Physical Model

The objective of this assignment is to implement a simple physical model, which simulates the movements of mass points in a plane. The input is a continuous graph (not necessarily planar) vertices of which are in fact points in 2D plane. The points (vertices) are subjected to a repulsive force (each point is repelled from other points) and the edges represent compulsive bindings (somewhat like springs or rubber bands). The simulation is conducted in quantized time with constant step. Every iteration updates the velocity vector and the position of each point. Please note that the model may be inspired by Newtonian physics, but it does not aspire to be accurate nor true from physical point of view. Furthermore, we are not using any physical units in this description.

The input graph is represented by a trinity (V, E, l), where V is a set of N vertices (stored in a zero-based array – i.e., the indices goes from 0 to N-1) and each vertex has assigned coordinates in Euclidean plane (two double-precision float numbers). E is a set of M edges represented as an array of index pairs (i, j), where the first index is always smaller than the second one $(i < j)$, and the indices are sorted in ascending order (primarily by the first index i). Finally, the l represents an array of lengths assigned to the edges (i.e., it has M values and i-th value is the length of i-th edge from E). Both indices and lengths are 32-bit unsigned integers.

The model has the following parameters, which are all double-precision floats. The value in parentheses is the initial value (and the value which will be used for testing).

- $vertexRepulsion$ (0.1) – the intensity of the vertex repulsion force
- $vertexMass$ (1.0) – the mass (weight) of each point (which influence momentum)
- $edgeCompulsion$ (20.0) – the intensity of the compulsion force of the binding edges
- $slowdown$ (0.995) – a dissipation constant which simulates the resistance of the environment (e.g., friction of the surface) and thus the slowdown of the points in each iteration
- $timeQuantum$ (0.001) – the time iteration step (quantization)

Each vertex maintains an associated velocity vector $\vec{v}$. At the beginning, the whole system is not at motion (all velocities are zero). The simulation performs the following steps in each iteration:

1. A new force vector is computed for each point. The force is a composed vector of all repulsive forces (from all other points) and compulsive forces caused by the edges.
2. The force vectors are applied to modify the velocity vector of each point.
3. All velocities are multiplied by the $slowdown$ value, which simulates the environmental interaction.
4. The points are moved according to their current velocities.

The repulsive forces are based on the behavior of the charged entities, which have the same charge (e.g., protons). According to Coulomb law, the repulsive force is computed as:

$$F = \frac{1}{4\pi\varepsilon_0\varepsilon_r}\frac{|Q_1||Q_2|}{r^2}$$

Since all variables from the previous equation are constant, except for the distance, we have simplified the situation and use an equation $F = vertexRepulsion/r^2$ instead.

The compulsive force, which is defined by the binding edges, is also quadratically proportional to the distance, but it increases as the distance grow. The length of an edge simulates its flexibility and thus the shorter the edge the more intensive force. The final formula is:

$$F = \frac{r^2 \cdot edgeCompulsion}{l}$$

The acceleration can be computed using the second Newton's law:

$$F = am \;\rightarrow\; a = \frac{F}{m} = \frac{F}{vertexMass}$$

Finally, we know that the change of the velocity is the product of acceleration and time and the movement is the product of velocity and time.

$$v = at = \frac{F \cdot timeQuantum}{vertexMass} \qquad s = v \cdot timeQuantum$$

All computations are performed discretely within each time quantum. Please note, that the forces and the velocities are combined as two-dimensional vectors.

Despite the fact that the computations are performed in double precision (64-bit floats), some pathological situations may not be avoided. For instance, if two points are very close, their distance could be rounded to zero (which would lead to division by zero error). To reduce the possibility of such problems, we introduce one precaution. If a square of distances of two points is lower than 0.0001, the value is saturated. In other words, the $r^2 = \max(r^2, 0.0001)$. This rule is applied only for the repulsive force (not the compulsive force).