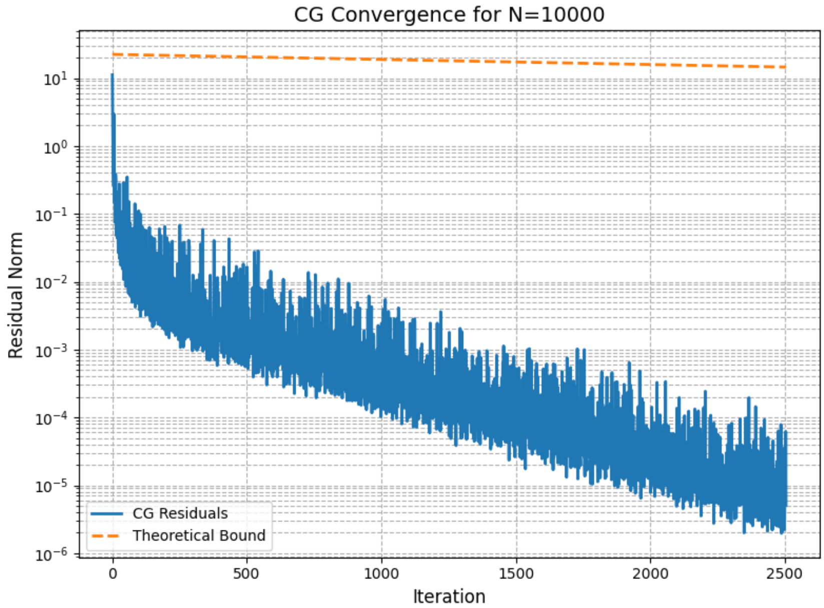

Figure 4: CG residual norm for $N = 10000$

5 Conclusion

Two implementations of the Conjugate Gradient algorithm were developed and tested:

- A matrix-free solver for the 2D Poisson equation showed efficient convergence and scalability.
- A dense matrix solver enabled analysis of CG convergence behavior and validated theoretical bounds.

Both methods achieved the desired tolerance of 10^{-8}. The results highlight CG's efficiency for large symmetric positive-definite systems and underscore the importance of matrix conditioning in convergence performance.