

Notes on Smooth Surfaces

June 19, 2018

1 Comments on the code and calling sequences

The current version of the code includes:

- Refinement Algorithm
- Adaptive Sigma
- Manas' FMM call for each Newton call
- Fast tree code for smooth sigma evaluation (nadadena.f90).
- Quadratic triangulated skeletons

The calling sequence is:

```
>> gfortran -o Smooth_Surface_Quadratic_swap_FMM_v4 -O3 nadadena.f90 ..  
.. Smooth_Surface_Quadratic_swap_FMM_v4.f90 tfmm3dlib.a  
>>./Smooth_Surface_Quadratic_swap_FMM_v4
```

The refinement algorithm is activated using the variable $n_{refinement}$ inside the file Smooth_Surface_Quadratic_swap_FMM_v4.f90. If $n_{refinement} = 0$ then there is no refinement, the number of triangles of the resulting smooth surface equals the number of triangles of the skeleton. If $n_{refinement} = 1$ then each smooth triangle is divided by 4. if $n_{refinement} = 2$ then each smooth triangle is divided by 4^2 , and so on..

The adaptive sigma for small details is activated using the variable adaptive_flag. If adaptive_flag=0 then there is no adaptivity, if adaptive_flag=1 then there is adaptivity. In general it could be interesting to disable the adaptivity only if the geometry is given by a very small number of triangles, to avoid some distortions

The FMM call is used on each Newton iteration using $iprec = 3$.

The fast tree call to evaluate the variable sigma and gradient of sigma is used on each Newton iteration before each FMM call.

There are several variables at the beginning of the program to establish several options:

n_order_sk:

Number of nodes per skeleton. Used as sources in the FMM code, usually 78 and not changed.

n_order_sf:

Number of nodes on each smooth triangle. You can use 45 or 78

n_refinement

n_refinement=0 if no refinement, n_refinement=1 to split each smooth triangle into 4, n_refinement=2 to split each smooth triangle into 4^2 , and so on.. the skeleton (the number of sources in the fmm call is not modified in any case)

adaptive_flag

This is to enable adaptivity, if adaptive_flag=0 then sigma is constant and has an appropriate value commensured with the size of the triangle. If adaptive_flag=0 then sigma is a smooth function proportional to the size of each triangle of the skeleton

nombre:

variable containing the name of the .msh file from GID as input

filename:

variable containing the name of the .gov file of the output file that will contain the smooth surface.

x_0, y_0, z_0

coordinates of a point that should be inside the resulting smooth surface to test the Gauss integral at the end of the process and estimate the accuracy of the smooth surface.

2 How to generate a valid .msh file with GID

Next I describe the procedure to make a .msh file with GID compatible with this code:

First create your model (with consistent normals pointing towards the exterior region), then defile the elements of the mesh as Quadratic (see 1).

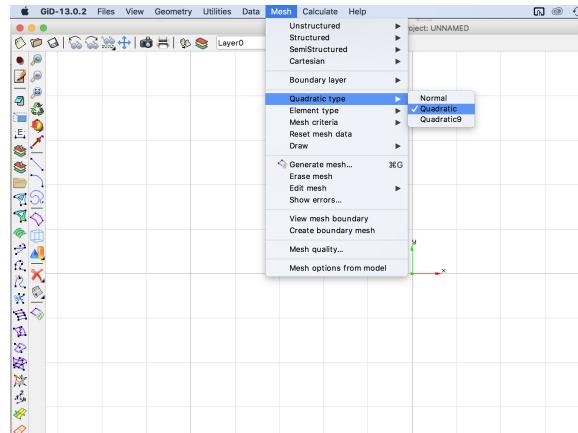


Figure 1:

Next define the element type as triangles (see 2)

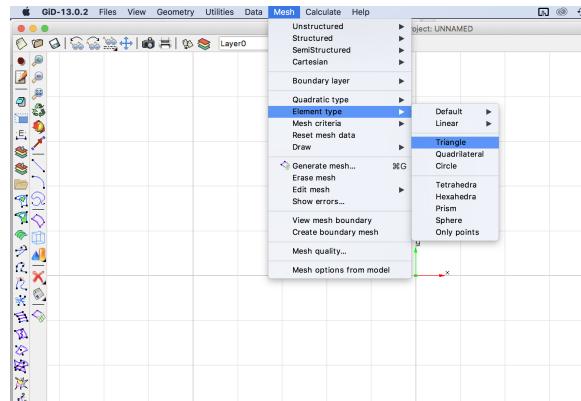


Figure 2:

and select all the surfaces on the geometry.

Next generate the mesh (see figure 3) deciding the maximum size of the triangular elements (if there are small details in the geometry, it will create smaller triangles in that region avoiding high differences in size of adjacent triangles)

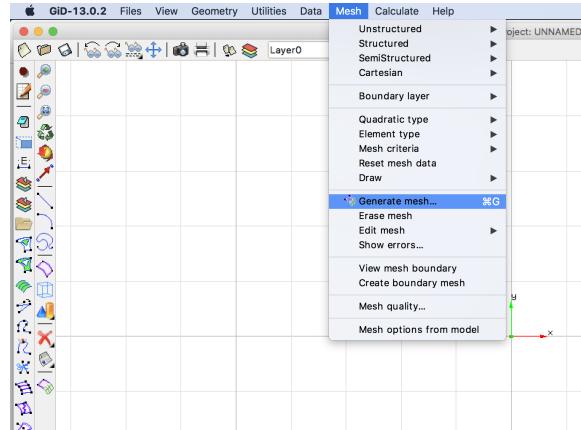


Figure 3:

Finally save the model using export>ASCII project.. (see 4).

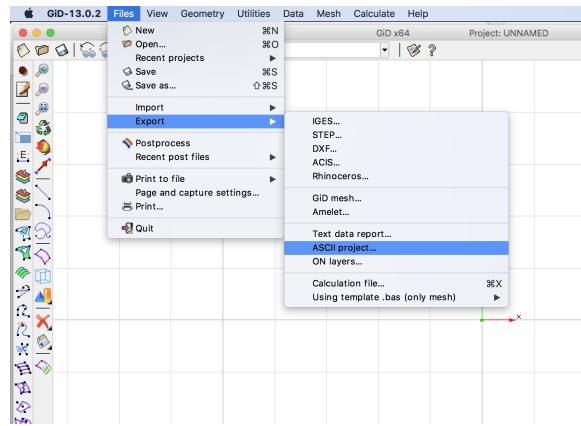


Figure 4:

It will generate several files in a folder and the .msh file is the useful one for us.

3 Different Examples

3.1 Round_2.msh (adaptive_flag=0)

Simple example with the smooth uniform geometry (figure 5 and 6).

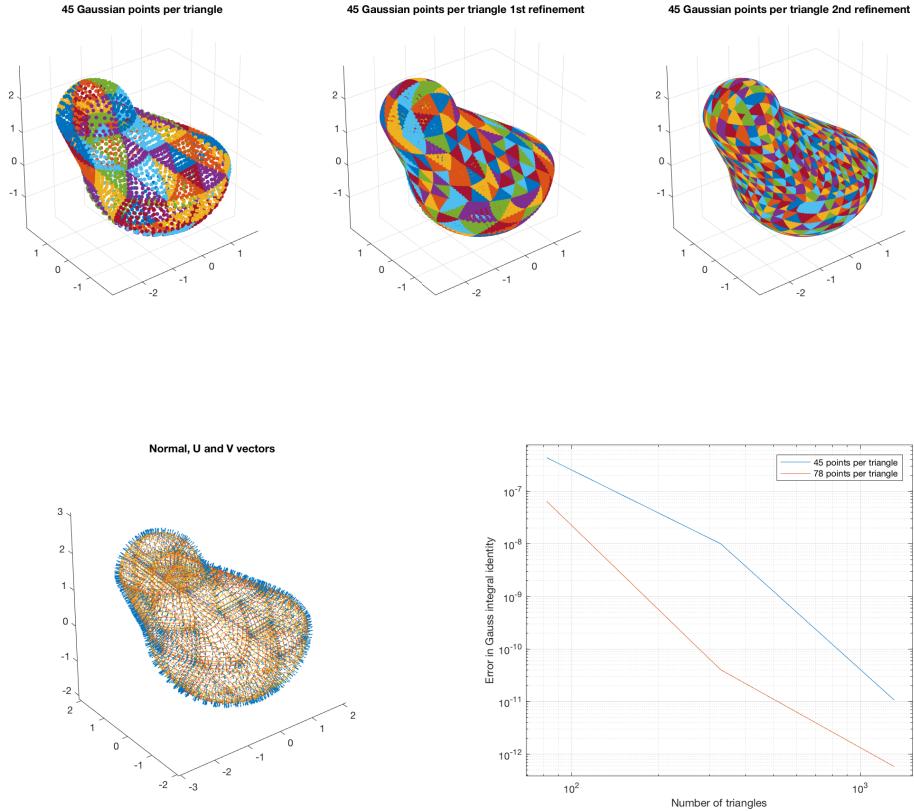


Figure 5: The first row of figures show the refinement process with $n_{refinement} = 0$, $n_{refinement} = 1$, $n_{refinement} = 2$ respectively. The figure down-left shows the normal vector and two tangent orthogonal vectors U, V (all unitary) on each discretization point. The figure down-right shows the convergence obtained.

