

Intel[®] oneAPI Math Kernel Library

James Inkster, Varshan Nagarajan, Jordan Sie

Agenda

- Introduction
- Linear Algebra
- Sparse Linear Algebra Functions
- Fast Fourier (FFT)
- Random Number Generators (RNG)
- Data Fitting
- Vector Math
- Summary Statistics
- Code Samples
- Conclusion

Introduction



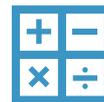
The fastest and most widely used math library for all intel based systems.



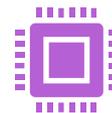
Can reach high performance with its enhanced library of optimized math routines.



It allows for faster math processing times, increased performance and shorter development times.



Linear Algebra, Sparse Linear Algebra, Data Fitting, Fast Fourier Transformations, Random Number Generation, Summary Statistics and Vector Math.

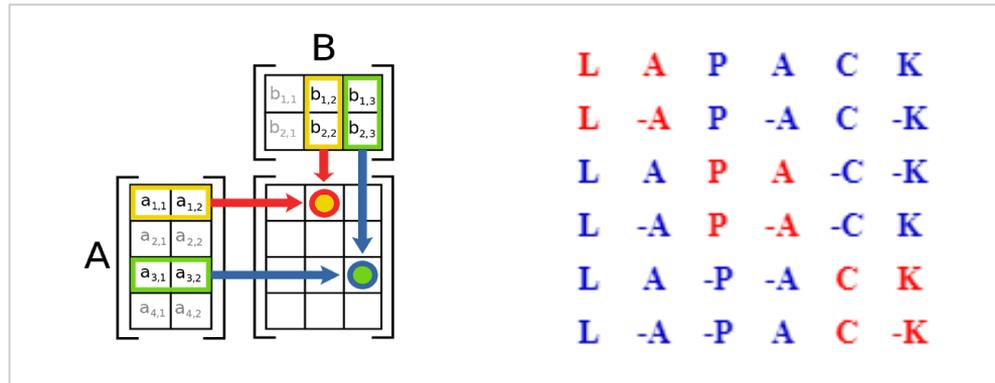


These features will allow you to optimize applications for all current and future intel CPUs and GPUs.

Linear Algebra

Intel's oneAPI MKL library features the ability to speed up linear algebra computations.

- Level 1: Vector-vector Operations
- Level 2: Matrix-vector Operations
- Level 3: Matrix-matrix Operations



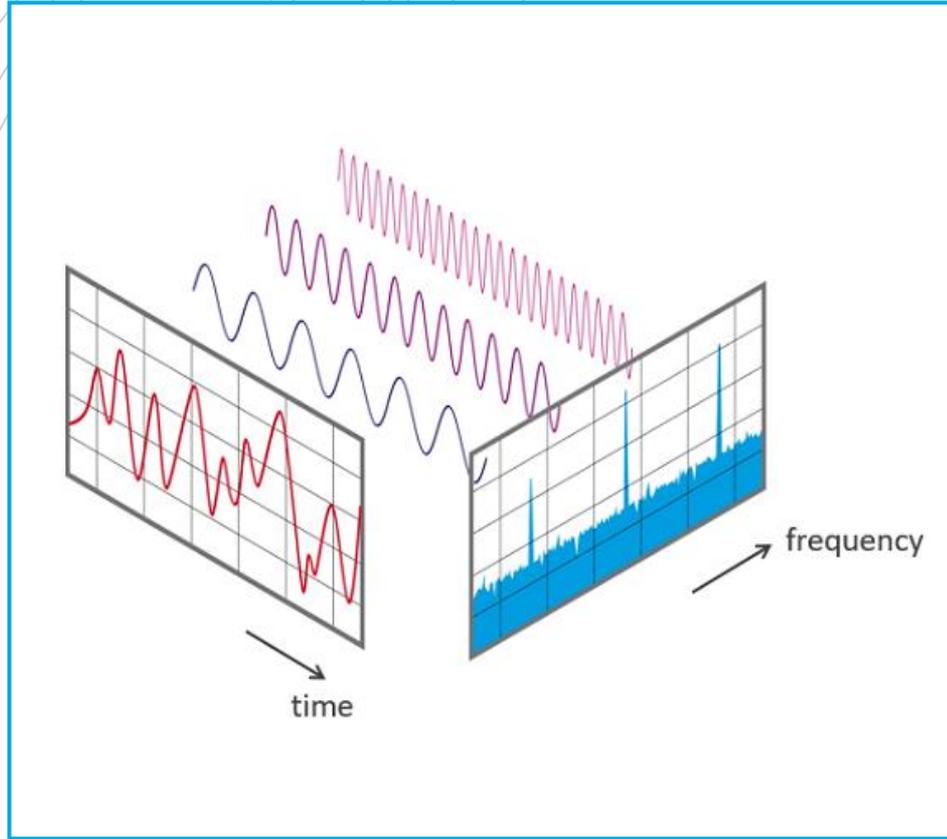


Sparse Linear Algebra Functions

- Able to perform low-level inspector-executor routines on sparse matrices.
 - Multiply sparse matrix with dense vector
 - Multiply sparse matrix with dense matrix
 - Solve linear systems with triangular sparse matrices
 - Solve linear systems with general sparse matrices

