# Worksheet 4, February 28, 2025

## 1 Solving ODEs with tridiagonal matrices

Consider the following *boundary value problem*:

$$\begin{aligned} \frac{\partial^2 g}{\partial x^2} &= f(x) \quad x \in [0,1] \\ g(0) = 0 &\quad g(1) = 0 \end{aligned} \tag{1}$$

**Q1** Find the analytical solution $g(x)$ of (1) in the case that $f(x) = 20\cos(2\pi x)$. (Hint: Integrate $f(x)$ twice and use the boundary conditions to define your unknown constants).

Sometimes we cannot find the solution analytically (e.g., if $f(x) = \sin(\exp(x))\cos(x)$), and need to solve this boundary value problem numerically.
We will solve this problem with $(n+2)$-many equally spaced points, denoted $x_i$ such that $x_0 = 0$ and $x_{n+1} = 1$ with spacing $h = 1/(n+1)$. Then, our numerical solution is $g_i \approx g(x_i)$.
A possible *finite-difference method* approximation for (1) is to write

$$\frac{1}{h^2}\left(g_{i+1} - 2g_i + g_{i-1}\right) = f(x_i)$$

This corresponds to a linear system of $n$ equations for the interior points $g_i$ (taking into account that $g_0 = g(0) = 0$ and $g_{n+1} = g(1) = 0$), which can be arranged as $Ag = f$ where:

- $A$ is a tridiagonal $\mathbb{R}^{n\times n}$ matrix with $-2/h^2$ on the diagonal, and $1/h^2$ on the super- and sub-diagonals (diagonals directly above and below the main diagonal).
- $g$ is your unknown array of $g_1$ through $g_n$.
- $f$ is an array of values $f(x_1)$ through $f(x_n)$, corresponding to the right-hand-side of (1).

Using $f(x) = 20\cos(2\pi x)$, do the following:

**Q2** For arbitrary $n$, write code to construct $A$ as a sparse matrix (in python, `scipy.sparse` library will be helpful for this), find its $LU$ factorization (`scipy.sparse.linalg.splu`), and use forward/backward substitution (`scipy.sparse.linalg.spsolve`) to solve for $g_1$ through $g_n$. Verify that your solution matches the analytical solution you found in **Q1**.

**Q3** Record the time it takes to compute the LU factorization and solve the linear system for each $n = 2^k$, $k = 5, \ldots 20$. How does the computational time vary with $n$? Does this surprise you? What would you expect if $A$ was *dense*? (Note: this relates to problem 1 on HW3).

## 2 Condition numbers and pivoted LU

**Q1** Suppose we want to solve a linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ for some $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{b} \in \mathbb{R}^n$, finding unknown $\boldsymbol{x} \in \mathbb{R}^n$. If we perturb the vector $\boldsymbol{b}$ by a relative perturbation $\|\delta\boldsymbol{b}\|_2 \,/\, \|\boldsymbol{b}\|_2$, what can we expect regarding the relative error $\|\delta\boldsymbol{x}\|_2 \,/\, \|\boldsymbol{x}\|_2$ of the solution $\boldsymbol{x}$?

Consider the linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ for

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 \\ 10^4 & 1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

**Q2** Solve for $\boldsymbol{x}$

**Q3** What are $\kappa_2(A)$ and $\kappa_\infty(A)$?

**Q4** Compute $\boldsymbol{x}$ if instead you add perturbations $\Delta\boldsymbol{b} = \left[10^{-3}, 0\right]^T$ or $\Delta\boldsymbol{b} = \left[0, 10^{-3}\right]^T$ to the right-hand-side. What do you notice? Do the relative errors in $\boldsymbol{x}$ and $\boldsymbol{b}$ agree with your answer in Q1?

**Q5** Find the LU decomposition, with and without pivoting, of a matrix $B$ given by

$$\boldsymbol{B} = \begin{bmatrix} 10^{-4} & 1 \\ 1 & 1 \end{bmatrix}$$

Notice that $\boldsymbol{B}$ is well-conditioned. Are both LU decompositions (with and without pivoting) also well-conditioned?