

MAS212 Scientific Computing and Simulation

#8: Linear systems & conditioning

Key resources:

- Lec 8: <http://sam-dolan.staff.shef.ac.uk/mas212/docs/18.pdf>
- Example code: <http://sam-dolan.staff.shef.ac.uk/mas212/code/>

1. Gauss-Jordan elimination. Here we will use Gauss-Jordan elimination to solve the linear system $\mathbf{Ax} = \mathbf{b}$ to find \mathbf{x} and \mathbf{A}^{-1} .

(a) Implement the steps in Lecture 9, using the code `gauss-jordan1.py` as a starting point, for the case

$$\mathbf{A} = \begin{pmatrix} 0 & 2 & 1 \\ 2 & -1 & 1 \\ 1 & 3 & 2 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 7 \\ 3 \\ 13 \end{pmatrix},$$

(b) Using the code `gauss-jordan2.py`, take the necessary steps to solve for

$$\mathbf{A} = \begin{pmatrix} 0 & 2 & 3 \\ 0 & 1 & 2 \\ 1 & 0 & 5 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1 \\ 1 \\ 7 \end{pmatrix},$$

(c) Write a function `gaussjordan(A, b)` to implement the Gauss-Jordan method for a square array A . Your function should print the augmented matrix at each step. See `gauss-jordan3.py` for hints.

2. Wilkinson's polynomial. In Lec. 9 we looked at Wilkinson's polynomial,

$$p_n^{(0)}(x) = (x-1)(x-2)\dots(x-n) = \prod_{k=1}^n (x-k)$$

and a particular 'perturbed' version,

$$p_n(x, \epsilon) = p_n^{(0)}(x) + \epsilon x^{n-1}.$$

Clearly, $p_n^{(0)}(x)$ has roots (zeros) at $x = 1, 2, \dots, n$. The introduction of a small ($\epsilon \ll 1$) perturbation changes these roots. In the lecture I claimed that a small perturbation could have a large effect when n is large. Let's investigate this claim.

(a) In `poly_example.py` is a function to calculate the coefficients of Wilkinson's polynomial for a given n , using the `sympy` module. The function `wilkinson_polynom_coeffs(n)` returns a list of the coefficients $[a_0, a_1, a_2, \dots]$ of the unperturbed polynomial, where

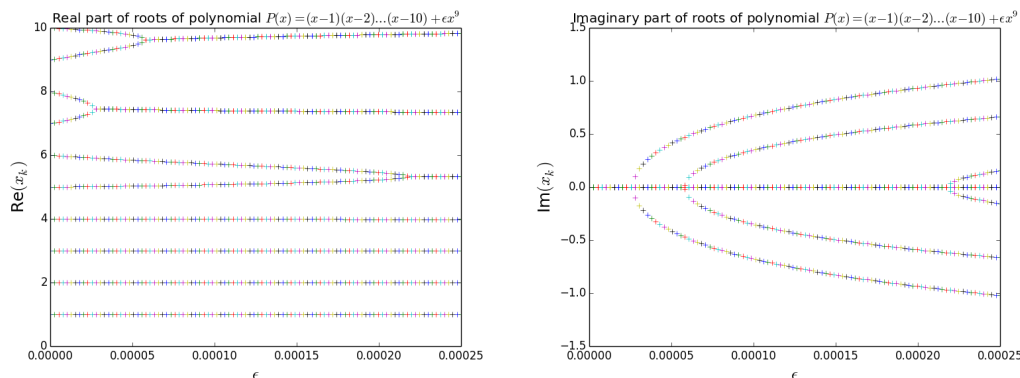
$$p_n^{(0)}(x) = a_0 x^n + a_1 x^{n-1} + a_2 x^{n-2} + \dots$$

Using `np.roots()`, find the roots of $p_n^{(0)}(x)$ for $n = 20$. What do you notice?

(b) Now calculate the roots of the perturbed polynomial $p_n(x)$ for $n = 10$ and (i) $\epsilon = 1 \times 10^{-5}$, (ii) $\epsilon = 4 \times 10^{-5}$ and (iii) $\epsilon = 8 \times 10^{-5}$. (*Hint:* To perturb the polynomial, change a_1 , the second

element in the list of coefficients.) What do you find?

(c) Make plots similar to the ones below, showing the real and imaginary parts of the roots as a function of ϵ for the case $n = 10$.



3. Row-sum norm and condition numbers.

The **row sum norm** of an $m \times n$ matrix A is defined as

$$\|A\| = \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}|$$

where a_{ij} is the element in the i th row and j th column of A . The **condition number** of a matrix is $C = \|A\| \|A^{-1}\|$, where A^{-1} is the matrix inverse.

(a) Write a function `row_sum_norm(A)` to compute the row sum norm of a numpy array or matrix. Write a function `condition_number(A)` to compute the condition number.

(b) Test your functions by computing the condition numbers of the following matrices:

$$(i) A_1 = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, \quad (ii) A_2 = \begin{pmatrix} 1 & 1 \\ 1 & 1.001 \end{pmatrix}, \quad (iii) A_3 = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{pmatrix}.$$

Check your answers using `np.linalg.cond()`. N.B. Please read the docstring for this function (`?np.linalg.cond`), as you will need to specify an optional parameter to calculate the condition number that corresponds to the row sum norm.

4. Hilbert matrices: http://en.wikipedia.org/wiki/Hilbert_matrix

Matrix A_3 in Q2(b)(iii) is an example of a Hilbert matrix.

(a) Let A_n be an $n \times n$ Hilbert matrix, and let C_n be its condition number. Write code to compute the condition number of C_n for a given n . Plot the condition number C_n as a function n , using logarithmic scales on both axes.

(b) Investigate the claim that $C_n \sim \alpha(1 + \sqrt{2})^{4n}/\sqrt{n}$, when n is large. Estimate α .