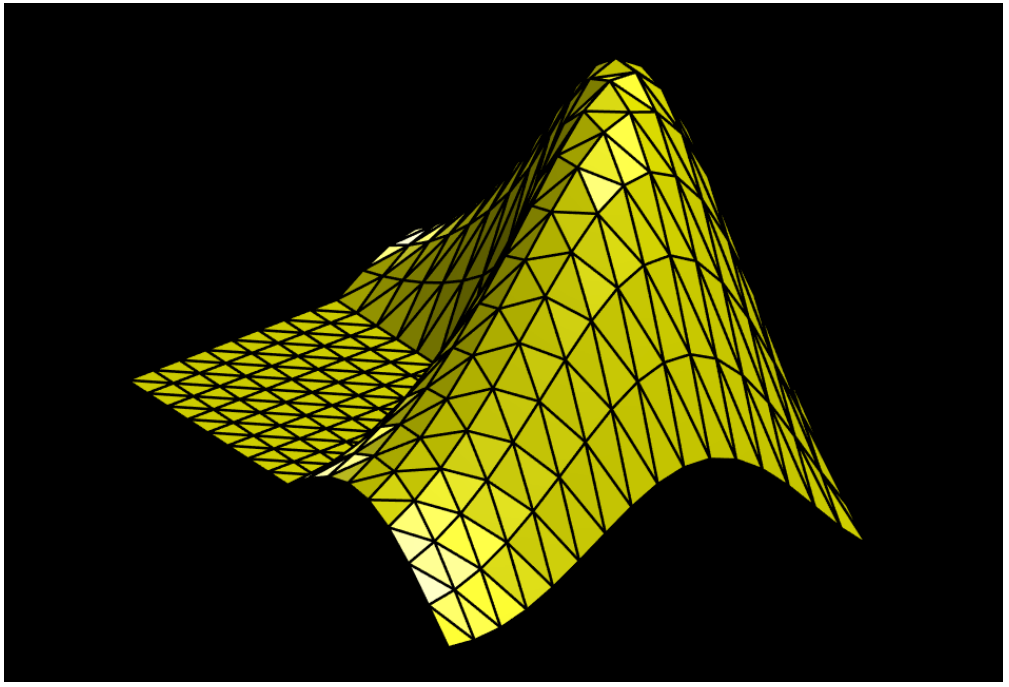
Vectorized mesh processing toolbox in Matlab



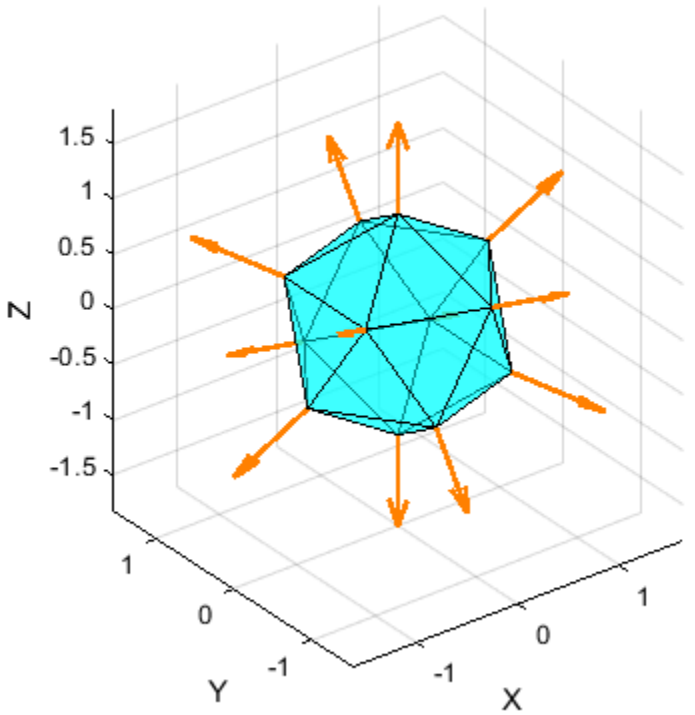
MATLAB

Nicolas Douillet 2020

I Vertex normals

Principle : mean of face normals in a given neighborhood

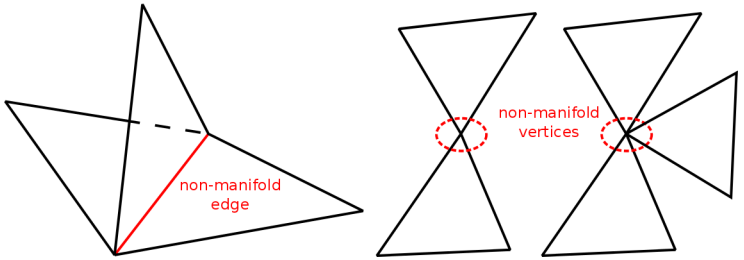
```
addpath('data/');
addpath('src/');

load('icosahedron.mat');
ngb_degre = 1;
select_vertex_normals(V,T,ngb_degre)
;
```