Fast Fourier Transforms

- Enabling technology today such as most digital communications, audio compression, image compression, satellite tv, FFT is at the heart of it.
- A **fast Fourier transform (FFT)** is an algorithm that computes the **discrete Fourier transform (DFT)** of a sequence, or its inverse (**IDFT**)



Random Number Generators



- Provides routines that use pseudorandom, quasi-random, and non-deterministic generators.
- developed using the calls to the highly optimized Basic Random Number Generators (BRNGs)
- Many BRNGs differ in speeds and properties

Classes

Engines

- source of randomness
- hold a state of random number generators

```
oneapi::mkl::rng::device::mrg32k3a
```

```
oneapi::mkl::rng::device::philox4x32x10
```

Distributions

- used for transformation of random numbers produced by engines to the appropriate distribution
- hold distribution's parameters

```
oneapi::mkl::rng::device::uniform
```

```
oneapi::mkl::rng::device::gaussian
```

```
oneapi::mkl::rng::device::lognormal
```

```
oneapi::mkl::rng::device::poisson
```

```
...
```

Free functions

Service Routines

responsible for the engine's state modification

```
template <typename Engine>  
oneapi::mkl::rng::device::skip_ahead(Engine& engine, ...)
```

Generation Routines

responsible for obtaining random numbers from a given engine with proper statistics defined by a given distribution

```
template <typename Distr, typename Engine>  
oneapi::mkl::rng::device::generate(Distr& distr, Engine& engine, ...)
```



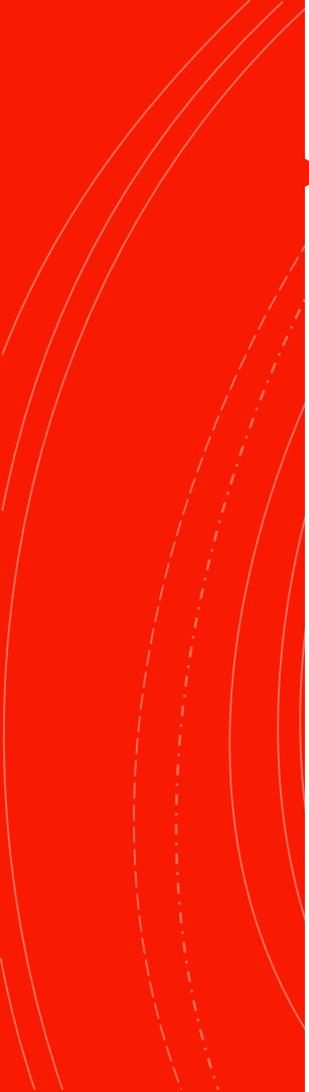
Data Fitting

- Provides capabilities that you can use to approximate functions, derivatives, integrals, and perform cell search operations

- Create a task or multiple tasks.
- Modify the task parameters.
- Perform a Data Fitting computation.
- Destroy the task or tasks.

Data Fitting

- **Task Creation and Initialization Routines** - routines that create a new Data Fitting task descriptor
- **Task Configuration Routines** - routines that set, modify, or query parameters in an existing Data Fitting task
- **Computational Routines** - routines that perform Data Fitting computations
- **Task Destructors** - routines that delete tasks and deallocate resources



Summary Statistics

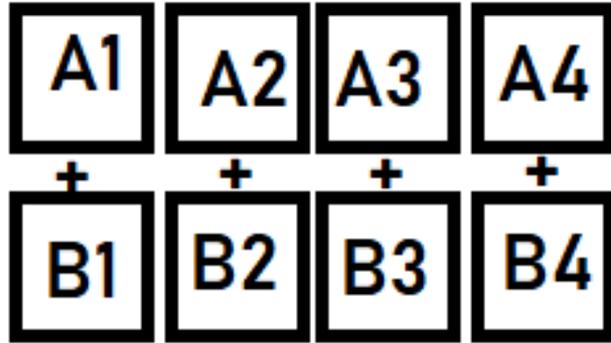
- Compute estimates for single, double and multi-dimensional datasets.
- For example, such parameters may be precision, dimensions of user data, the matrix of the observations, or shapes of data arrays.
- Detect outliers in datasets
- Support missing values in datasets
- Parameterize correlation matrices
- Compute quantiles for streaming data

Vector Math

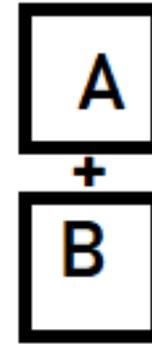
- There are two main set of functions for the Vector Math library that the intel MKL uses they are
 - VM Mathematical Functions
Which allows for it to compute values of mathematical functions e.g. sine, cosine, exponential, or logarithm on vectors that are stored in contiguous memory.
 - VM Service Functions are used for showing when catching errors made in the calculations or accuracy. Such as catching error codes or error messages from improper calculations.



