

The library for matrix multiplication was compiled as part of a research conducted by experts of the '*Basic Projects*' company. The main reason for using the Assembly language was to achieve maximum speed and performance. HPC, or High Performance Computing, and its practical use in complex systems modeling is one of our group's main activities. Our experience in computing task programming goes back many years to the days of the Intel 8087 math co-processor and first IBM PC computers. The proposed new theoretical methods and computer modeling principles provide solutions for tasks such as computer simulation of large systems and complex equipment which were previously considered unsolvable.

For matrix multiplication, 256-bit AVX vector extensions are used and the process is optimized for use on multiprocessor server platforms. The current version of the library is compatible with Microsoft Windows and Intel's Sandy Bridge CPU's. The project is constantly evolving. Currently, we plan to implement the support for new functional extensions (AVX512) and Linux as well as expand computing operations.

The library was developed by the two specialists:

Alexander Parfus. Head of the IT Department. Candidate of Engineering Sciences. Author of computing algorithms and the technology for '*extreme optimization*' of multilevel cache memory performance for efficient matrix multiplication on modern multiprocessor systems. Author of the functionality separation model and interface between upper level and low-level computing libraries.

Ilya Manusov. Chief specialist/hardware support and system programming. Conducted a survey of critical performance factors. Wrote and debugged algorithms for AVX256 vector extensions in the Assembly language. Implemented benchmarks for computing procedures and standard cache memory read/write operations for various cache levels and sizes.

The library uses a single thread but is also optimized for multithreaded processing on multiprocessor systems. The management of parallel running threads is designed to be executed by a top level program.

The library is written in the Intel x64 Assembly language. The DLL files run on 64-bit versions of Microsoft Windows. Flat Assembler (FASM) 1.71.17, FDBG debugger and FASM Editor 2.0 were used. It is recommended to use the latter to ensure that tabulation and syntax highlighting are correct.

The library uses the first generation 256-bit vector extension Intel AVX and is currently optimized for Intel Sandy Bridge CPUs. Near-peak performance was achieved for Sandy Bridge which corresponds to two vector operations of addition and multiplication per core per cycle. Notably, in some cases, the

library exceeded the performance of Intel MKL by a few percent. The results of comprehensive tests will be published in the near future. The performance of the current library version may be less than optimal in other processors with AVX256 support. The library version for the AVX512 vector extension (Intel Xeon, Skylake core) is currently being developed.

The library uses two basic algorithms selected by a top-level program according to the dimensions of the multiplied matrices to ensure maximum performance relative to the task. For a more detailed description, see

[*15-09-27_AVX_256_Specific_Specification_v200.DOC*](#).

[*15-10-13_AVX_OS_DLL_Specification_v202.DOC*](#) describes the configuration libraries for detecting compatible functional extensions, cache size, and other parameters.

Description of the main configuration library files for AVX_OS.

Translated module:

AVX_OS.asm

Function debugging modules:

Debug_DLL_DetectCache.asm

Debug_DLL_DetectVXLib.asm

Connected modules:

DetectCache.inc

DetectVXLib.inc

Description of the main AVX_256_Specific library files.

Translated module:

Specific_AVX1_256_1.asm

Function debugging modules:

Debug_DLL_InitLibrary.asm

Debug_DLL_PackA.asm

Debug_DLL_PackASmall.asm

Debug_DLL_PackB.asm

Debug_DLL_PackBSmall.asm

Debug_DLL_UnPackA.asm

Debug_DLL_UnPackASmall.asm

Debug_DLL_UnPackB.asm

Debug_DLL_UnPackBSmall.asm

Debug_MultiplyMatrixBig.asm

Debug_MultiplyMatrixSmall.asm

Connected modules:

<DIR> MMB

<DIR> MMS

Debug_Context.inc

InitLibrary.inc

PackASmall.inc

PackA_TRANSP.inc

PackB.inc

PackBSmall.inc

UnPackA.inc

UnPackASmall.inc

UnPackB.inc

UnPackBSmall.inc

About us.

The core of the team consists of former Soviet Union Academy of Sciences institutes highly qualified specialists, doctors and candidates of Sciences, whose area of activity: «big systems» simulation. They are representatives of the scientific school of Acad. Georgiy Evgenyevich Pukhov, as well as other specialists.

As an example, let's consider a gas transport systems (GTS) modeling tasks decision. First of all we are talking about the decision of such basic tasks:

- identification of actual modes of operation of hydraulic structures and the parameters of its elements (the solution of a problem of calculus of variations);
- the solution of the optimal control of the operating modes of the GTS.

Actual transient behaviour operating modes of GTS are considered in both cases. GTS may be considered at any level of complexity, up to continental GTS level.

We need to be prepared for the following dimensions of the system with such

tasks: the degrees of freedom (number of variables) may amount to «millions and tens of millions of unknowns»; technological restrictions of equipment may amount to «hundreds of thousands and millions».

The «Curse of dimensionality» makes it impossible to solve such problems of the known mathematical methods.

The term «Curse of dimension» (also known as «Curse of dimensionality difficulties» or «Difficulties»), reflects a rapid growth in the number of required calculations that arise with the increase of object modeling. The considered problems belong to the exponential task complexity.

Until recently, the task of «GTS operating modes optimization» was impossible to calculate. This is right for any «Large systems» optimization tasks. The cause of these problems – automatic transfer to the computer field methods and principles of numerical analysis, developed long before the first computers.

At the heart of our development lays out the **new principles and methods** create accurate numerical models of complex GTS (**and in General complex systems**), which allow modeling systems of many complexity levels. New theoretical methods and principles of computer modeling provide solution of tasks of computer simulation of «systems» and complex equipment, which were previously considered unsolvable.

Ensures the removal of the «Curse of dimensionality», the pressures on the problems of modeling and optimization, «large systems».

Methods and algorithms are developed «high level», in combination with the techniques of «extreme optimization» low-level computational algorithms that allow to solve the problems supercomputer level, using a computing platform for mass application.

Published source computational libraries are one example of such optimization.

Some algorithms for hardware accelerated computing is to be included in this project.

* * *

Contacts:

Head of development Alexander Parfus

E-mail: AVP@basic-projects.com

Chief specialist (hardware) Ilya Manusov

E-mail: IVM@basic-projects.com