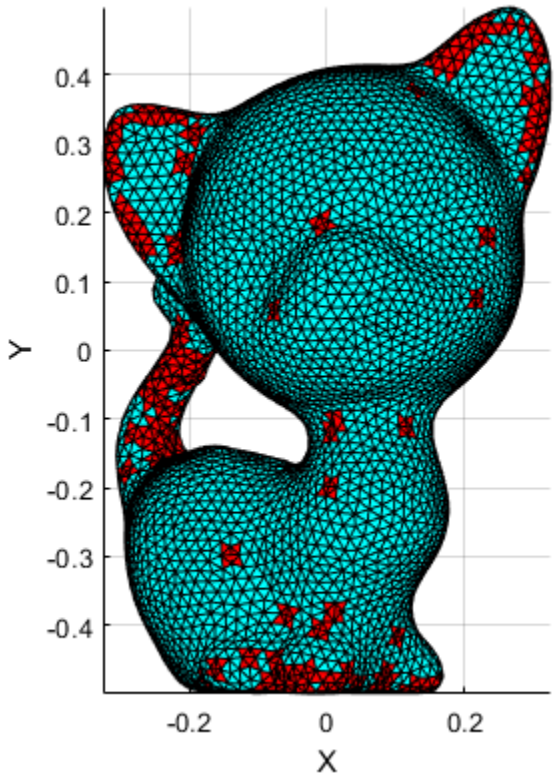
II Compute and select non manifold triangles

Principle : a non manifold edge is shared

between 3 or more triangles.



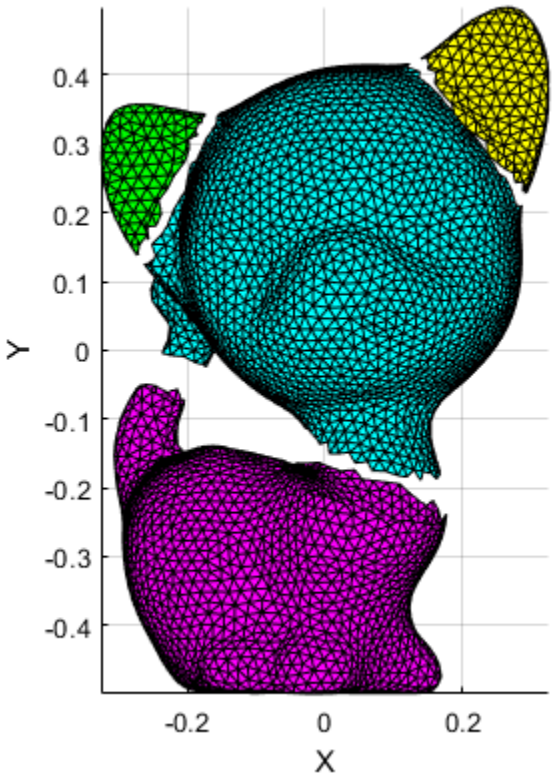

```
load('kitten_nmnfld.mat');
nmnfld_tgl_idx_list =
select_non_manifold_triangles(V,T);
view(0,90);
```



III Compute and select connected components

Principle : triangle neighbor search ‘glutton’ algorithm.

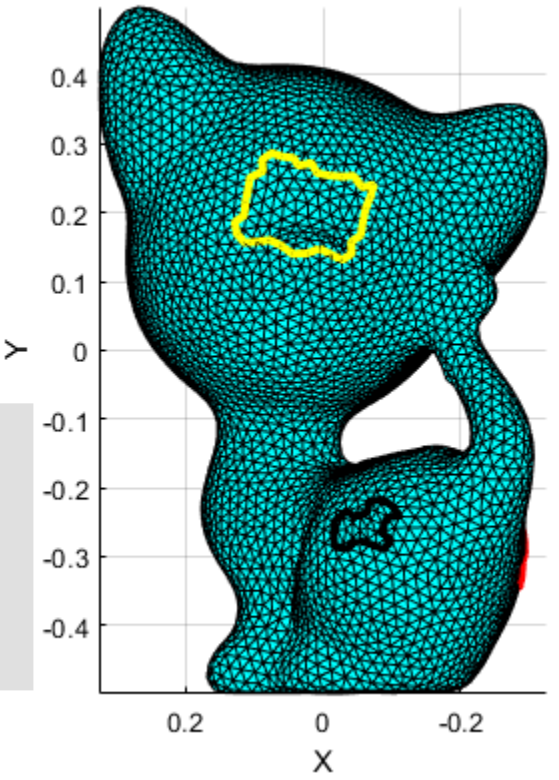
```
load('kitten_components.mat');
[cc_nb,components] =
segment_connected_components(T);
show_mesh_components(V,components);
view(0,90);
```



IV Compute and select holes and boundary

Principle : detect and reorder non shared edges.

