Cholesky decomposition using OpenCL

In this example we will create a random 5x5 matrix using uniform distribution and calculate its Cholesky decomposition matrix. The calculation will be done by using a GPU card and OpenCL with a group of MatDeck functions that incorporate ArrayFire functionalities.

First, we will set the environment for the GPU to be used in calculations. Using the function, afp_supported_backends, a list of all supported backends that can be used for calculations will be created. In our case, calculations can be made on the CPU or using OpenCL framework.

$$afp\_supported\_backends() = \begin{bmatrix} \text{"cpu"} \\ \text{"opencl"} \\ \text{"cuda"} \end{bmatrix}$$

The default environment for calculations is CPU, we can change the current environment with the function, afp_set_backend, and check which environment is currently in use with the afp_backend function.

```
afp_set_backend("opencl") = true
afp_backend( ) = "opencl"
```

In each environment, there can be several devices which support calculations within it. To check the number of devices which support calculations in the current environment we use afp_get_device_count, and use the functions afp_get_device and afp_set_device to check/change the current device.

```
afp_get_device_count( ) = 3

afp_get_device( ) = 0
afp_set_device(1) = true
```

To display information about currently selected device use the function, afp_device_info

$$afp\_device\_info() = \begin{bmatrix} \text{"Intel(R)\_HD\_Graphics\_620"} \\ \text{"OpenCL"} \\ \text{"Intel(R) OpenCL"} \\ \text{"2.1"} \end{bmatrix}$$

Finally, we have set OpenCL as a calculation backend and set the device with number 1 - integrated Intel graphic card as a device on which we will do all calculations.

Let's create a uniformly random 5x5 matrix with real values.

```
A := afp_randu(5 , 5 , "real")
```

We can print variable A to check that the input matrix is generated.

$$A = \begin{bmatrix} 0.601 & 0.55 & 0.158 & 0.364 & 0.675 \\ 0.028 & 0.286 & 0.371 & 0.416 & 0.611 \\ 0.981 & 0.341 & 0.354 & 0.581 & 0.523 \\ 0.213 & 0.751 & 0.645 & 0.896 & 0.557 \\ 0.065 & 0.411 & 0.967 & 0.371 & 0.79 \end{bmatrix}$$

Now, we can do Cholesky decomposition calculations on matrix A and place resulting the vector in variable B. The second argument determines if we want to display the upper or lower triangular matrix.

B := afp_cholesky(A , true)

$$B = \begin{bmatrix} 0.775 & 0.709 & 0.204 & 0.469 & 0.871 \\ 0 & -0.216 & 0.371 & 0.416 & 0.611 \\ 0 & 0 & 0.354 & 0.581 & 0.523 \\ 0 & 0 & 0 & 0.896 & 0.557 \\ 0 & 0 & 0 & 0 & 0.79 \end{bmatrix}$$

If we want to display the lower triangular matrix as our resulting matrix, we will use false as the second argument

$$\text{afp\_cholesky}(A , \text{false}) = \begin{bmatrix} 0.775 & 0 & 0 & 0 & 0 \\ 0.036 & 0.534 & 0 & 0 & 0 \\ 1.265 & 0.554 & -1.552 & 0 & 0 \\ 0.274 & 1.388 & 0.645 & 0.896 & 0 \\ 0.084 & 0.763 & 0.967 & 0.371 & 0.79 \end{bmatrix}$$