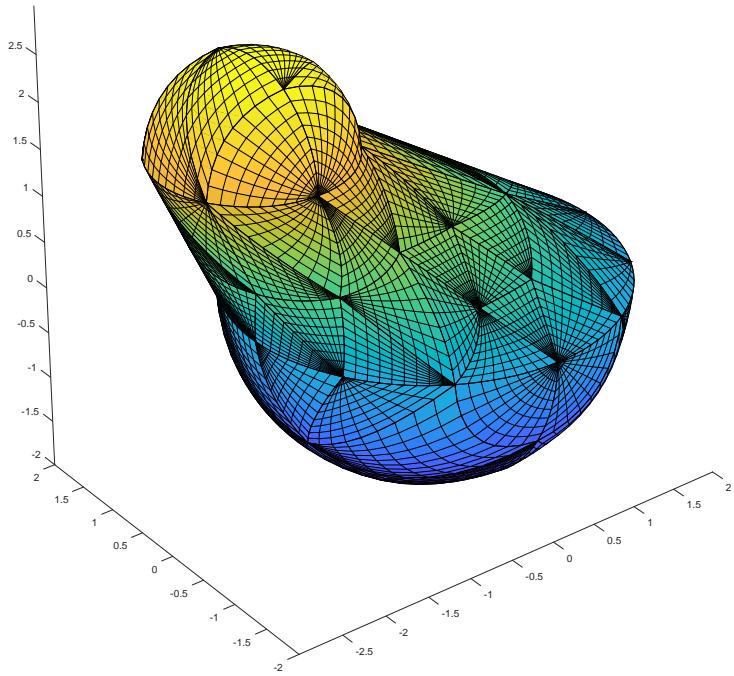


Figure 6: Round_2.msh geometry. Skeleton based on quadratic triangles. The apparent distortion of the triangles is only present in this figure, not in the file (is a plot issue). The set of sources used for the FMM call are 78 Gauss nodes on each quadratic triangle

3.2 Round_1.msh (adaptive_flag=0)

Simple example with the smooth uniform geometry (figure 7 and 8).

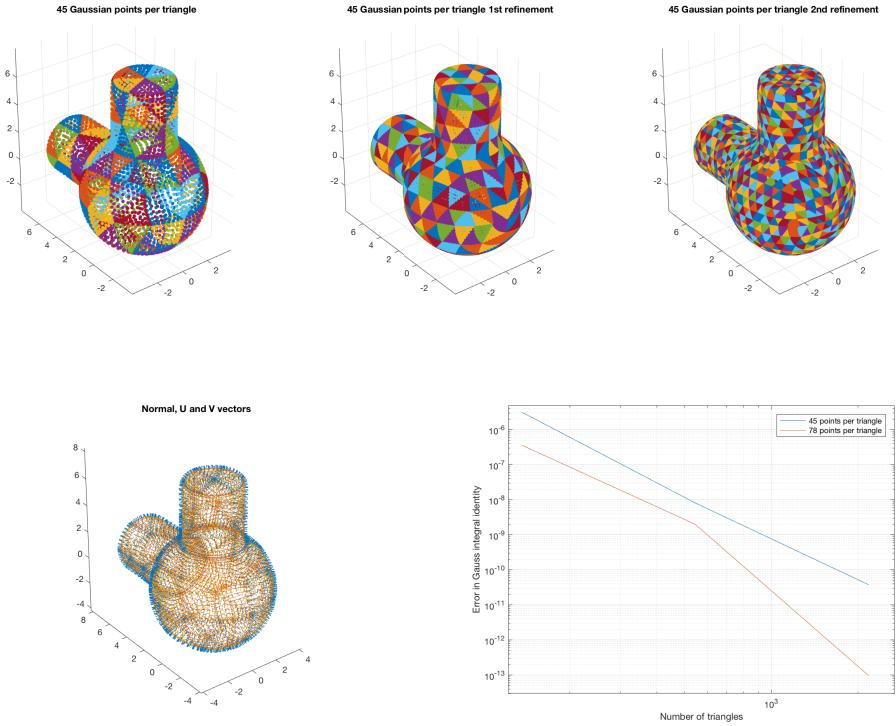


Figure 7: The first row of figures show the refinement process with $n_{refinement} = 0$, $n_{refinement} = 1$, $n_{refinement} = 2$ respectively. The figure down-left shows the normal vector and two tangent orthogonal vectors U, V (all unitary) on each discretization point. The figure down-right shows the convergence obtained.

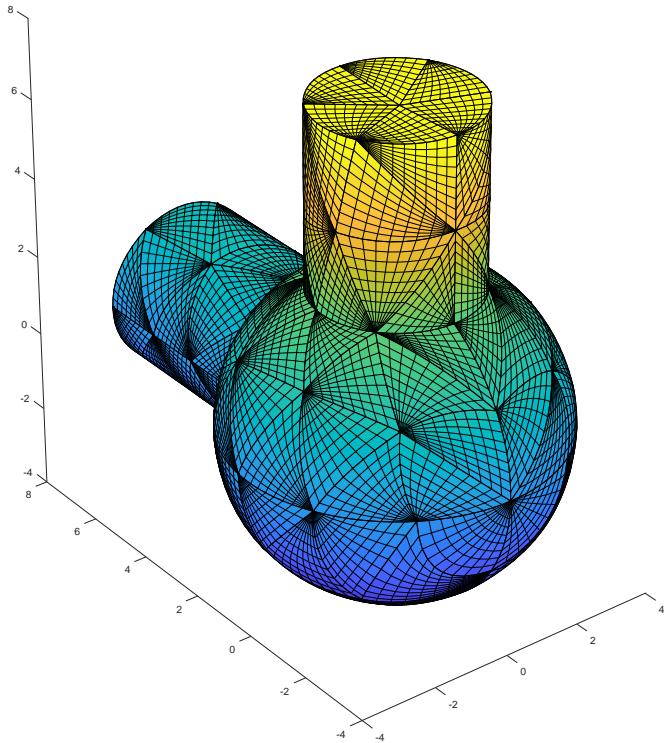


Figure 8: Round_1.msh geometry. Skeleton based on quadratic triangles. The apparent distortion of the triangles is only present in this figure, not in the file (is a plot issue). The set of sources used for the FMM call are 78 Gauss nodes on each quadratic triangle

3.3 Cube_substraction.msh (adaptive_flag=0)

REMARK: We see a saturation in the accuracy at 12 digits (figure 9 down right) because of the Newton threshold and other FMM threshold (iprec=3).

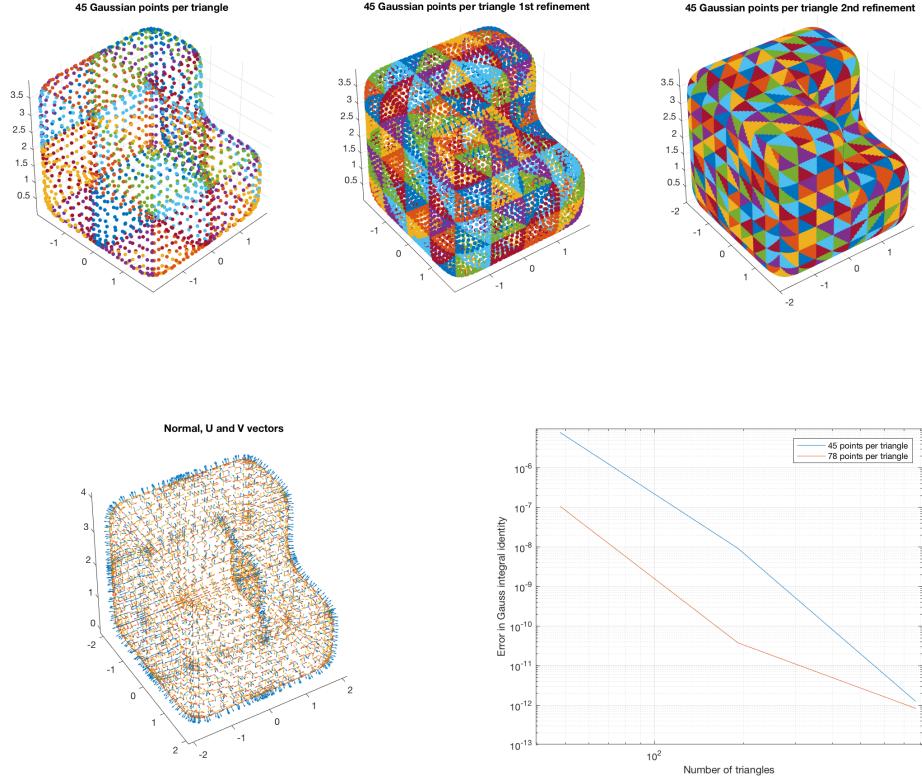


Figure 9: The first row of figures show the refinement process with $n_{refinement} = 0$, $n_{refinement} = 1$, $n_{refinement} = 2$ respectively. The figure down left shows the normal vector and two tangent orthogonal vectors U, V (all unitary) on each discretization point. The figure down right shows the convergence obtained. We see a saturation in the accuracy at 12 digits because of the Newton threshold and other FMM threshold (iprec=3)