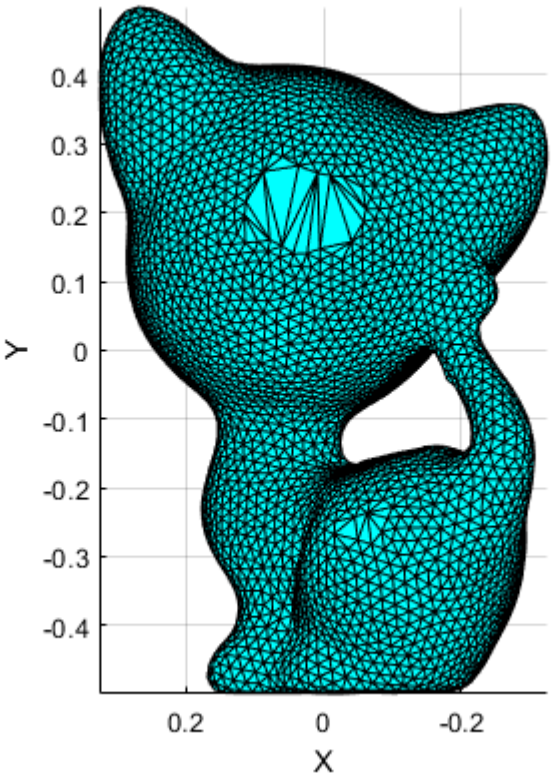
```
load('kitten_holed.mat');
nmnfld_vtx_idx = select_non_manifold_vertices(V,T,false);
[V,T] = clone_solve_nmnfld_vertices(V,T,nmnfld_vtx_idx);
boundaries = select_holes_and_boundary(V,T);
view(0,90);
```



V Fill mesh holes

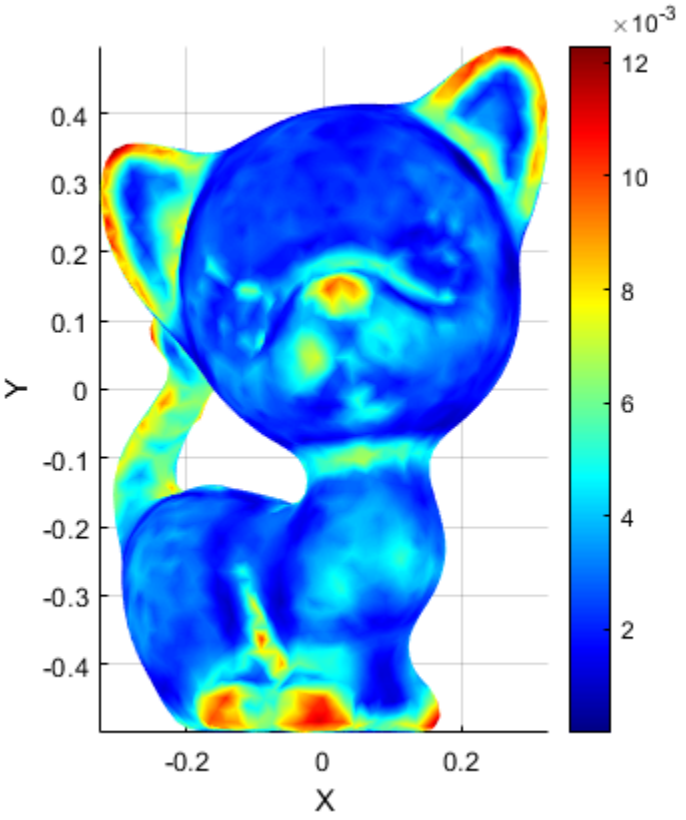
Principle : without data / point addition.
prior sharp angles and
curvature.

```
[V,T] = remove_non_manifold_vertices(V,T);
boundaries = select_holes_and_boundary(V,T);
view(-180,-90);

Perimx = 200; % max perimeter length
T = fill_mesh_holes(V,T,boundaries,'closed',perimx);
plot_mesh(V,T);
view(-180,-90);
```



VI Compute and show curvature

```
load('kitten.mat');
ngb_degre = 2;
N = compute_vertex_normals(V,T,ngb_degre,'raw');
Curvature =
compute_mesh_curvature(V,T,N,ngb_degre,'mean');
show_mesh_curvature(V,T,curvature);
view(0,90);
```