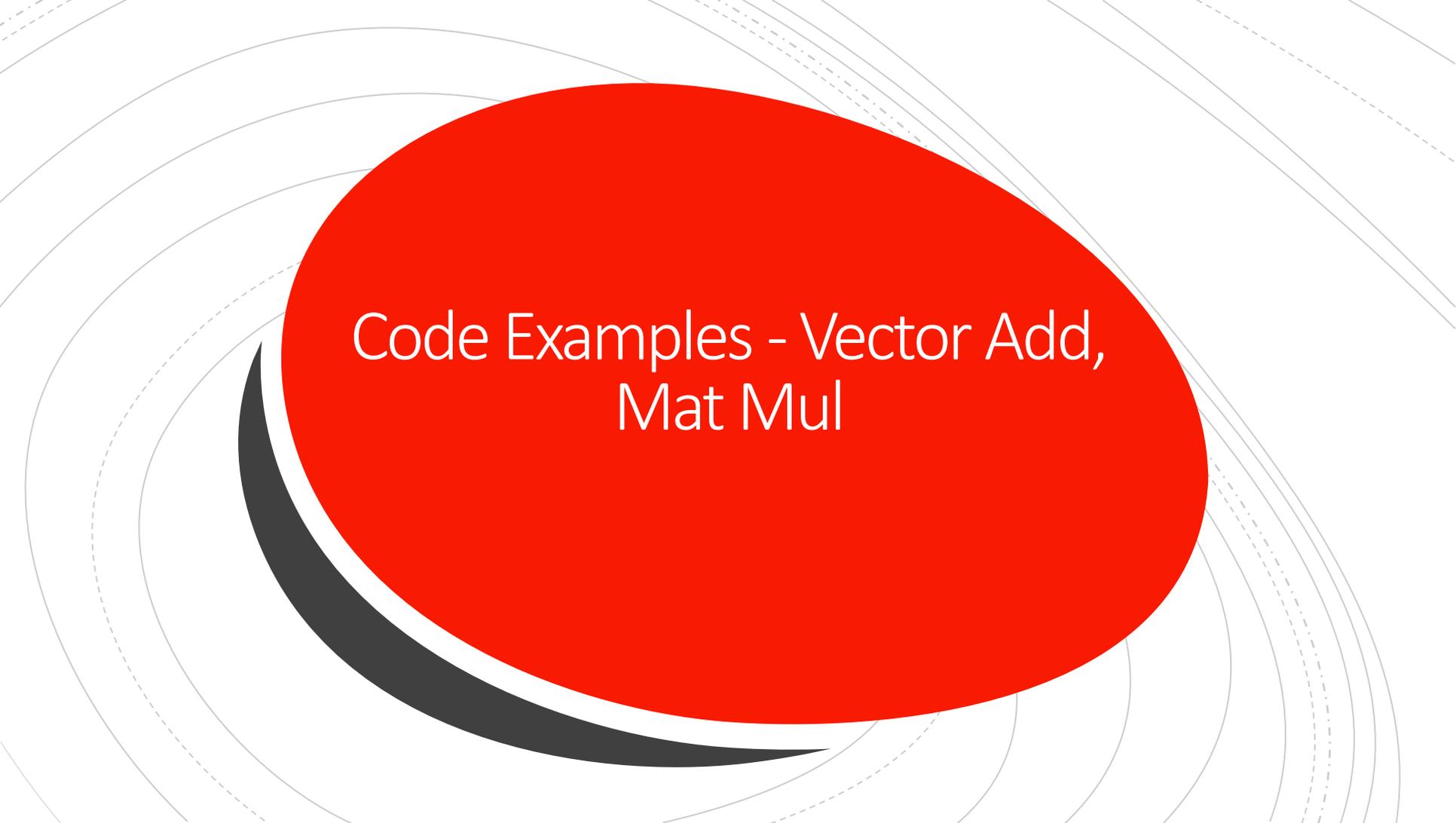
SIMD



SCALAR



Example Vector Math



Code Examples - Vector Add,
Mat Mul

Conclusion

- Available from the oneAPI base toolkit
- Supports programming languages like C, C++, C#, DPC++ and Fortran
- These features will help any financial, science or engineering applications run at an optimized level.
- Constantly updated on the Intel oneAPI website with lots of examples and tutorials available on their github.
- Questions?