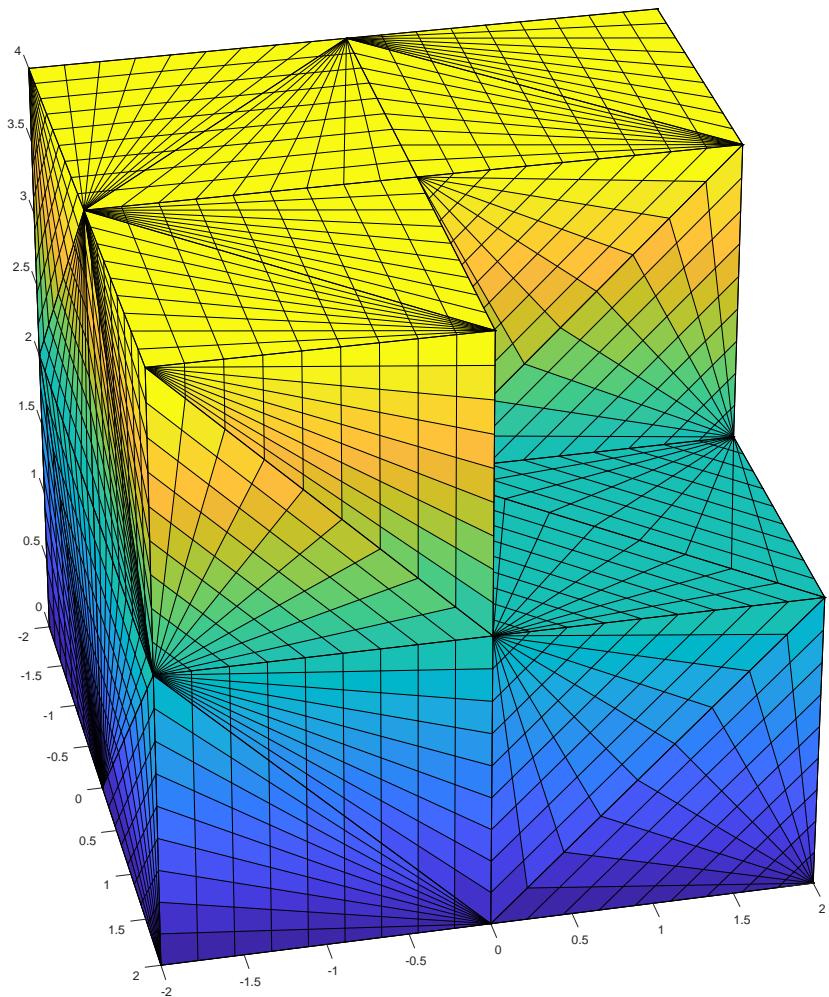


Figure 10: `cube_subtraction.msh` geometry. Skeleton based on quadratic triangles. The apparent distortion of the triangles is only present in this figure, not in the file (is a plot issue). The set of sources used for the FMM call are 78 Gauss nodes on each quadratic triangle

3.4 esfera_esfera.msh (adaptive_flag=0)

Simple example of a spherical geometry with a spherical cavity (figure 11 and 12).

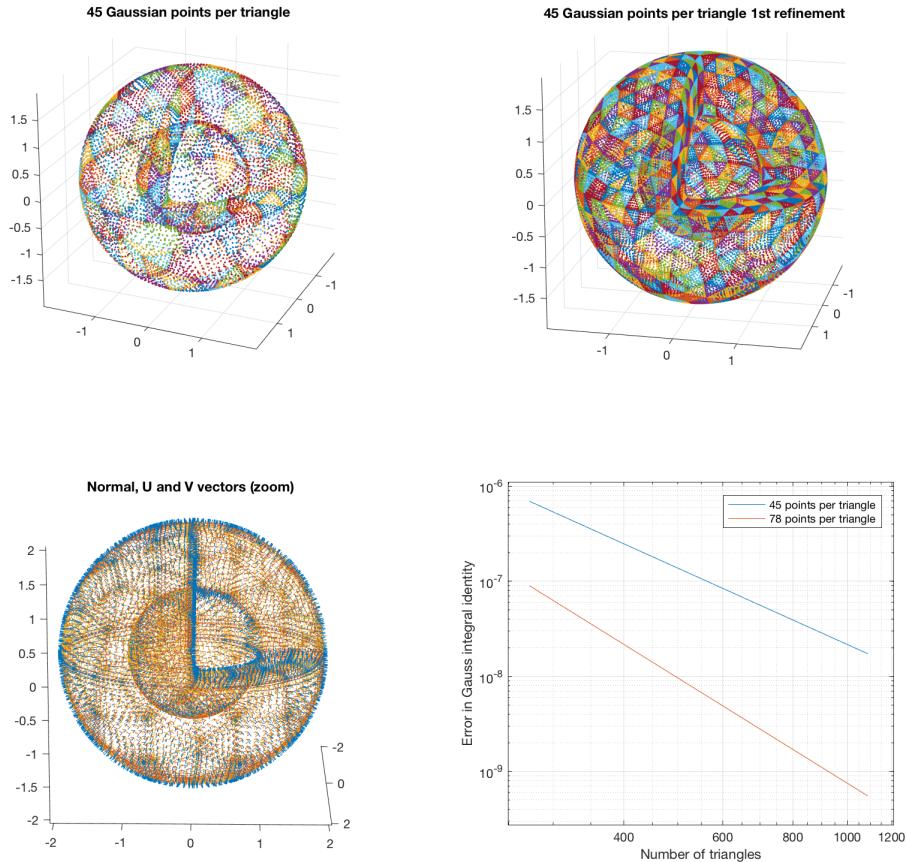


Figure 11: The first row of figures show the refinement process with $n_{refinement} = 0, n_{refinement} = 1$ respectively. The figure down left shows the normal vector and two tangent orthogonal vectors U, V (all unitary) on each discretization point. The figure down right shows the convergence obtained. We can see the method working on simple geometries with cavities

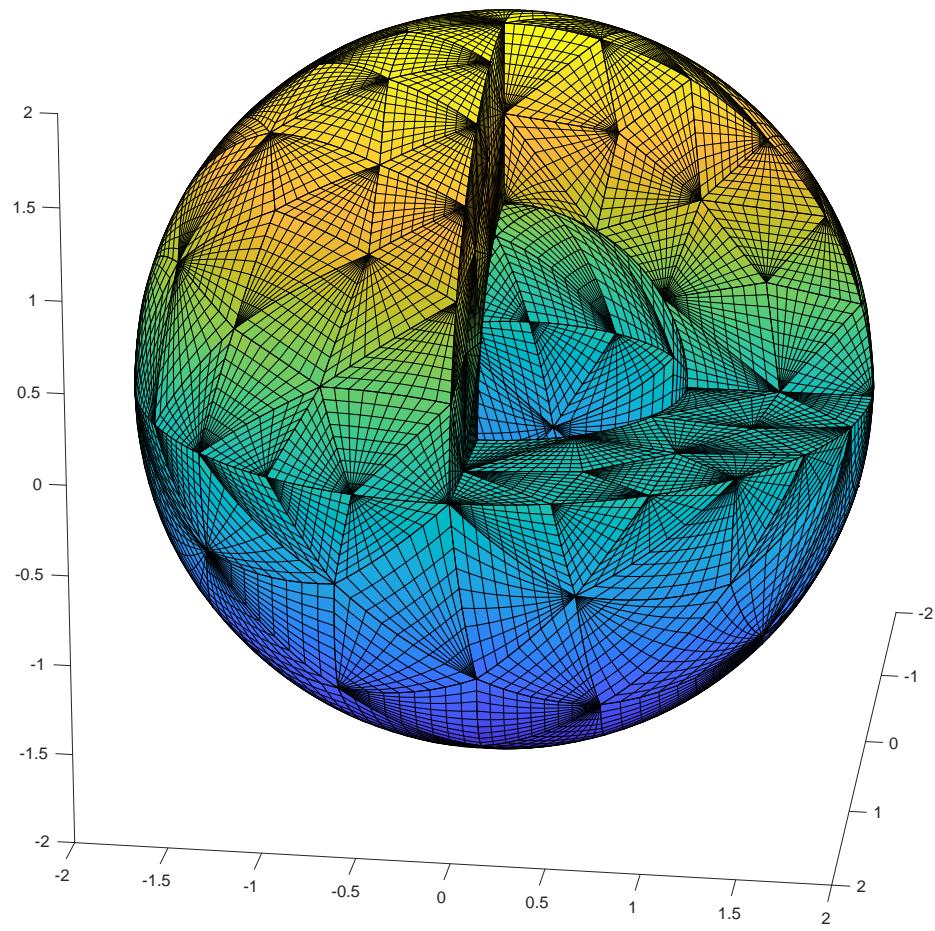


Figure 12: esfera_esfera.msh geometry. Skeleton based on quadratic triangles. The apparent distortion of the triangles is only present in this figure, not in the file (is a plot issue). The set of sources used for the FMM call are 78 Gauss nodes on each quadratic triangle

3.5 high_genus.msh (adaptive_flag=0)

Example of a geometry with $genus = 25$ (figure 13 and 14).

