Numerical methods – Exercise 7

Šimun Šopar

The goal of Exercise 7. was to write a C++ program which for two given square matrices calculates their sum and matrix product. The program also tells if the matrices are diagonal or triangular.

PART 1 – Input

First part of the code is input. The code first takes one parameter *n*, this is the dimension of the matrices A and B. The code than takes input for the members of the matrices A and B. After that the code prints out matrices A and B. This is so the user can check that there are no errors in defining matrices A and B. Matrices are printed out one member at a time, in the corresponding row and column. Each member is formated as %7.2lf, which means they take up at least 7 spaces, two for the decimals, one for decimal dot and 4 or more for the digits (if the number is negative, one place will be taken for minus sign). This means the members will be precise to 2 digits. Also, if the members are numbers smaller than 10^4, they will all take the same amount of space on the screen, making the matrix more readable. If they are larger, the members may not be as aligned when printing the matrices. This can be changed by changing the format of the data in code

PART 2 – Sum

Function *sumAB* takes two matrices and prints out their sum from the formula $c_i = a_i + b_i$, where *C* is the sum matrix, and *c* denotes members of that matrix. Printing is formated the same as before.

PART 3 – Product

Function *prodAB* takes two matrices and prints out their matrix product from formula $c_{ij} = \sum_k a_{ik} b_{kj}$, where again *C* denotes the final, product matrix. Printing is formated as before.

PART 4 – Type

Final part of the code, function *type* takes in a matrix and determines if the matrix is triangular or diagonal or neither. It does so by going through non-diagonal members of the matrix. If members of the upper triangle of the matrix are all zero, boolean variable *upper* takes value 1. If members of lower are all zero, variable *lower* takes value 1. If both *upper* and *lower* are 1, then the matrix is diagonal. If one is 1 and the other is 0, the matrix is triangular. Else, the matrix is neither of these.