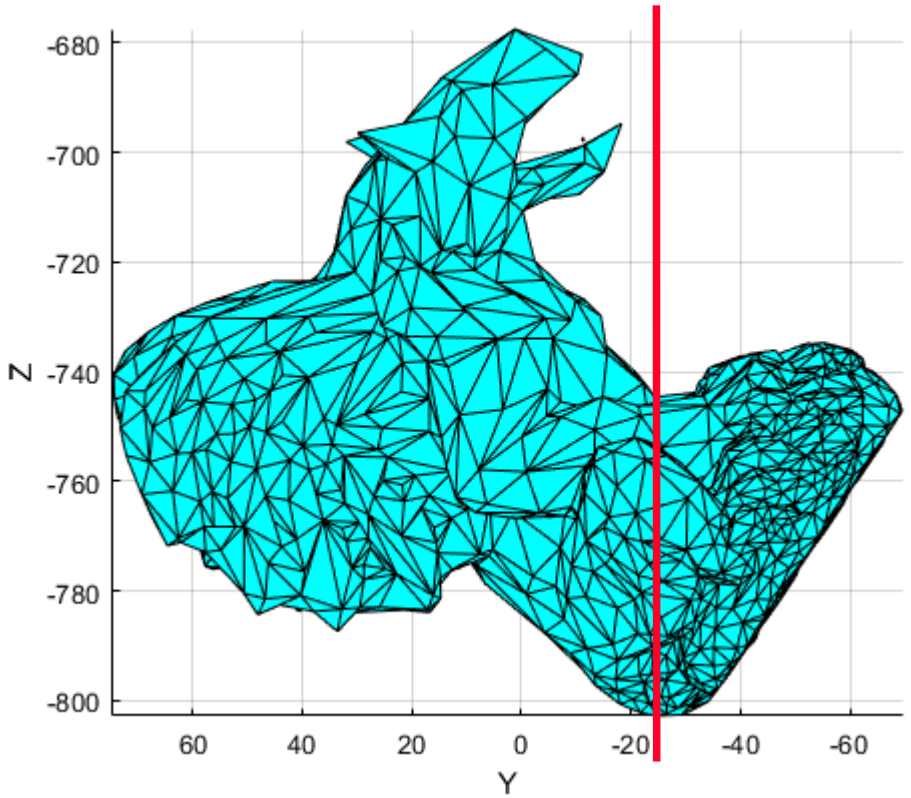


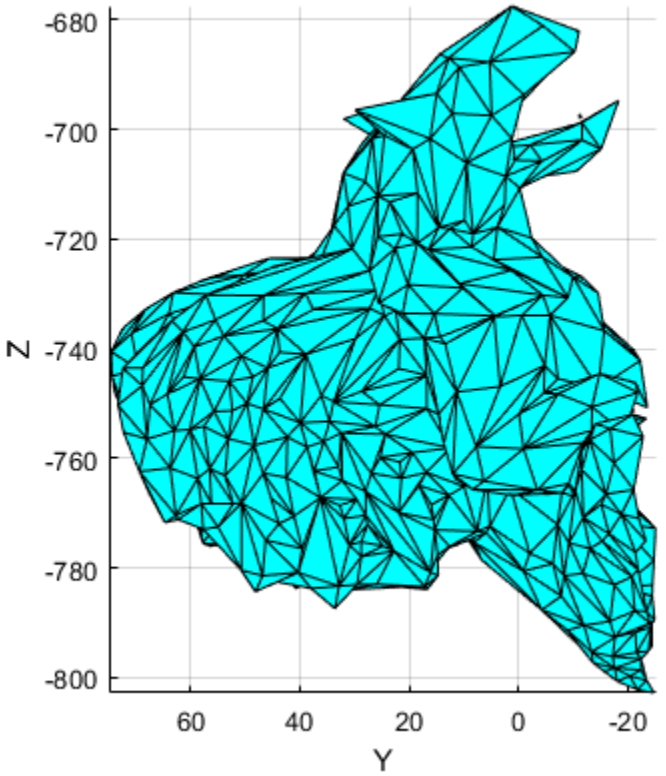
VII Mesh subselection : principle

Principle : localize vertices in the right part of the half space. Deduce edges and triangles belongings or not to submesh.

```
load('Gargoyle_3k.mat');
plot_mesh(V,T);
view(-90,0);

N = [0 1 0]; % plane normal vector
I = [0 -25 800]; % one point belonging to the plane
[V_out,T_out] = submesh_selection(V,T,n,I); % Gargoyle top part

plot_mesh(V_out,T_out);
view(-90,0);
```