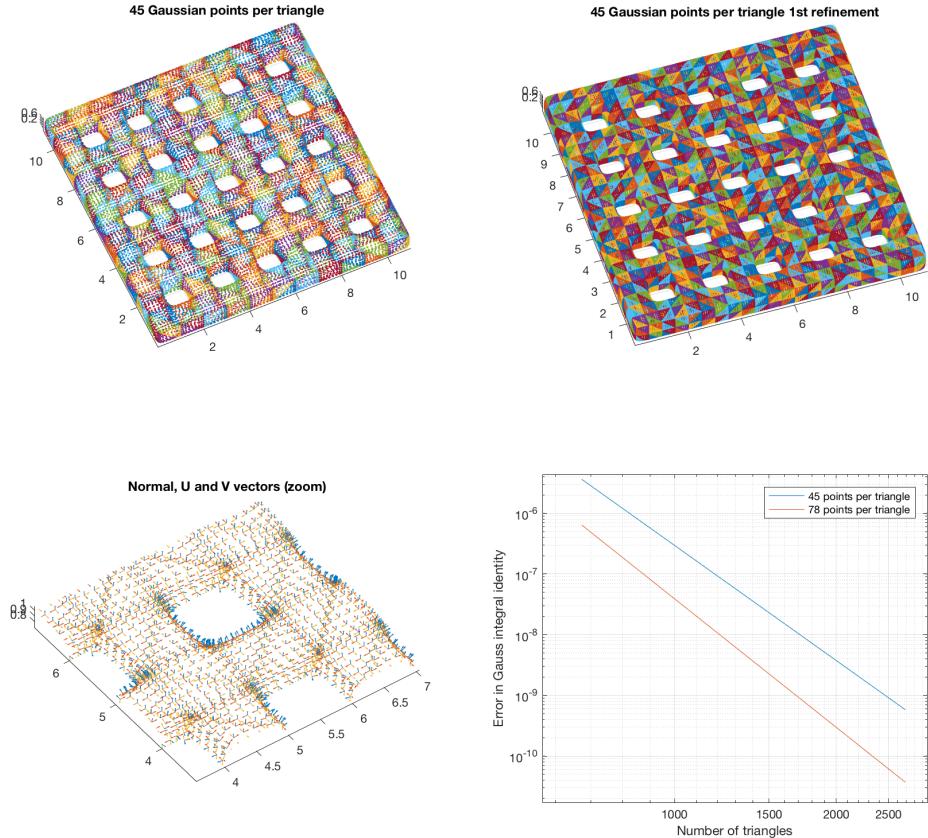


Figure 13: The first row of figures show the refinement process with $n_{refinement} = 0, n_{refinement} = 1$ respectively. The figure down-left shows the normal vector and two tangent orthogonal vectors U, V (all unitary) on each discretization point. The figure down-right shows the convergence obtained.

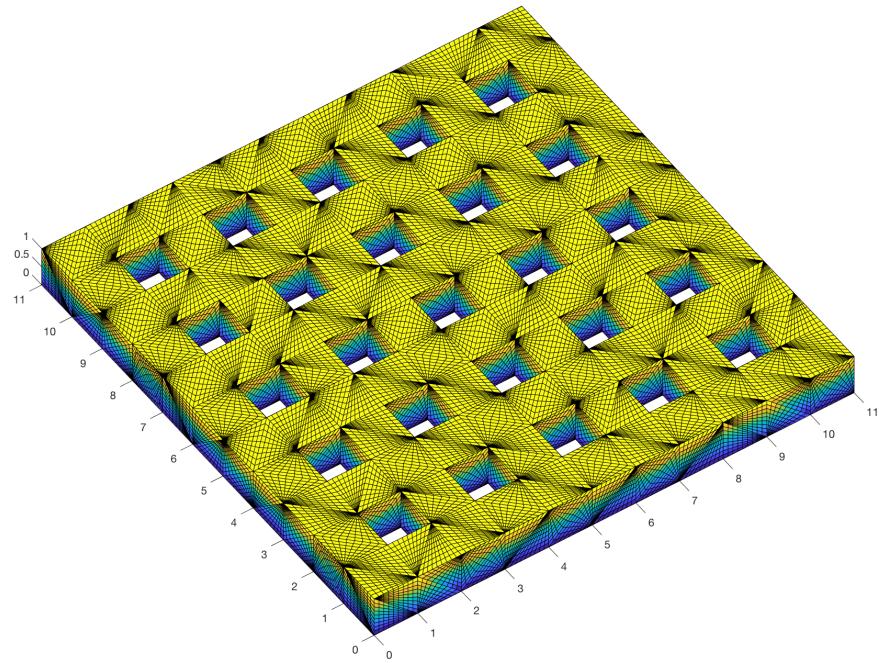


Figure 14: `high_genus.msh` geometry. Skeleton based on quadratic triangles. The apparent distortion of the triangles is only present in this figure, not in the file (is a plot issue). The set of sources used for the FMM call are 78 Gauss nodes on each quadratic triangle

3.6 Multiscale_1.msh (adaptive_flag=1)

Simple example with a small detail on top to test adaptivity (figure 15 and 16).

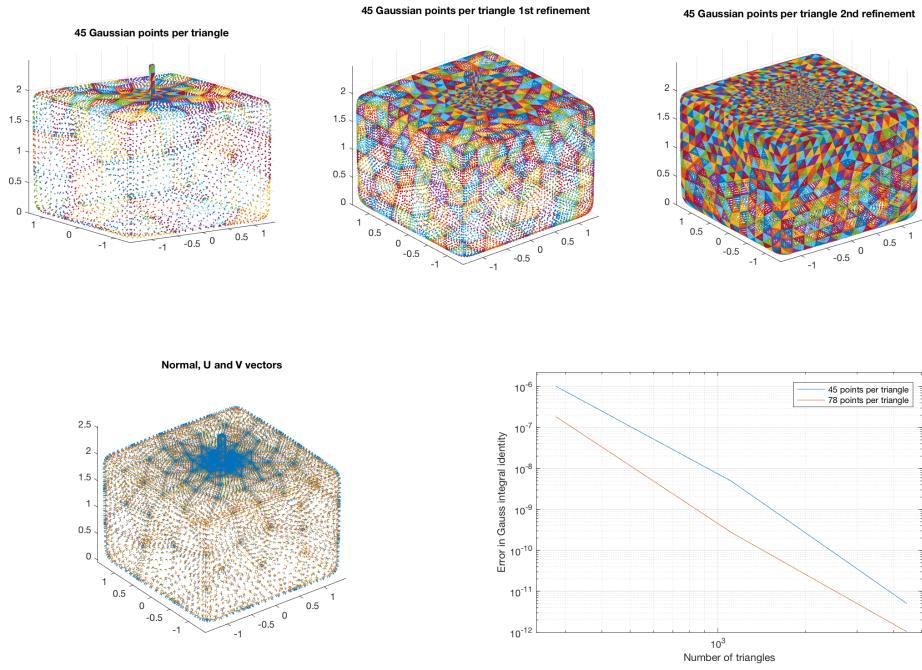


Figure 15: The first row of figures show the refinement process with $n_{refinement} = 0, n_{refinement} = 1, n_{refinement} = 2$ respectively. The figure down left shows the normal vector and two tangent orthogonal vectors U, V (all unitary) on each discretization point. The figure down right shows the convergence obtained.

