Assignment #1 - Vectorization

In this assignment you will practice with the intrinsics set of SIMD vector instructions.

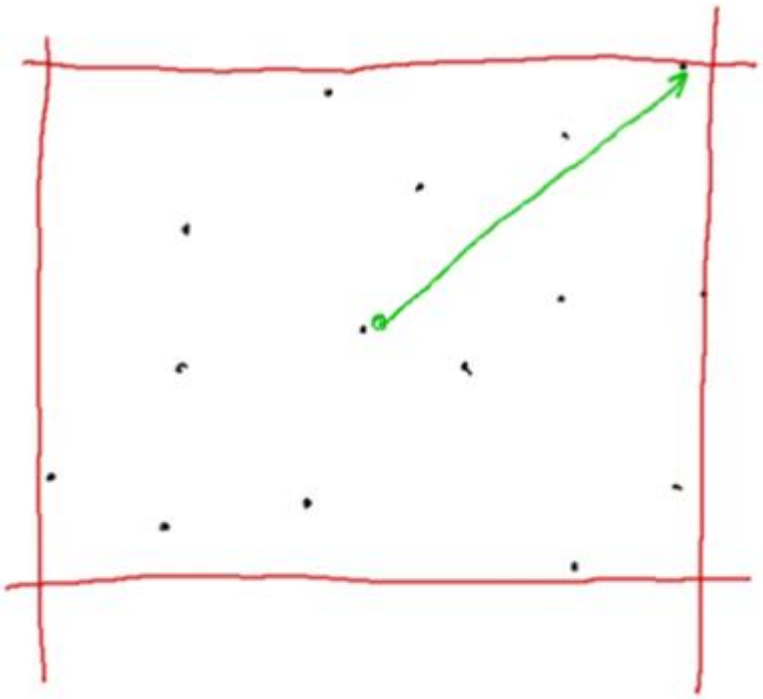
In computer graphics, a 3D mesh is a collection of points representing an object. They can be created algorithmically, manually, or by scanning a real object.

Finding the *smallest enclosing ball* is an interesting problem of computational geometry used, for example, to speedup the detection of colliding objects in a given 3D environment. Each mesh is firstly approximated with a tight ball, then these balls are (somehow) tested to discover those couple that collide.

The problem to solve in this assignment is to implement a solution that make use of the vectorized instructions to find an enclosing disc (not necessarily the smallest) for a given set of random *d*-dimensional points. The procedure is quite simple and consists of 3 main steps:

1. Sweep the set of points to find the minimum and maximum value on each dimension.
2. Use those values to outline a *d*-dimensional box then calculate its center *c*.
3. Sweep again the set of points and find the farthest point *p* from *c:* the radius $r = ||p-c||$.

Finally return the disc of center *c* and radius *r*. Look at the sketch in the figure below for *d*=2:



TASK #1

In lab1.zip Download lab1.zip, you can find a file implementing the sequential version of the algorithm. Your task is to implement the *parallel** version of the struct *mesh* by keeping the same interface without varying the other structs nor the main function.

*Parallelization has to be done by means of the vector extension available on your machine (SSE, AVX, ...) or if you prefer on your GPU (if it support CUDA, in this case we can discuss how to change the rest of the program). **Do not forget** to use a proper data layout and memory alignment allocation when you re-design the data structures defined in the given file.

TASK #2 (optional)

Add a new vectorized function to the struct *mesh* that, given the index *i* of an axis ($0 \leq i < d$), returns the index of the point with maximum coordinate-value on the *i*-th axis.

Once you are done, submit on *Canvas* the .cpp file with the solution of your group.