**TP ALPHA SHAPES**

**Conseils et liste des fonctions à utiliser**

Note that a Facet f is a std::pair<Cell\_handle, int>

to access the pair elements:

Cell\_handle c = f.first;

int index = f.second;

To know whether a cell is infinite

Dt::is\_infinite(Cell\_handle c): return true if cell is infinite

To get neighbor of cell:

Cell\_handle n = c->neighbor(index); // return neighboring cell of c, facing vertex number i

To access vertex i of Cell\_handle:

Vertex\_handle vi = c->vertex(i)

To access the point from a vertex handle:

const Point& p = v->point()

Note that the cells are oriented consistently, ie to recover the proper

orientation of the facet points:

// assume index = index of facet in cell c

if ( index % 2 == 1 ) // odd index

{

p1 = c->vertex((index + 1) % 4)->point();

p2 = c->vertex((index + 2) % 4)->point();

p3 = c->vertex((index + 3) % 4)->point();

}

else

{

p1 = c->vertex((index + 2) % 4)->point();

p2 = c->vertex((index + 1) % 4)->point();

p3 = c->vertex((index + 3) % 4)->point();

}

To compute the circumsphere center of a cell (of its 4 points)

const Point oc = CGAL::circumcenter(p0, p1, p2, p3);

To compute the circumsphere center of a facet (of its 3 points)

const Point oc = CGAL::circumcenter(p0, p1, p2);

To compute orientation of a point oc with respect to oriented plane

if ( CGAL::orientation( p0,p1,p2, oc ) == CGAL::POSITIVE ) ...

or

if ( CGAL::orientation( p0,p1,p2, oc ) == CGAL::NEGATIVE ) ...

To compute the squared distance between two points:

const double sqd = CGAL::squared\_distance(p, q);

To compute the distance between two points:

const double d = std::sqrt(CGAL::squared\_distance(p, q));

To compute max and min:

const double v = CGAL::max(double, double);

const double v = CGAL::min(double, double);