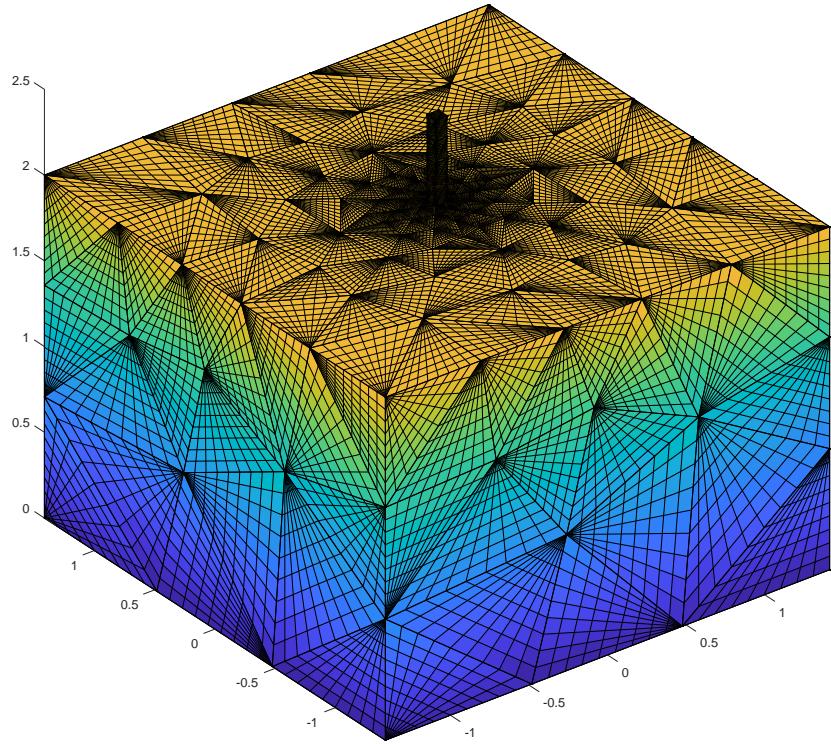


Figure 16: Multiscale_1.msh geometry. Skeleton based on quadratic triangles. The apparent distortion of the triangles is only present in this figure, not in the file (is a plot issue). The set of sources used for the FMM call are 78 Gauss nodes on each quadratic triangle

3.7 capsule_multiscale.msh (adaptive_flag=1)

Another example with a small detail to test adaptivity (figure 17 and 18).

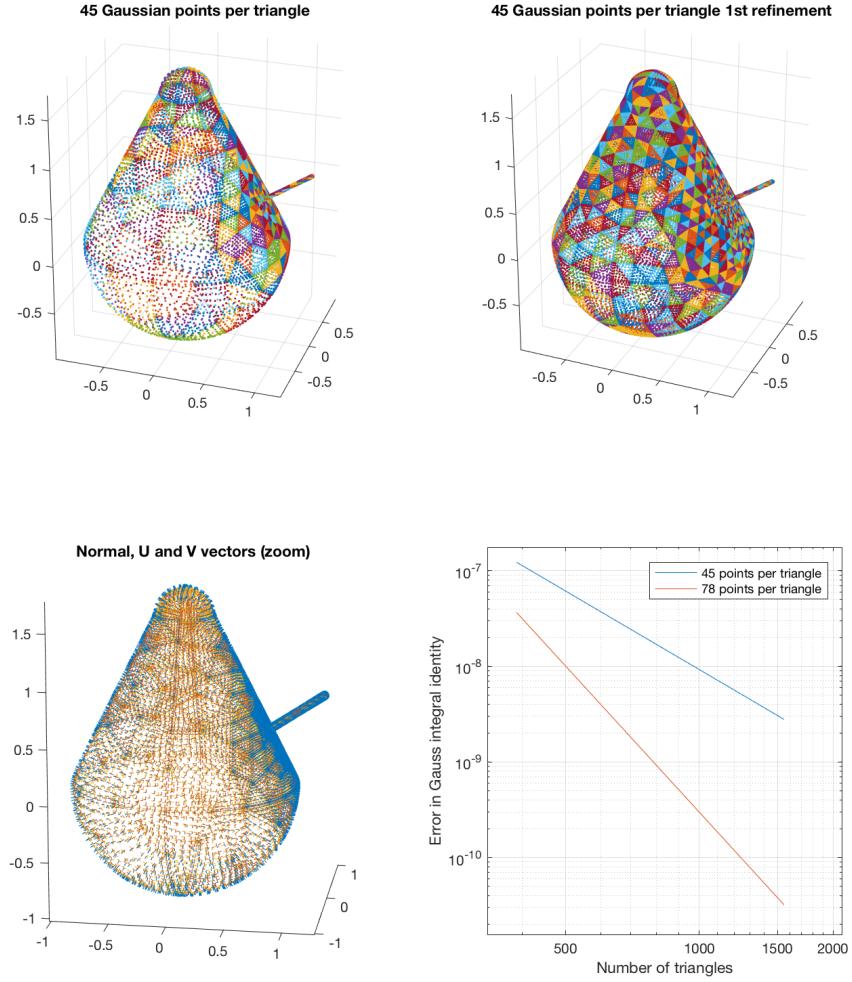


Figure 17: The first row of figures show the refinement process with $n_{refinement} = 0, n_{refinement} = 1$ respectively. The figure down left shows the normal vector and two tangent orthogonal vectors U, V (all unitary) on each discretization point. The figure down right shows the convergence obtained.

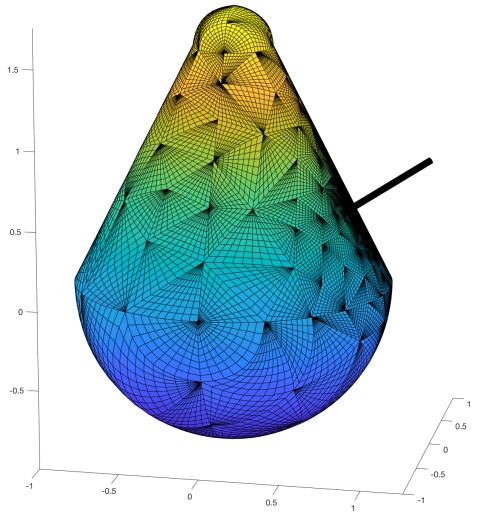


Figure 18: `capsule_multiscale.msh` geometry. Skeleton based on quadratic triangles. The apparent distortion of the triangles is only present in this figure, not in the file (is a plot issue). The set of sources used for the FMM call are 78 Gauss nodes on each quadratic triangle

3.8 cubo_esfera_multires.msh (adaptive_flag=1)

Example of a geometry with a spherical cavity and a small detail inside the cavity (figure 19 and 21).

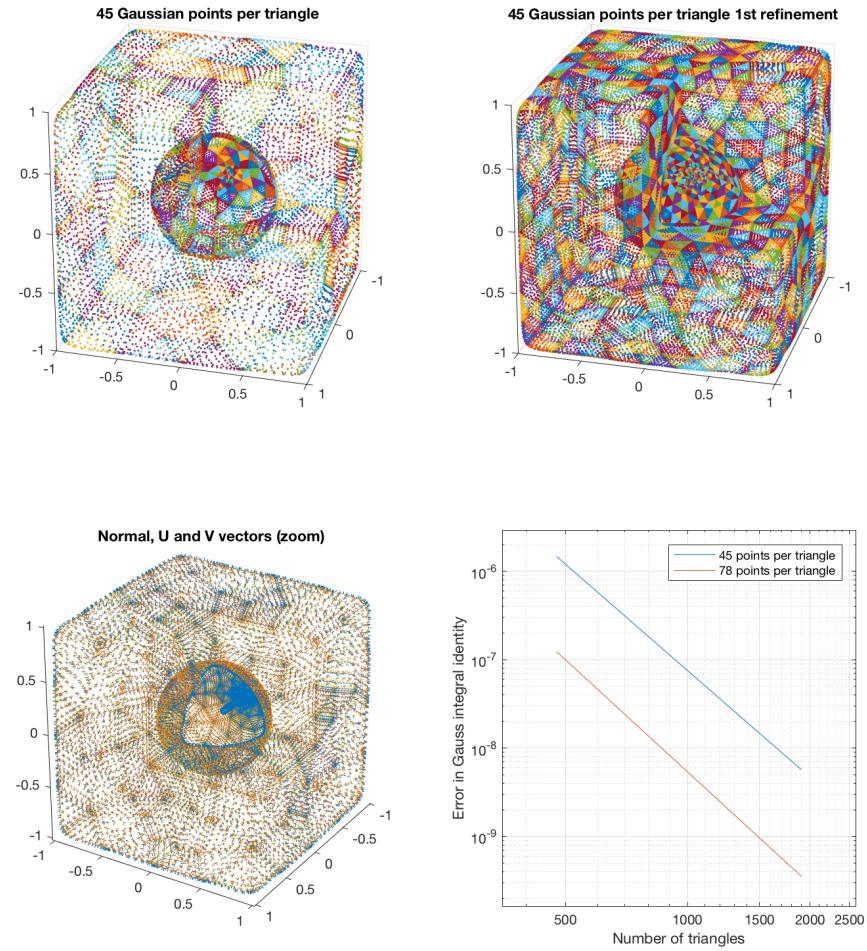


Figure 19: The first row of figures show the refinement process with $n_{refinement} = 0, n_{refinement} = 1$ respectively. The figure down left shows the normal vector and two tangent orthogonal vectors U, V (all unitary) on each discretization point. The figure down right shows the convergence obtained.

