Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Programming Techniques for Scientific Simulations Exercise 10

HS 15
Prof. M. Troyer

Problem 10.0 Penna simulation with `penna_vector`

Change the underlying container in the Penna simulation to the `penna_vector` provided in the `git` repository in `examples/week11/penna_vector.hpp` and find out the speed up with respect to the version using `std::list` due to the cache effects.

Problem 10.1 Factory design pattern

As we have seen, virtual functions introduce runtime polymorphism. This can be used to create a factory for integrands to be used with our simpson integration.

1. Look at the provided skeleton code `factory_skeleton.cpp` and complete it so that the user can choose the function to be integrated out of the three options: $\sin(\lambda x)$, $\exp(\lambda x)$, $\cos(\lambda x)$.

Problem 10.2 Matrix-matrix multiplication at home

The aim of this exercise is to program your own matrix-matrix multiplication of dense, real, dynamically-allocated matrices.

1. As a first step, implement a simple version in the skeleton code provided.
2. Once you have a correct implementation optimize your code as much as you can by for example exploiting caching effects and vectorization.

Problem 10.3 Matrix-matrix multiplication with libraries

Compare the time required by your own implementation of matrix-matrix multiplication with the same operation performed by a highly optimized library (i.e. BLAS).
To use it from C++, you need to declare DGEMM (`http://www.netlib.org/blas/dgemm.f`) as follows:

```
extern "C" {
  void dgemm_(const char& TRANSA , const char& TRANSB,
  const int& M, const int& N, const int& K,
  const double& alpha ,const double* A, const int& LDA,
  const double* B, const int& LDB,
  const double& beta , double* C, const int& LDC);
}
```

BLAS/LAPACK Installation If you work on your laptop, install BLAS and LAPACK (e.g. ATLAS, which you can obtain from `http://www.netlib.org/atlas/`). If you work on one of the computers in the exercise room, BLAS and LAPACK should already be installed. If you work with Mac OS X, you can either install ATLAS or use the pre-installed `Accelerate` framework by linking with `-framework Accelerate` instead of the standard BLAS/LAPACK linker options. To use the BLAS routines, you have to link your program with (`-lblas`). As the BLAS routines are written in FORTRAN, you might also have to link against the FORTRAN runtime library (`-lgfortran`).