MAP55672 (2024-25) — Case studies 3

Conjugate Gradient (CG) solver

`(Lecturer: P. Fritzsch, mailto:fritzscp@tcd.ie)`

Instructions

- Complete all parts of the assignment by **midnight on Wednesday 16th April 2025**. Create a git repository with your solutions and submit the link to me by e-mail.
- The git repository should include all source files and a summary pdf containing a description of solutions (incl. tables and plots, if necessary). The summary report should reflect your understanding of the material and contain a short description of your submitted code. As such it should contain all necessary information to (compile and) run your code for validation purposes.
- *It is strictly forbidden to use code generating tools such as LLM's. Their use will result in 0 marks for this case study.*

3 The CG algorithm

The goal of this case study is to implement the CG algorithm for the solution of a square linear system $Ax = b$ for positive definite, symmetric regular matrices A and right hand sides $b \neq 0$.

3.1 Basics: The Poisson problem.

Consider the Poisson problem on a unit square $x = (x_1, x_2) \in \Omega = (0,1)^2$ with function $f : \Omega \to \mathbb{R}$,

$$-\Delta u(x) = f(x)\,, \quad \text{on interior of } \Omega \qquad \Delta u(x) \equiv \left(\frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2}\right) u(x)$$

$$u(x) = 0\,, \quad \text{on the boundary } \partial\Omega\,,$$

and unknown solution function $u(x) : \Omega \to \mathbb{R}$.

Use *symmetric finite difference* approximations to the partial derivatives appearing in the Laplacian Δ, to express above's partial differential equation with specified boundary conditions as a linear system $Ay = b$ where y is a vector containing the solution function $u(x)$ at discrete values of $x = (ih, jh)$, $0 \leq i, j \leq N$ on a regular 2D grid. Think about a convenient ordering of grid points and use it to express the finite-difference approximation of the Laplacian, A, as well as the appropriate right-hand side b.

Explain all necessary steps and explicit choices you made.

3.2 Serial implementation of CG.

Implement a serial variant of the CG algorithm (see, e.g., algorithm 6.19 in Saad) and solve the linear system defined in section 3.1 using the function $f(x) = 2\pi^2 \sin(\pi x_1)\sin(\pi x_2)$. Keep an eye on the performance through efficient data handling and explain what you did to avoid unnecessary or repetitive calculations.

Solve the system in double precision arithmetic for increasing number of grid points $N = 8, 16, 32, 64, 128, 256, \cdots$ up to a fixed residual $r = 10^{-8}$. Tabulate the corresponding number of CG iterations and time to solution.[1] Present and discuss your findings in an appropriate format, and plot the resulting function $u(x)$.

3.3 Convergence of CG.

Take your own CG implementation to study the dense linear system $Ay = b$ with $y \in \mathbb{R}^N$ and

$$(A)_{ij} = \frac{N - |i-j|}{N} \,, \qquad\qquad (b)_j = 1 \,,$$

for $1 \le i, j \le N \in \{10^2, 10^3, 10^4, 10^5\}$.

Implement the stopping criterion

$$|r_k| \le \max(\,\texttt{reltol * } |r_0|, \texttt{abstol})$$

where $r_k = Ax_k - b$ with

absolute tolerance: `abstol=0` and
relative tolerance: `reltol=`$\sqrt{\texttt{eps(double)}}$.

Here, `eps(double)` is the machine epsilon for double precision numbers on your computer. Estimate the rate of convergence of the iterative method by collecting the residual in each iteration step and compare it to the bound

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

given through the spectral condition number $\kappa = \kappa(A) = \lambda_{\max}/\lambda_{\min}$. You may obtain the condition number using an external program of your choice. Plot your results for each N and interpret your findings.

[1] To get sufficiently good timings, your computer should not perform other tasks at the same time. Furthermore, repeating a single run several times should give you an even better average run time for the algorithm.