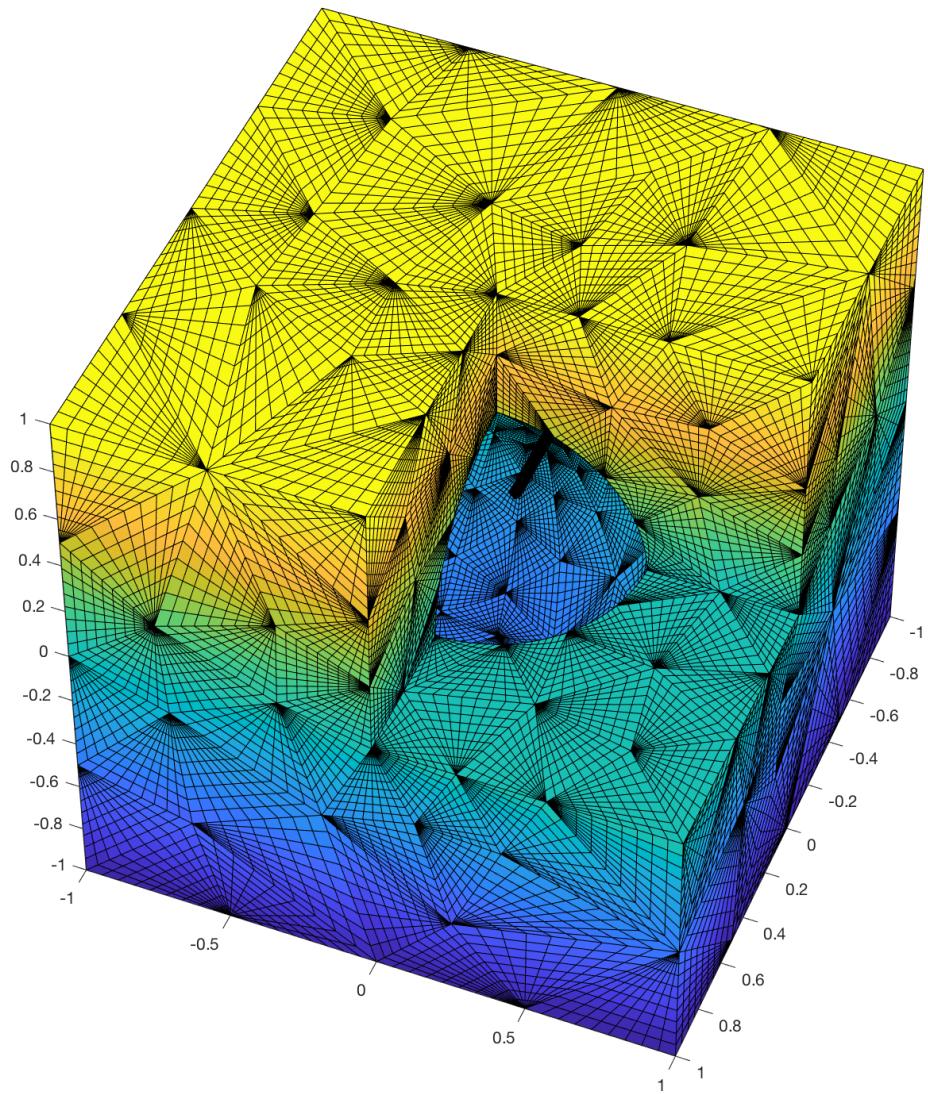


Figure 20: cubo_esfera_multires.msh geometry. Skeleton based on quadratic triangles. We see the small detail inside the inner spherical cavity. The apparent distortion of the triangles is only present in this figure, not in the file (is a plot issue). The set of sources used for the FMM call are 78 Gauss nodes on each quadratic triangle

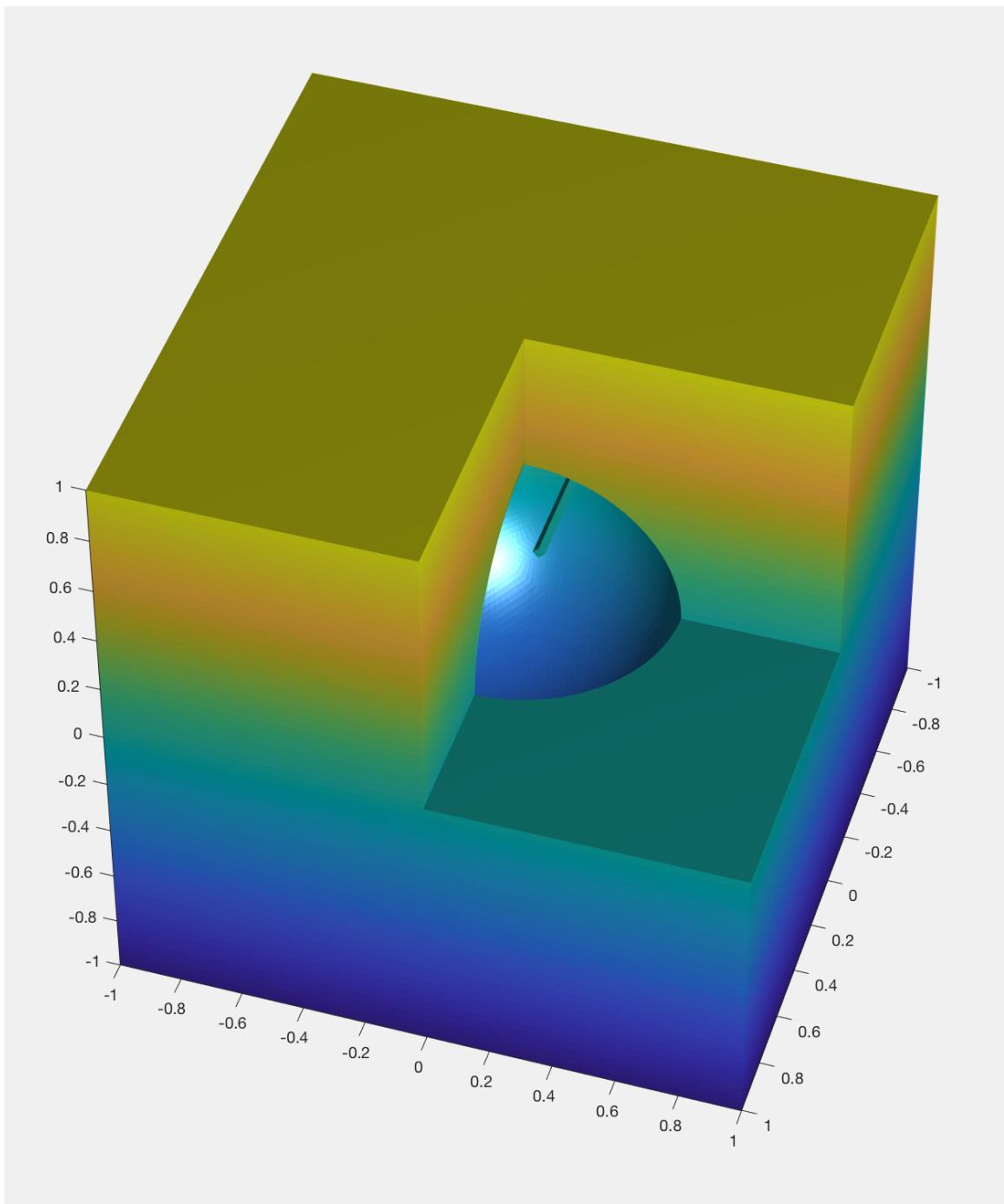


Figure 21: cubo_esfera_multires.msh geometry. Skeleton based on quadratic triangles. We see the small detail inside the inner spherical cavity. The apparent distortion of the triangles is only present in this figure, not in the file (is a plot issue). The set of sources used for the FMM call are 78 Gauss nodes on each quadratic triangle

3.9 pico_2.msh (adaptive_flag=1)

Geometry with a sharp peak (figure 22 and 23). This is to show that sharp peaks are possible avoiding the caustic effect if we round the skeleton and refine the skeleton when approaching to the peak.

