

# Intel MKL

Intel Math Kernel Library (Intel MKL) is a library of optimized math routines for science, engineering, and financial applications. Core math functions include BLAS, LAPACK, ScaLAPACK, sparse solvers, fast Fourier transforms, and vector math. The routines in MKL are hand-optimized specifically for Intel processors.

## Documentation

The reference manual for INTEL MKL may be found [here](#).

It includes:

- **BLAS (Basic Linear Algebra Subprograms) and Sparse BLAS Routines** - Sparse Basic Linear Algebra Subprograms (BLAS) perform vector and matrix operations similar to BLAS Level 1, 2, and 3 routines. Sparse BLAS routines take advantage of vector and matrix sparsity: they allow you to store only non-zero elements of vectors and matrices.
  - BLAS Level 1 Routines and Functions (vector-vector operations)
  - BLAS Level 2 Routines (matrix-vector operations)
  - BLAS Level 3 Routines (matrix-matrix operations)
  - Sparse BLAS Level 1 Routines and Functions (vector-vector operations).
  - Sparse BLAS Level 2 and Level 3 (matrix-vector and matrix-matrix operations)
- **LAPACK Routines** - used for solving systems of linear equations and performing a number of related computational tasks. The library includes LAPACK routines for both real and complex data. Routines are supported for systems of equations with the following types of matrices:
  - general
  - banded
  - symmetric or Hermitian positive-definite (both full and packed storage)
  - symmetric or Hermitian positive-definite banded
  - symmetric or Hermitian indefinite (both full and packed storage)
  - symmetric or Hermitian indefinite banded
  - triangular (both full and packed storage)
  - triangular banded
  - tridiagonal.
    - For each of the above matrix types, the library includes routines for performing the following computations:
      - factoring the matrix (except for triangular matrices)
      - equilibrating the matrix
      - solving a system of linear equations

- estimating the condition number of a matrix
- refining the solution of linear equations and computing its error bounds
- inverting the matrix.
- **ScaLAPACK Routines** - Routines are supported for both real and complex dense and band matrices to perform the tasks of solving systems of linear equations, solving linear least-squares problems, eigenvalue and singular value problems, as well as performing a number of related computational tasks. All routines are available in both single precision and double precision.
- **Vector Mathematical Functions**
  - sin
  - tan
  - ...
- **Statistical Functions**
  - RNG
  - Convolution and Correlation
- **Fourier Transform Functions**
  - DFT Functions
  - Cluster DFT Functions - this library was designed to perform Discrete Fourier Transform on a cluster, that is, a group of computers interconnected via a network. Each computer (node) in the cluster has its own memory and processor(s). Data interchanges between the nodes are provided by the network. To organize communication between different processes, the cluster DFT function library uses Message Passing Interface (MPI). Given the number of available MPI implementations (for example, MPICH, Intel® MPI and others), Cluster DFT works with MPI via a message-passing library for linear algebra, called BLACS, to avoid dependence on a specific MPI implementation.

## Benchmarks

These benchmarks are offered to help you make informed decisions about which routines to use in your applications, including performance for each major function domain in Intel® oneAPI Math Kernel Library (oneMKL) by processor family. Some benchmark charts only include absolute performance measurements for specific problem sizes. Others compare previous versions, popular alternate open-source libraries, and other functions for oneMKL [2].

[image-1611056010894.png](#)

[image-1611056010898.png](#)

## Why is Intel MKL faster?

Optimization done for maximum speed. Resource limited optimization – exhaust one or more resource of system [3]:

- **CPU:** Register use, FP units.
- **Cache:** Keep data in cache as long as possible; deal with cache interleaving.
- **TLBs:** Maximally use data on each page.
- **Memory bandwidth:** Minimally access memory.
- **Computer:** Use all the processor cores available using threading.
- **System:** Use all the nodes available.

# Compilation

## Compile with intel/2020

```
#Environment setup
module purge
module load intel/2020
module load intel/2020.mkl
source /cvmfs/sw.el7/intel/2020/mkl/bin/mklvars.sh intel64

icc -mkl <source_file.c> -o <output_binary_name>

./<output_binary_name> #Execute binary
```

## Compile with intel/mvapich2/2.3.3

```
#Environment setup
module purge
module load intel/2020
module load intel/2020.mkl
module load intel/mvapich2/2.3.3
source /cvmfs/sw.el7/intel/2020/mkl/bin/mklvars.sh intel64

mpicc -mkl <source_file.c> -o <output_binary_name>

./<output_binary_name> #Execute binary
```

## Compile with gcc-8.1

```
#Environment setup
module purge
module load gcc-8.1
```

```

module load intel/2020.mkl
source /cvmfs/sw.el7/intel/2020/mkl/bin/mklvars.sh intel64

#Program compile
gcc -L${MKLROOT}/lib/intel64 -WI,--no-as-needed -lmkl_intel_lp64 -lmkl_intel_thread -lmkl_core -liomp5 -
lpthread -lm -ldl <source_file.c> -o <output_binary_name>

#Execute binary
./<output_binary_name>

```

## Performance Test

To test performance, we start by running an example and perform the following calculation:  $C = \alpha * A * B + C$  where A, B and C are matrices of the same dimension.

### WITH MKL

	GCC	MPICC	ICC
n = 2000	0.19 s	0.14 s	0.16 s
n = 20000	51.86 s	50.01 s	49.71 s

### WITH MKL AND MPI

	1 Node	2 Nodes	3 Nodes
MVAPICH2			
MPICH			
INTEL MPI			

## References

- [1] [https://software.intel.com/sites/products/documentation/doclib/mkl\\_sa/11/mkl\\_lapack\\_examples/c\\_bindings.htm](https://software.intel.com/sites/products/documentation/doclib/mkl_sa/11/mkl_lapack_examples/c_bindings.htm)
- [2] <https://software.intel.com/content/www/us/en/develop/tools/oneapi/components/onemkl.html>
- [3] [intel.cn/content/dam/www/public/apac/xa/en/pdfs/ssg/Intel\\_Performance\\_Libraries\\_Intel\\_Math\\_Kernel\\_Library\(MKL\).pdf](https://intel.cn/content/dam/www/public/apac/xa/en/pdfs/ssg/Intel_Performance_Libraries_Intel_Math_Kernel_Library(MKL).pdf)

Revision #6

Created 27 January 2021 10:54:25 by Miguel Viana

Updated 28 January 2021 12:24:34 by Miguel Viana