VII Mesh subselection : results



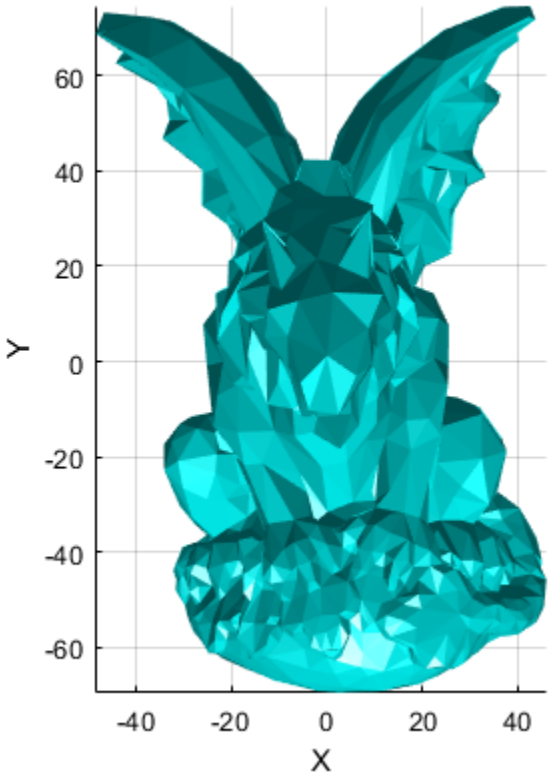


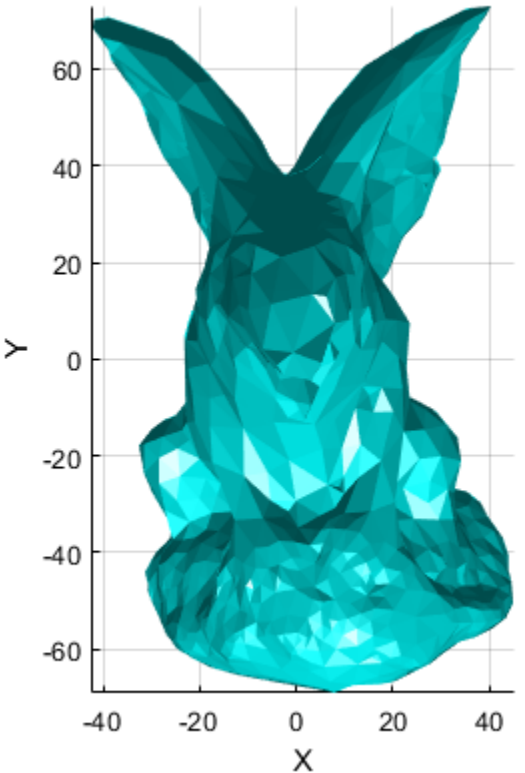
VIII Mesh smoothing : principle

Principle : vertex is replaced by the mean of its local neighbors.

```
plot_mesh(V,T), shading interp, camlight right;
view(0,90);

N = compute_vertex_normals(V,T,2);
nb_iterations = 1;
ngb_degre = 1;
V = mesh_smooth(V,T,nb_iterations,ngb_degre);

plot_mesh(V,T), shading interp, camlight right;
view(0,90);
```

