PC1

1 b)通用模型，对角线元素取不同值，非对角线元素也取不同值

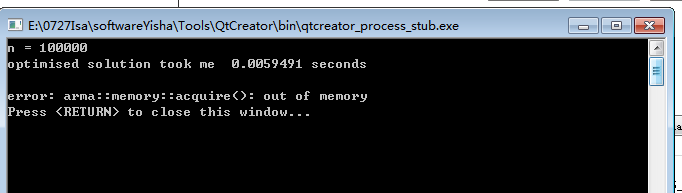
The algorithm for solving this set of equations is rather simple and requires two steps only, a decomposition and forward substitution and finally a backward substitution.

1. Develop a general algorithm, with different elements along the diagonal and the non-diagonal elements.
2. Find also the precise number of floating point operations needed to solve the above equations.
3. specify the values of the array elements *a*, *b* and *c* by inserting their explicit values.
4. select *n*=10, *n*=100 and *n*=1000
5. Compare your results (make plots) with the closed-form solution for the different number of grid points in the interval *x*∈(0,1). The different number of grid points corresponds to different step lengths *h*.

1 c)

Numbers of FLOPS for specific tri-diagonal matrix and CPU time comparison between tri-diagonal matrix and the general algorithm

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| n | | 10 | 10^2 | 10^3 | 10^4 | 10^5 | 10^6 |
| FLOPS for optimized algorithm | | 27 | 297 | 2997 | 29997 | .. | ... |
| Time  (s) | Optimized algorithm  (tri-diagonal matrix) | 5.987e-06 | 1.3256e-05 | 7.2268e-05 | 0.00056138 | / | / |
| general algorithm | 0.00103399 | 0.0021548 | 0.133163 | 13.4013 | / | / |



The number of FLOPS needed in optimized algorithm is given by 3\*(n-1).

When n is big enough, CPU time for optimized algorithm (tri-diagonal matrix algorithm ) is much shorter than general algorithm (LU decomposition). We can see from the table(\*\*\*) that the time of general algorithm becomes larger than the optimized algorithm when n ≥103.

Due to lack of memory, the program could not generate results when n was set larger than 103.

1 d)

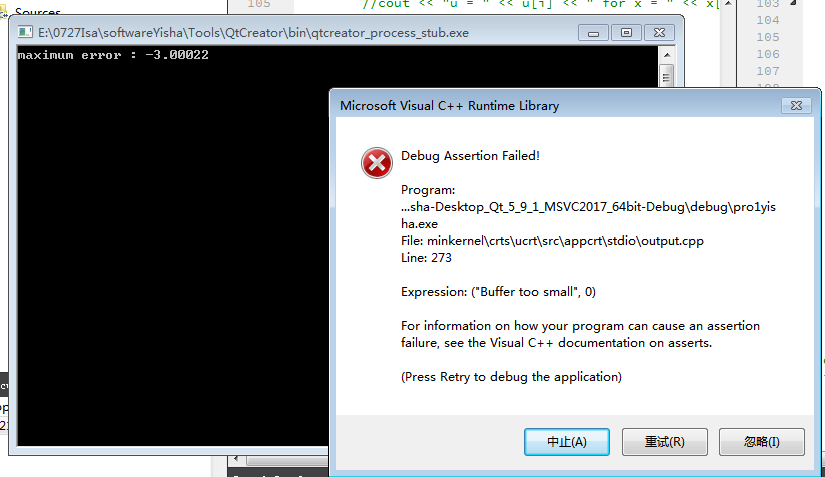
1. Compute the relative error

as function of *log*10(*h*) for the function values *ui* and *vi*. For each step length extract the max value of the relative error. Try to increase *n* to *n*=107. Make a table of the results and comment your results. You can use either the algorithm from b) or c).

Max value of the relative error for different n

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| n | 10 | 100 | 1000 | 104 | 105 | 106 | 107 |
|  | -1.1797 | -3.08804 | -5.08005 | -7.07928 | -8.84297 | -6.07547 | -5.52523 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

The max relative error is reduced with the increasing of n. The program did not make a file of errors when n is 1000 or larger due to assertion error.



2.

Make a table of the results and comment the differences in execution time How many floating point operations does the LU decomposition use to solve the set of linear equations? Can you run the standard LU decomposition for a matrix of the size 105×105? Comment your results.

Out of memory for LU decomposition when the matrix size is

Report requirements

Furthermore, when setting up your git repository for a given numerical project, you should create a folder where selected benchmarks are placed. These benchmarks could represent a calculation with specific input parameters.

should include dynamic memory handling of matrices and vectors.

chapter 6 of the lecture notes, in particular section 6.4 and subsequent sections.

Develop a general algorithm first which does not assume that we have a matrix with the same elements along the diagonal and the non-diagonal elements（通用算法，非相同元素）