Programming techniques for physical simulations
Exercise 11

November 25, 2009

The aim of this week's exercise is to program a fast library for matrix-matrix multiplication of dense, real, dynamically-allocated matrices. As a first step, implement a simple version in the skeleton code provided. Once you have a correct implementation you add improvements while keeping the interface fixed. Some suggestions for improvements have been made in the lecture - you might want to exploit caching effects and vectorization, but of course we welcome any additional ideas you have!

```cpp
inline void set(double *A, double *B, const int n);

inline void mult(const double *A, const double *B, double *C, const int n);

int main(){

	const int n = 700;
	int n_iterations = 10;

	double *A = new double[n * n];
	double *B = new double[n * n];
	double *C = new double[n * n];

	set(A, B, n);

	for (int i = 0; i < n_iterations; ++i) {
		mult(A, B, C, n);
	}
}

inline void set(double *A, double *B const int n)
{
	//implement!
}

inline void mult(const double *A, const double *B, double *C, const int n)
{
	//implement!
}
```