VIII Mesh smoothing : results





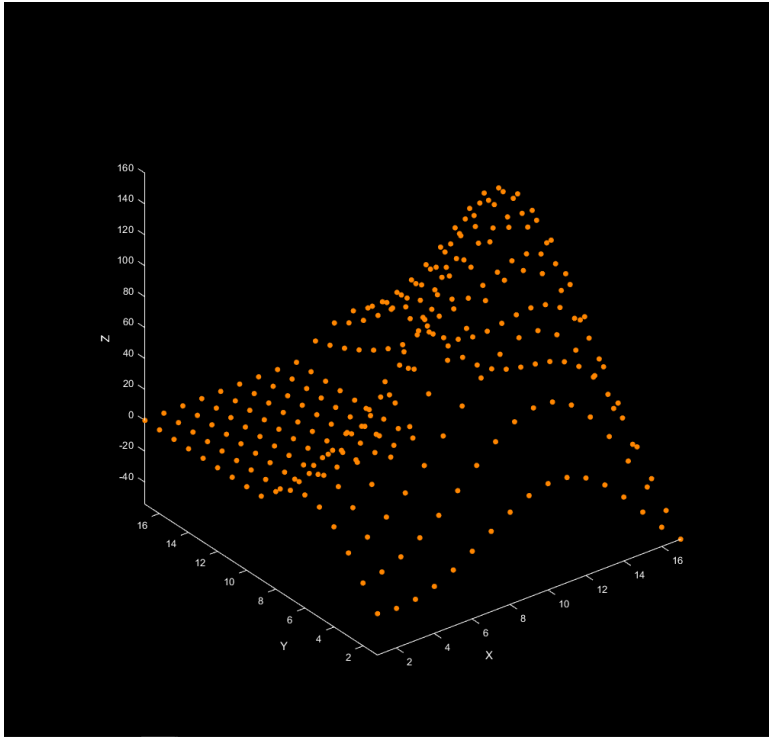
IX Convex hulls : principle

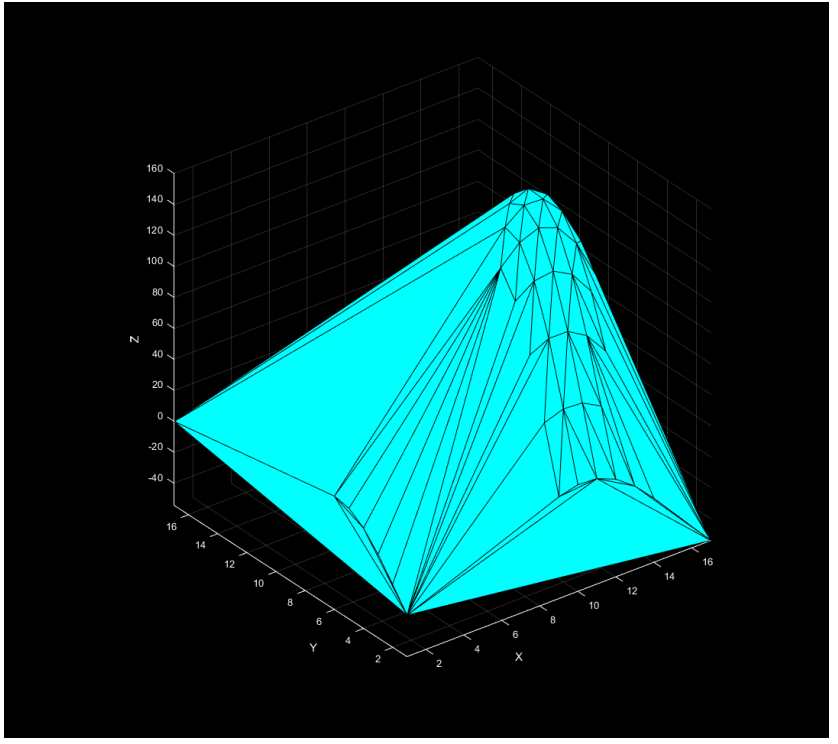
```
nb_vtx = 128;
X = 2*(rand(nb_vtx,1)-0.5);
Y = 2*(rand(nb_vtx,1)-0.5);
Z = 2*(rand(nb_vtx,1)-0.5);

Rho = X.^2 + Y.^2 + Z.^2;
i = Rho <= 1; % keep points inside the unit sphere
X = X(i);
Y = Y(i);
Z = Z(i);
V = cat(2,X,Y,Z);

plot_point_set(V,'o','y',4);
axis equal, view(3);

[V,Qh] = quick_hull(V);
plot_mesh(V,Qh);
axis equal, view(3);
```

Principle :

Divide and conquer/quick hull algorithm with inside points removal or gift wrapping/Jarvis algorithm.

IX Convex hulls : results





Sources & test functions : many more

Functions list

deprecated

- find_edges_from_vertex
- find_triangles_from_vertex
- find_triangle_indices_from_edge
- quadrangle
- query_one_vertex_neighbor_indices
- triangle

src

- add_triangles
- add_vertices
- clone_solve_nmnfld_vertices → flat triangles debug ?
- collect_intersecting_triangle_indices → vectorize + ne pas enlever les T-vertices
- compute_component_areas
- compute_face_normals
- compute_face_selection_area
- compute_hole_and_boundary_perimeters
- compute_mesh_bounding_boxe
- compute_mesh_curvature : + gaussian, + normal, + RMS (discrete)
- compute_mesh_total_area
- compute_mesh_volume
- compute_point_to_mesh_min_distance
- compute_vertex_normals
- detect_mesh_holes_and_boundary : input = component set ; loop process on each input
- erode_mesh_boundary_and_holes : identify contour (hole / boundary)

- fill_mesh_holes : + add constraints avoid flat triangles, curvature continuity, + en priorité les triangles de produits scalaire proches de 0 en val abs dans les plans « orthogonaux » à celui du contour (ou de produit vectoriel max),

- find_edges_from_vertex_list
- find_neighbor_triangle_indices
- find_triangle_indices_from_edges_list → à trier par 2, par edge ?
- find_triangle_sets_from_vertex_list (indices separated in a cell array)
- find_triangles_from_vertex_list (all indices together)
- flip_faces_orientation
- ismesh2Dmanifold
- ismeshwatertight
- invert_face_orientations
- line_plane_intersection
- mesh_smooth
- mesh_union
- plot_mesh
- plot_point_set
- point_to_line_distance
- point_to_plane_distance
- quad2trimesh
- query edges list
- query_every_vertices_neighbor_indices
- rebuild_triangulation_from_edge_list
- remove_duplicated_triangles
- remove_duplicated_vertices
- remove_flat_triangles
- remove_isolated_triangles
- remove non manifold triangles
- remove non manifold vertices
- remove_self_intersecting_triangles
- remove_small_components
- remove_triangles
- remove_unreferenced_vertices
- remove_vertices
- reorient_all_faces_coherently
- segment_connected_components
- select_face_normals
- select_holes_and_boundary
- select_non_manifold_triangles
- select_non_manifold_vertices
- select_self_intersecting_triangles
- select_vertex_normals
- show_holes_and_boundary : vectorize ?
- show_mesh_components
- show_mesh_curvature
- show_triangle_selection
- show_vertex_selection
- submesh_selection

