Tutorial - 14

In this tutorial, we will design and optimize a couple of numerical methods written in OpenACC.

1. Design and implement QR decomposition using modified Gram-Schmidt algorithm given below. This method takes as input a matrix $A[M][N]$, $M > N$, and produces two matrices $Q[M][N]$ and $R[N][N]$ where Q is an orthogonal matrix, and R is an upper triangular matrix such that $A = QR$. Initialize the matrix A with random floating point values. Optimize the OpenACC program for data transfer and parallelism for $M = 1500$ and $N = 1000$.

Algorithm 1: Modified Gram-Schmidt algorithm

function modified Gram-Schmidt(A)
$M, N \leftarrow$ dimensions(A)
$Q \leftarrow$ copy(A)
$R \leftarrow$ zeros(N, N)
for $i = 0, \ldots(N-1)$ do
  $R_{i,i} \leftarrow ||Q_{:,i}||$
  $Q_{:,i} \leftarrow Q_{:,i}/R_{i,i}$
  for $j = (i+1), \ldots(N-1)$ do
    $R_{i,j} \leftarrow Q^T_{:,j} Q_{:,i}$
    $Q_{:,j} \leftarrow Q_{:,j} - R_{i,j} Q_{:,i}$
return Q, R

2. Design, implement, parallelize and optimize a convolution filter in OpenACC. You are given a 1D vector of integers $A[N]$ and a filter $F[K]$, $K << N$ and K is odd. The filter is applied at each element in the vector (respecting the boundaries) which changes $A[i]$ appropriately. Each $A[i]$ is replaced by the summation of the products of the filter elements with neighbors of $A[i]$. Test your code for $K = 5$ and $N = 1000, 2000, 3000$. An example of input and output vectors for a particular filter are given below: