**Intel® oneMKL – Exercises**

1. To compute product of two square matrices A*B, one might use ?gemm function. However, specialized functions like ?symm or ?hemm allow for computing A*B under certain assumptions on A. Verify if computing A*B, where A is Hermitian is faster using ?hemm than using more general ?gemm functions. If there is a speedup, is it the same for single and double precision complex matrices? How is the speedup related with matrix dimensions?
2. Solve the symmetric eigenvalue problem using appropriately chosen LAPACK driver routine. Then, perform the same operation with computational routines. Is there a performance difference? What do you think are advantages and disadvantages of using both approaches?
3. Solve sparse system of complex linear equations with structurally symmetric coefficient matrix using PARDISO solver.
4. How much faster is matrix-vector multiplication for single precision floating point numbers from the same operation performed on double precision floating point numbers? Write a program that for values of N =100, 1000, 10 000 generates random single and double precision NxN matrices and N-element vector, performs their multiplication and then displays the difference in execution time. Make sure that the inputs to both dgemv and sgemv are the same, i.e. when you generate single precision matrix, make sure its double-precision counterpart has the same contents.
5. Assume there is a 100x100 matrix with 10 nonzero elements and
    a. The matrix is diagonal
    b. All the nonzero elements are in the same row, one next to another
    c. All the nonzero elements are near one corner of the matrix

    Which sparse matrix format would be best for each case?
6. Create a sparse matrix (or pick one from Sparse Matrix Collection), and export it to different formats: COO, CSC and CSR using IE matrix manipulation routines. Compare sizes of all the exported matrices. Is there any of them a clear winner? How does the size depend on the number of nonzero elements?