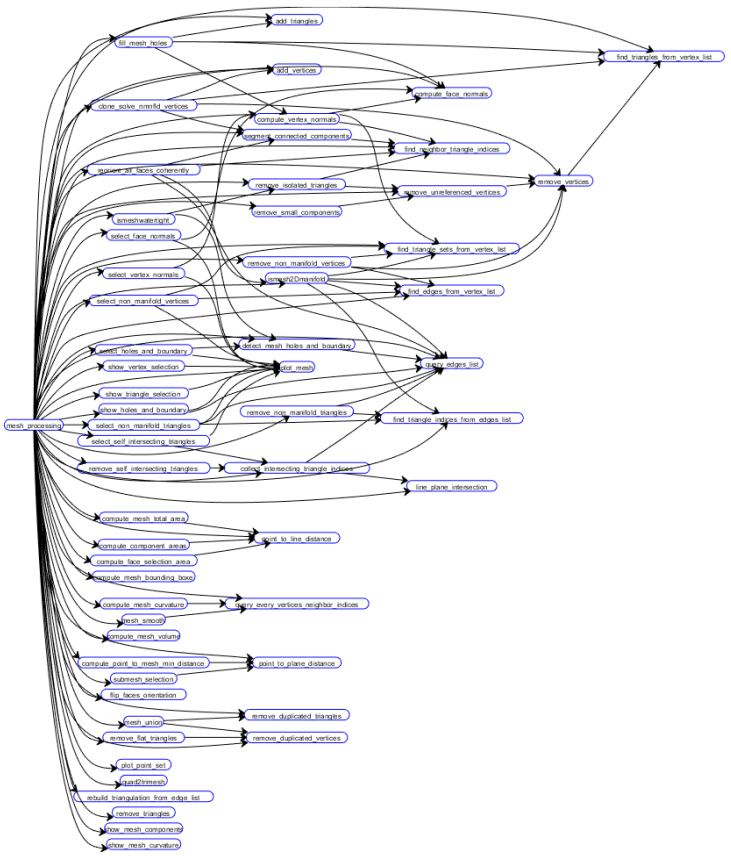
tests

x test_erode_holes_and_boundary
x test_clone_solve_nmnfld_vertices
x test_compute_component_areas
x test_compute_mesh_curvature
x test_compute_hole_and_boundary_perimeters
x test_compute_mesh_volume
x test compute_point_to_mesh_min_distance
x test_compute_total_area
x test_compute_vertex_normals
x test_fill_mesh_holes
x test_ismesh2D manifold
x test_ismeshwatertight
x test_mesh_smooth
x test_mesh_union
x test_quad2trimesh
x test_query_every_vertices_neighbor_indices
x test_remove_isl_triangles
x test_remove_non_manifold_triangles
x test_remove_non_manifold_vertices
x test_remove_self_intersect_tgl
x test_remove_small_components
x test_reorient_all_faces_coherently
x test_show_components
x test_show_holes_and_boundary
x test_segment_connected_components
x test_select_holes_and_boundary
x test submesh_selection

data

- Archimede_spiral.mat
- Armadillo.mat
- Armadillo_20k.mat
- brain.mat
- bunny_16k.mat
- bunny_69k.mat
- cross_filled_cube.mat
- cross_filled_octahedron.mat
- cube.mat
- cube_quad_meshed.mat
- dodecahedron.mat
- filled_cube.mat
- half_filled_octahedron.mat
- filled_octahedron.mat
- Gargoyle_1.7M.mat
- Gargoyle_3k.mat
- Gargoyle_5k.mat
- geoid_lvl_16.mat
- icosahedron.mat
- kitten.mat
- kitten_holed.mat
- kitten_nmnfld.mat
- log_spiral.mat
- meshed_mtlb_logo.mat
- randoriented_ico.mat
- octahedron.mat
- singularity.mat
- sinico.mat
- tetrahedron.mat
- torus.mat
- torus_light.mat

Functions dependancy graph



Documentation

Each file integrates its own header help / documentation :

```
%% detect_mesh_holes_and_boundary : function to detect vertices which are part of
% the mesh boundary and list their indices in boundary vectors.
%
% Author & support : nicolas.douillet (at) free.fr, 2020.
%
% From the vertex and triangle lists, this function computes
% the mesh boundaries when there are some (opened surface
% or presence of holes in the mesh).
%
% Principle is based on detecting and sorting non shared edges.
%
%
% Input arguments
%
%
%       [ |  |  | ]
% - T = [i1 i2 i3 ], positive integer matrix double, the triangulation, size(T) = [nb_triangles,3].
%       [ |  |  | ]
%
%
% Output argument
%
% - boundaries : cell array of positive integer row vectors double, size(boundaries) = [nb_holes,1].
```

