Computer Aided Numerical Methods – I

Report on Assignment – 2

Problem Statement:

Find the temperature along the length of a rod that is 1m long. Its hot end is at T = 100°C and its cold end is at T = 20°C. Use the following:

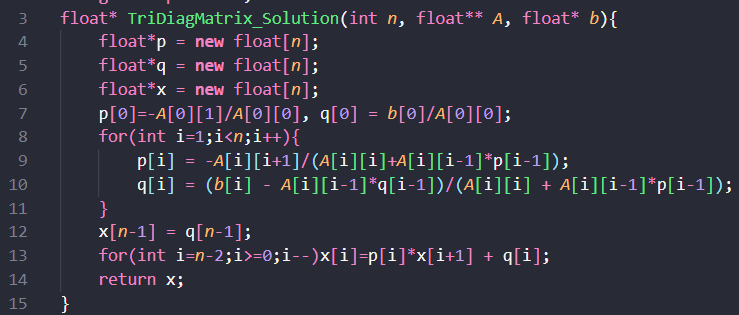
(Take )

Explanation of Code:

0 – Preprocessing

This program has been coded in C++ language. Hence, the main library ‘iostream’ has been included to allow use of the basic input and output functions of C++. ‘Using namespace std;’ prevents having to prefix those functions with “std::’.

I – Tri-Diagonal Matrix Algorithm (TDMA) Function

 This function takes a pointer to Tri-Diagonal Matrix ‘’, size of ‘’, a pointer to nx1 vector ‘’ as input and returns a pointer to solution vector ‘’ such that .

Functionality:

1. Performs the Algorithm to return solution vector ‘x’ to .
2. We can reduce the system using row operations to the system such that:

and

1. We initialize and allocate memory for three vectors: 1) x: the solution vector containing unknowns ; 2) p: the vector containing values and 3) q: the vector containing values .
2. By equating and to corresponding values of and after performing row operations, we can get that:
   * and
   * and

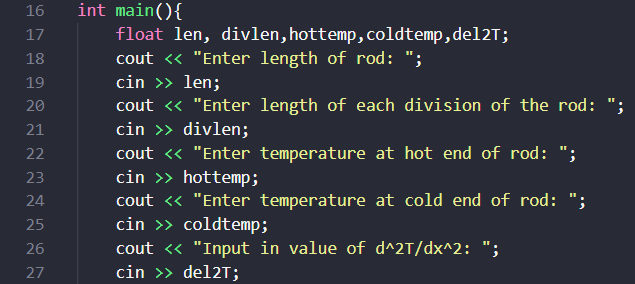
where and .

1. We assign values of and directly and then iteratively assign values of and using a loop from to .
2. From the reduced equation , we get:
3. We assign value of directly and then iteratively assign values of using a reverse loop from to .
4. With this, we obtain solution vector satisfying .

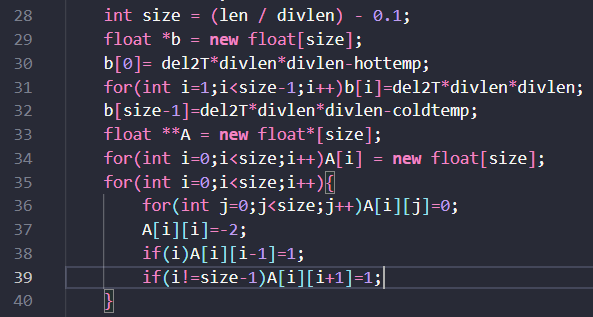
II – Main Function

This function handles user input, TDMA, and solving the system.

Functionality:

1. User Input:
   * This part of the code reads length of

rod ‘len’, length of each division of the rod ‘divlen’, temperature at hot end of rod ‘hottemp’, temperature at cold end of rod ‘coldtemp’ and value of ‘del2T’.

1. Building up Matrix A and Vector b:
   *  Now we construct the required matrix

A and vector b to get the intermediate temperatures. These intermediate temperatures will give us a good idea of the temperature along the rod. We can find these values using the approximation:

, where is temperature of the rod at the division.

Therefore, we can write

Take case for , then .

As is a boundary point, it is known to us, and we can write .

Again, take case for , where is the number of divisions. We get

.

As is also a known boundary point, we can write .

Now, we have equations with unknowns:

Left boundary equation: .

intermediate equations:

Right boundary equation: .

We can construct the equation , where

, and .

Solving this system will give us the intermediate temperatures and allow us to determine the temperature along the rod. As is tri-diagonal matrix, we can use the TDMA described above.

We can build matrix A by setting all diagonal elements to -2 and the upper and lower bi-diagonal elements to 1. Similarly, we make vector b by setting the first and last elements according to the corresponding boundary conditions and putting in the value of for the remaining elements. In accordance with the problem statement we set:

Length of rod, len = 1m

Length of division, divlen = = 0.05m

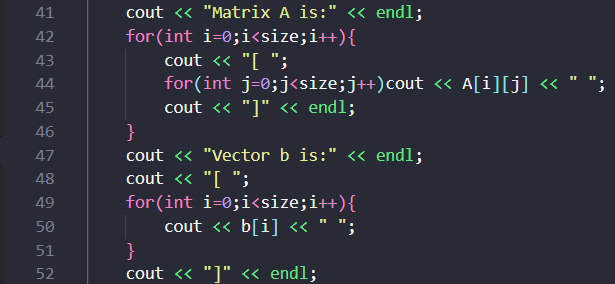
Size of matrix A and vector b = size = len/divlen – 0.1 (Due to floating point errors)

Temperature at hot end, hottemp = = 100°C

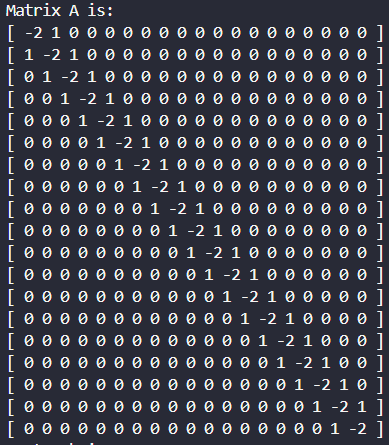
Temperature at cold end, coldtemp = = 20°C

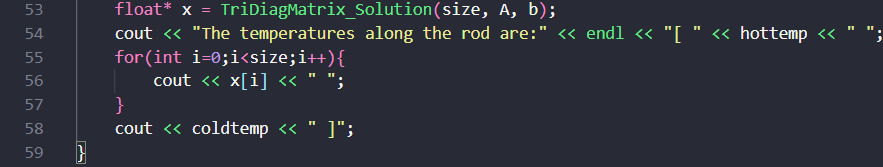
Value of , del2T = 0, then

So, our vector .

1. Printing Matrix A and Vector b:
   * This code snippet prints the previously

built-up matrices A and b like so:



1. Solving the Linear System:
   * Calls TriDiagMatrix\_Solution(size, A, b) to

compute vector x.

* + Displays the temperature at hot end, solution vector x and temperature

at the cold end in order as one total vector consisting of the temperatures along the rod.

Solution to Assignment