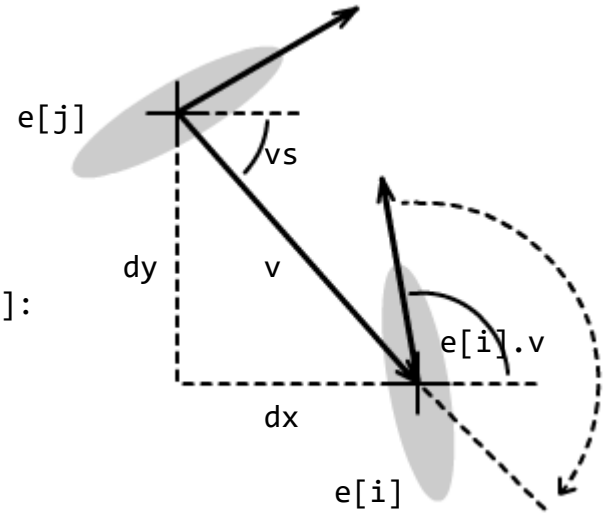
considering e[i]:



The avoidance routine goes like this: Every pair of ellipses, e[i] and e[j], is considered. If the distance between e[i] and e[j] is less than a chosen avoidance radius, the velocity vector of ellipse e[i] is rotated towards the direction of the separation vector v going from e[j] to e[i]. The rotation is proportional to the angle between the vector v and the velocity vector of e[i]. The proportionality factor is a chosen avoidance factor times the quotient of the squared avoidance radius and the squared distance between the ellipses. In this way the rotation changes inversely with the distance. In order to avoid a runaway rotation for small distances, the squared distance has a positive number added. Squared values are used because they are already at hand, thus avoiding the need of a square root:

```
for (i = 0; i < n; i++) {
  for (j = i + 1; j < n, j++) {
    dx = e[i].x - e[j].x;
    dy = e[i].y - e[j].y;
    d2 = dx * dx + dy * dy;
    if (d2 < r2) {
      vs = Math.atan2(dy, dx);
      d2 = d2 + a;
      e[i].v -= f * r2 * (e[i].v - vs) / d2;
      e[j].v -= f * r2 * (e[j].v - (vs + pi)) / d2;
    }
  }
}
```

where:

```
i, j   : counting variables
n      : number of ellipses
e[i].x : abscisse of ellipse number i
dx     : horizontal distance between ellipses i and j
dy     : vertical do.
d2     : squared distance between ellipses i and j
r2     : squared avoidance radius, 10000 [pix²]
vs     : angle of the separation vector from e[j] to e[i]
a      : a = 200 [pix²], in order to avoid r2/d2 -> ∞ for small distances
e[i].v : velocity angle of ellipse number i
f      : avoidance factor, 0.01
pi     : pi, used to get the angle of the separation vector from e[i] to e[j]
```

2023-01-05 B.J.