MA5233 Computational Mathematics

Assignment 3

Deadline: 12 November 2020, 7pm
Total marks: 20

1 Mixed Dirichlet and Neumann boundary conditions [10 marks]

The goal of this task is to show that $A_n u_n = b_n$ with

$$A_n = n^2 \begin{pmatrix} 2 & -1 & & & \\ -1 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -2 & 2 \end{pmatrix}, \quad u_n \approx \begin{pmatrix} u\big(\frac{1}{n}\big) \\ u\big(\frac{2}{n}\big) \\ \vdots \\ u\big(\frac{n-1}{n}\big) \\ u\big(\frac{n}{n}\big) \end{pmatrix}, \quad b_n = \begin{pmatrix} f\big(\frac{1}{n}\big) \\ f\big(\frac{2}{n}\big) \\ \vdots \\ f\big(\frac{n-1}{n}\big) \\ f\big(\frac{n}{n}\big) \end{pmatrix},$$

is a second-order convergent finite difference discretisation of the one-dimensional Poisson equation with mixed Dirichlet and Neumann boundary conditions, i.e. the problem of determining $u : [0,1] \to \mathbb{R}$ such that

$$-u''(x) = f(x) \quad \text{for all} \quad x \in [0,1], \qquad \text{and} \qquad u(0) = 0, \quad u'(1) = 0.$$

1. [1 mark] Briefly explain why the formula relating the function $u : [0,1] \to \mathbb{R}$ to the vector of point values $u \in \mathbb{R}^n$ is $u[i] = u\big(\frac{i}{n}\big)$ and not $u[i] = u\big(\frac{i}{n+1}\big)$ like in class. (One or two sentences are enough.)
2. [2 marks] Show that $u_k[i] = \sin\big(\frac{\pi}{2}\, k\, \frac{i}{n}\big)$ are eigenvectors of A_n for odd $k \in \mathbb{N}$, and determine the corresponding eigenvalues.

   Hint. You may assume without proof that $u_k[n+1] = u_k[n-1]$ for odd k.
3. [2 marks] Conclude that $\|A_n^{-1}\|_{2,n} = O(1)$ for $n \to \infty$.
4. [1 mark] Show that if $u'(1) = 0$, then

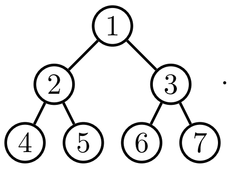
$$-u''(1) = 2n^2 \Big(-u\big(\tfrac{n-1}{n}\big) + u\big(\tfrac{n}{n}\big)\Big) + O(n^{-1}).$$

5. [1 mark] Show that $A_n e = c$, where $e[j] = \frac{j}{n^2}$, $\quad c[j] = \begin{cases} 1 & \text{if } j = n, \\ 0 & \text{otherwise.} \end{cases}$
6. [1 marks] Conclude that $\|u - u_n\|_{2,n} = O(n^{-2})$, where $u[i] = u\big(\frac{i}{n}\big)$ denotes the vector of point values of the exact solution, and $\|u\|_{2,n} = \frac{1}{\sqrt{n}} \sqrt{\sum_{i=1}^n u[i]^2}$.

   Hint. We have seen in class that $(A_n u - b_n)[j] = O(n^{-2})$ for all $j \in \{1, \dots, n-1\}$.
7. [2 marks] Complete the function `convergence()` such that it creates a plot demonstrating that $\|u - u_n\|_{2,n} = O(n^{-2})$.

2 Fill path theorem and nested dissection [6 marks]

Consider a sparse matrix A whose associated graph is given by

$G(A) =$ 

.

1. [2 marks] Determine the sparsity pattern of the LU factorisation of A. State your result as a single matrix where the diagonal entries are numbered 1 to 7, nonzero entries in A are marked with $\bullet$ and fill-in entries in $L + U$ are marked with `x`.
   You do not have to motivate your answer.
2. [2 marks] Explain why $V_{\text{sep}} = \{1\}$ is a good separator for the first step of the nested dissection recursion. (This is an open-ended question. Try to answer as comprehensively yet concisely as possible.)
3. [2 marks] Draw a copy of the graph of A where the vertices are numbered such that LU factorisation of the corresponding matrix does not incur any fill-in.
   You do not have to motivate your answer.

3 Conjugate Gradients [4 marks]

We have seen in class that the conjugate gradient algorithm can be implemented in just eight lines of code.

Algorithm 1 Conjugate gradients

```
1: $x_0 = 0, r_0 = b, p_0 = r_0$
2: for $k = 1, \dots, m$ do
3:     $\alpha_k = (r_{k-1}^T r_{k-1}) / (p_{k-1}^T A p_{k-1})$
4:     $x_k = x_{k-1} + \alpha_k \, p_{k-1}$
5:     $r_k = r_{k-1} - \alpha_k \, A p_{k-1}$
6:     $\beta_k = (r_k^T r_k) / (r_{k-1}^T r_{k-1})$
7:     $p_k = r_k + \beta_k \, p_{k-1}$
8: end for
```

1. [2 marks] Complete the function `conjugate_gradients(A,b,m)` such that it implements this algorithm.
2. [2 marks] Write a function `test()` which checks that your `conjugate_gradients()` function is correct. Your test should either print `Test passed` or `Test failed`, or it may produce a single plot for human inspection. In the latter case, please write one or two sentences to explain which features of the plot show that your code is correct.

   Note that no practical test can ever prove that your code is correct for all possible inputs. Instead of a perfect test, you should therefore aim for a test which makes it reasonably unlikely that `conjugate_gradients()` would pass if it contained a mistake.