On going and upcoming developments (1)

- Rebuild triangulation from disordered edge list **

- Mesh_simplify with quadric_edge_collapse *****

- Mesh_slice **** ✓

- Improve curvature computation ****

- Mesh_split_&_stich ****

- Mesh smooth : + cotan option ***

- Improved fill holes + select the hole by its id ***

On going and up coming developments (2)

- Mesh refinement ✓

- Merge_close_vertices

- remove_unacceptable T_vertices

- speed up algorithms with hybrid for loop / vectorized ; enable // computing (parfor)

- Mesh segmentation

- Intersection of two meshes

Inspirations : Meshlab  CGAL

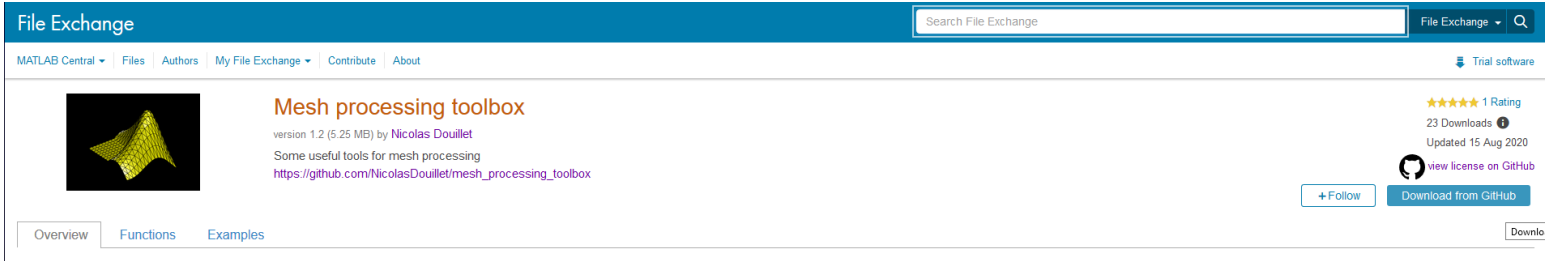
Links to download

Github :

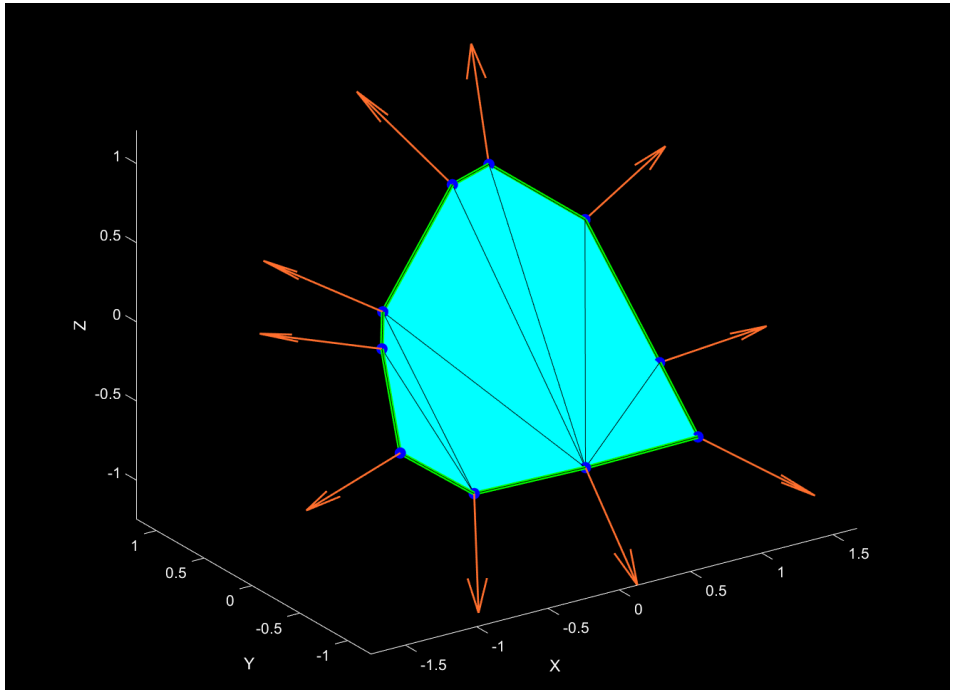
https://github.com/NicolasDouillet/mesh_processing_toolbox

Mathworks file exchange :

https://fr.mathworks.com/matlabcentral/fileexchange/77004-mesh-processing-toolbox?s_tid=prof_contriblnk



More to come : mesh generation toolbox



Discrete contour mesh patch ✓

- Same principle as fill_mesh_holes
- No triangle context
- Automatic contour points reordering
- 3D or 2D

Multiresolution mesh

From the quick hull divide & conquer algorithm principle. Reverse way for concavities.