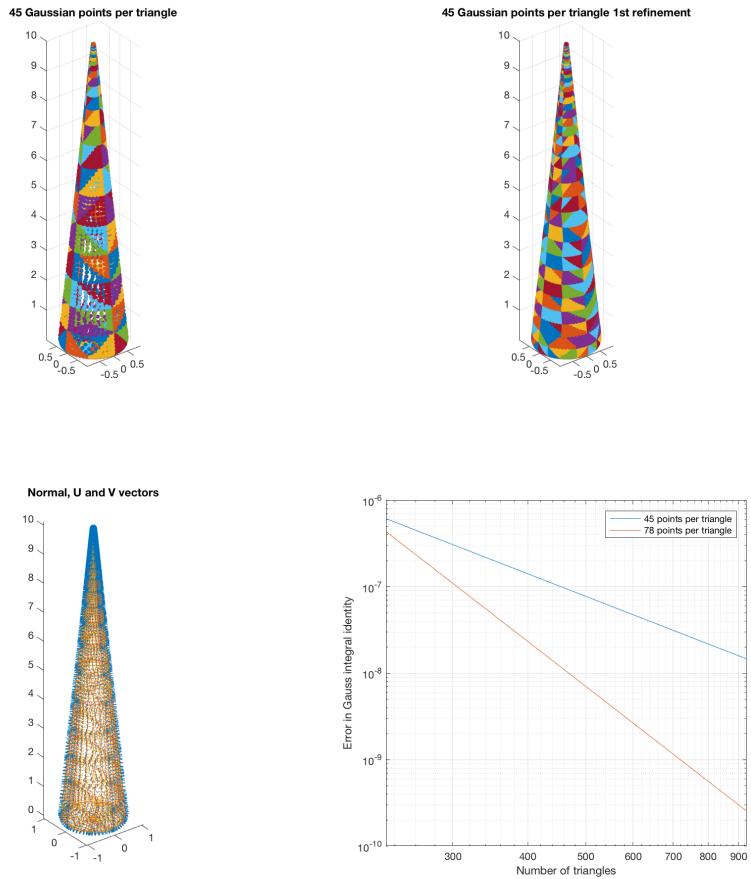


Figure 22: The first row of figures show the refinement process with $n_{refinement} = 0, n_{refinement} = 1$ respectively. The figure down left shows the normal vector and two tangent orthogonal vectors U, V (all unitary) on each discretization point. The figure down right shows the convergence obtained.

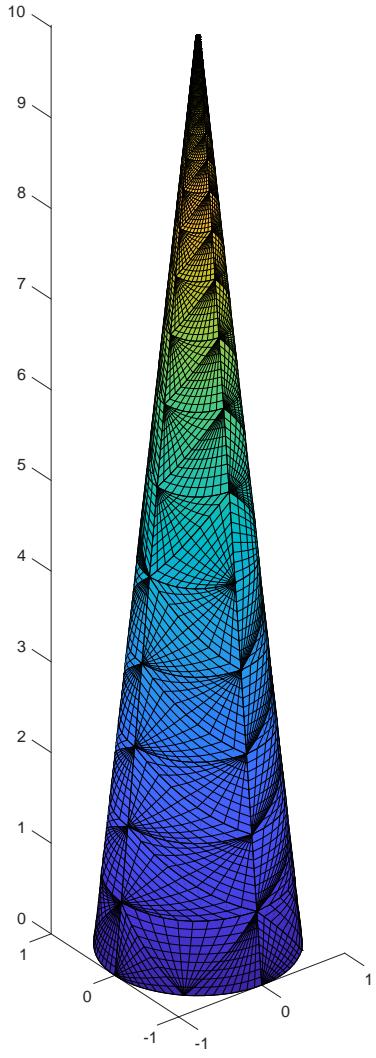


Figure 23: pico_2.msh geometry. Skeleton based on quadratic triangles. The apparent distortion of the triangles is only present in this figure, not in the file (is a plot issue). The set of sources used for the FMM call are 78 Gauss nodes on each quadratic triangle

3.10 sci_fi_2.msh (adaptive_flag=0)

Another example of a complex geometry. Genus 12 and complex tubular connections. No refinement study yet, accuracy obtained for $n_{refinement} = 0$ and n_order_sf=45 is $Err = 5.8 \cdot 10^{-8}$

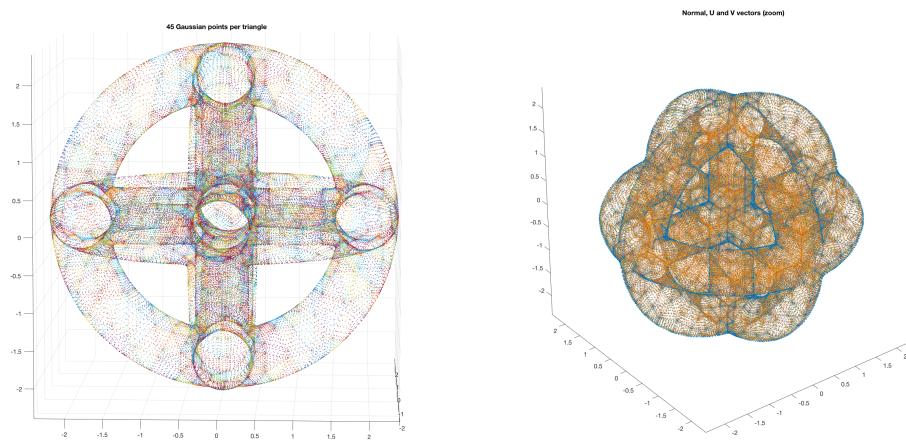


Figure 24:

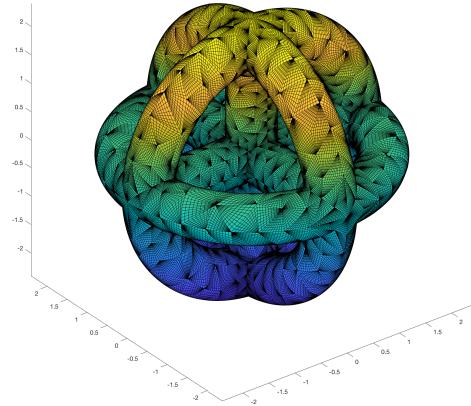


Figure 25: Different view of the skeleton

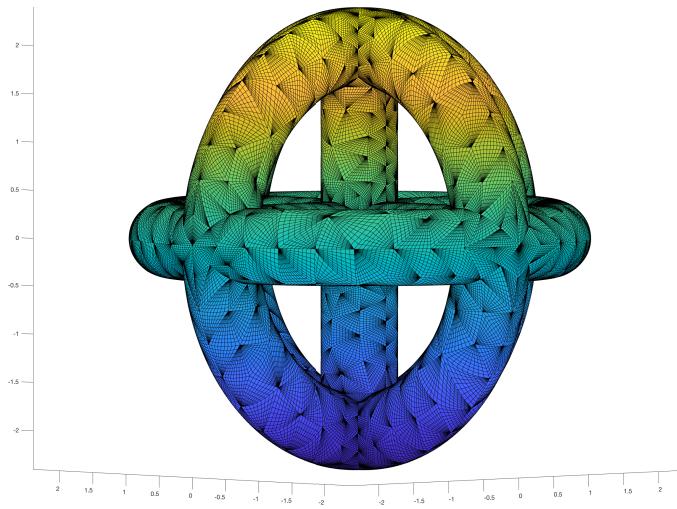


Figure 26: Different view of the skeleton

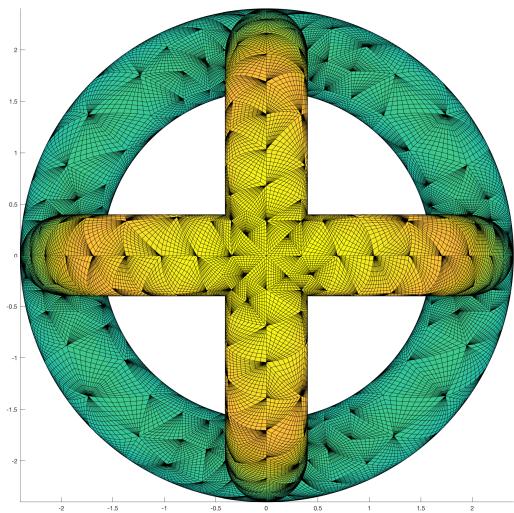


Figure 27: Different view of the skeleton

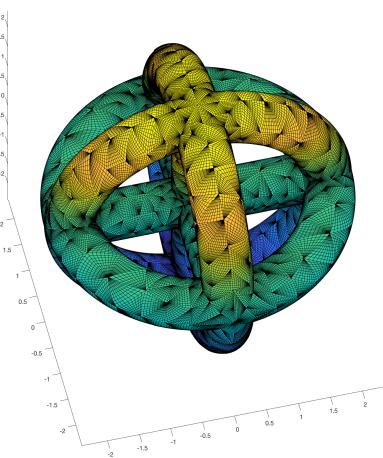


Figure 28: Different view of the skeleton

3.11 open_cavity_30deg_v2.msh (adaptive_flag=0)

Open cavity. Standard challenging problem for high frequency EM scattering. No refinement study yet, accuracy obtained for $n_{refinement} = 0$ and n_order_sf=45 is $Err = 6.3 \cdot 10^{-6}$

