Technisch Wetenschappelijke Software
Scientific Software

Homework Assignment 5

C++, Matrix–Vector Multiplication Expression Template

Pieter Appeltans, Emil Loevbak & Wouter Baert Leuven, December 15, 2020

1 Introduction

This assignment focuses on the matrix–vector product

$$(X^T X + \beta I_N)\mathbf{x} \tag{1}$$

with $X \in \mathbb{R}^{N\times N}$, I_N the identidy matrix of dimension N, $\beta > 0$ and $\mathbf{x} \in \mathbb{R}^N$. This expression originates from spectral regression discriminant analysis (SRDA) where one wants to solve the system

$$\mathbf{y} = (X^T X + \beta I)\mathbf{x}$$

for given $\mathbf{y} \in \mathbb{R}^N$. Because $X^T X + \beta I$ is positive definite, we can use the Conjugate Gradients method. As you know from exercise session 8, this requires the evaluation of matrix–vector product (1) for numerous right hand-side vectors. It is thus important that this matrix–vector multiplication is implemented as efficient as possible. You are given 6 different implementations for this product that you have to compare. Figure 1 shows the execution time of these implementations in function of N. To reproduce this figure, you can use the provided `time_xtx.cpp`, which takes 3 command line arguments: the size of the system N, the number of experiments and the number of discarded initial experiments. For example, after compilation, you can run `./time_xtx 100 50 5` with `time_xtx` the name of your executable to perform timings for $N = 100$ over 50 experiments where the first 5 timings are discarded.

The first implementation (`NO-ET` in Figure 1) uses the functor `xtx_op1` in `time_xtx.cpp` and no expression templates. To run this implementation compile `time_xtx.cpp` **without** setting the compile-option `-DEXPR`. The next three implementations use expression templates. To this end, we have provided new expression templates for the transpose of a matrix and for the matrix–vector multiplication. A more detailed description of these expression templates is given below. Using these Expression Templates, four implementations of equation 1 are given in `time_xtx.cpp` in the functors `xtx_op1` (OP1 in the figure), `xtx_op2` (OP2 in the figure), `xtx_op3` (OP3 in the figure) and `xtx_op4` (OP4 in the figure), respectively. You can test these implementations by setting the compile-options `-DEXPR` and respectively `-DOP1`, `-DOP2`, `-DOP3` and `-DOP4`. The final implementation (`OP5` in Figure 1) is specific for equation 1. More specifically, it exploits the fact that

$$\mathbf{y} = \sum_{i=1}^{N} X(i,:)^T \Big(X(i,:)\mathbf{x} \Big) + \beta\mathbf{x} \tag{2}$$

with $X(i,:)$ the ith row of X. It can be tested by **only** setting the compile-option `-DEXPR`. The implementation can be found as a lambda expression in the `main` function of `time_xtx.cpp`.

Important: Use the `-std=c++17` compile-option when compiling `time_xtx.cpp`.

Description of the expression templates

- **Vector sum**, **Vector minus** and **Scalar–Vector Multiplication** are as defined in exercise session 9. Their implementation can be found in `vector_expressions.hpp`.
- **Transpose:** the transpose expression is a matrix expression and translates the `operator()(size_type i, size_type j)` to `A(j,i)` with `A` the original matrix or expression. Its implementation can be found in `matrix_expressions.hpp`.
- **Matrix–Vector multiplication:** the matrix–vector multiplication expression for $y = Ax$ is a vector expression. The `operator()(size_type i)` returns $y(i) = \sum_{j=1}^{N} A(i,j) * x(j)$ with N the number of columns of A. Its implementation can be found in `vector_expressions.hpp`.

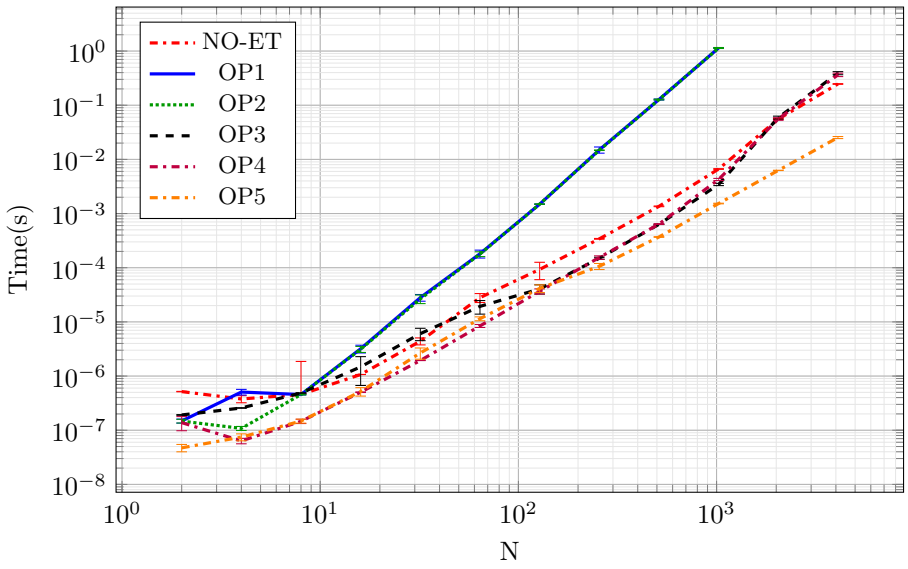

Figure 1: The execution time of the 6 implementations for equation 1 in function of N where the executables are complied using the compile-options `-DNDEBUG`, `-std=c++17` and `-O3`. Results are averaged over 500 runs where the first 10 experiments are discarded.

2 Assignment

1. Explain in detail the differences or similarities between the 6 implementations. Mention in your discussion the (dis)advantages of Expression Templates.

3 Practical information

The deadline for submission on Toledo is **31 December at 14:00**. This deadline is strict! Do not wait until the last minute to submit, as we will not accept technical issues as an excuse for late submissions. Note that the university is closed between Christmas and New Year, and we will therefore not answer our mails.

We expect you to submit a PDF, for which you can use the report template included with this assignment. Please also mention the total amount of time spent on the assignment in your report. Your report should be named hw5_lastname_firstname_studentnumber.pdf, i.e., if your name is John Smith and your student number is r0123456, your file should be called hw5_smith_john_r0123456.pdf.