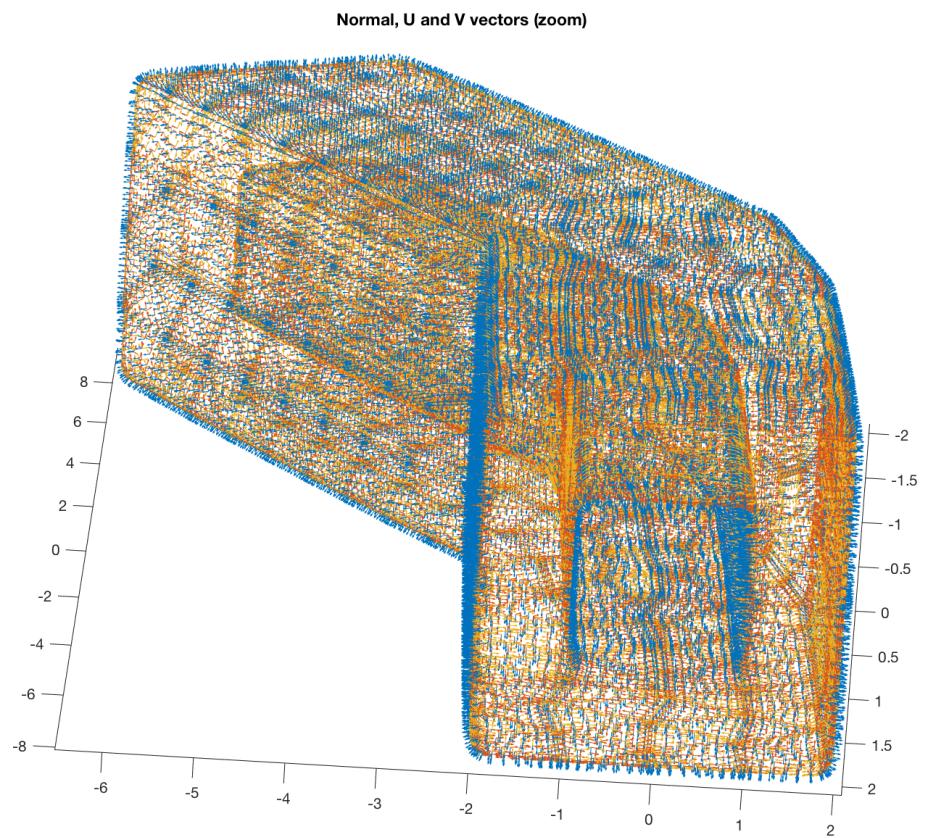


Figure 29:

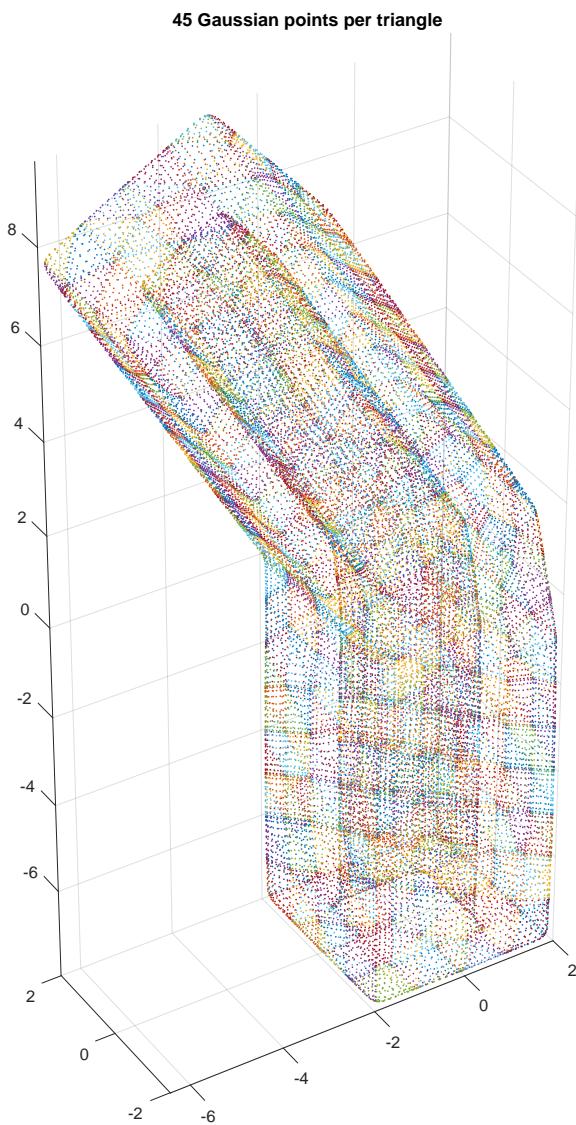


Figure 30:

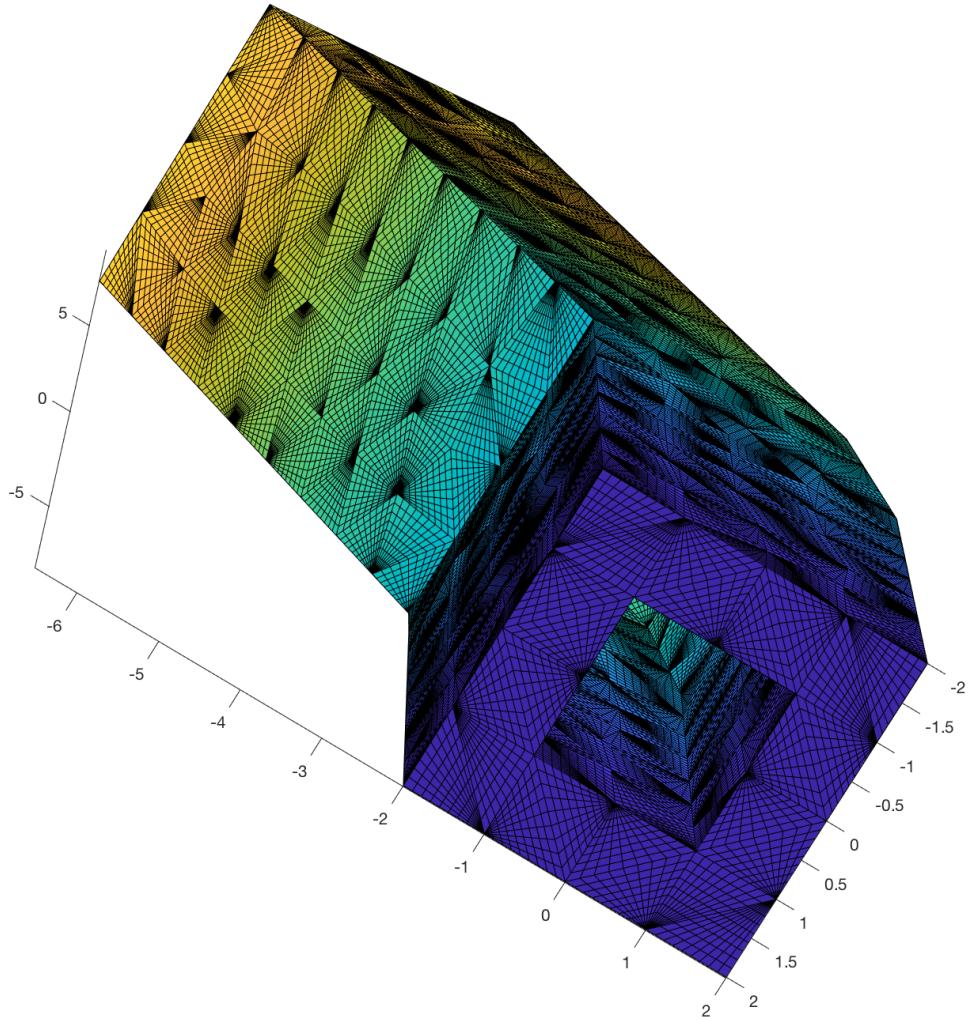


Figure 31: View of the skeleton used for the open cavity

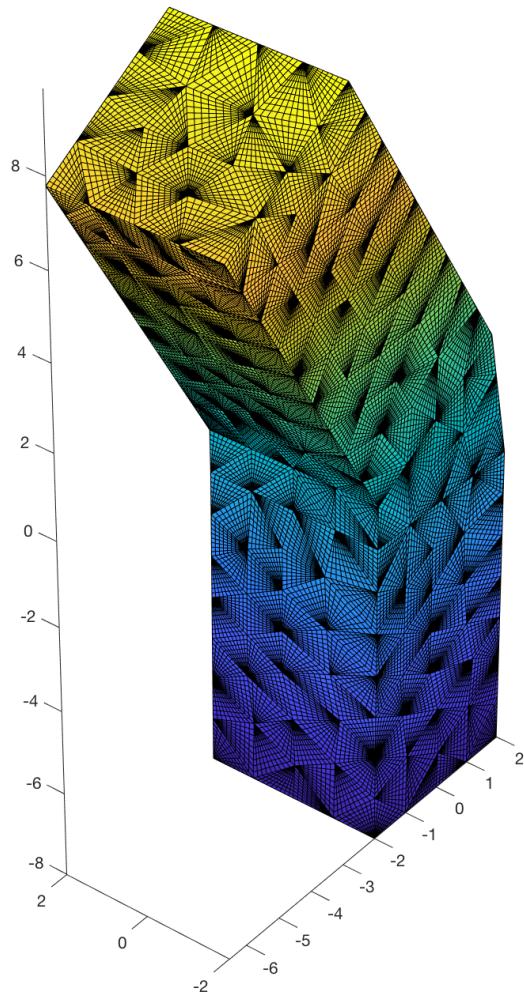


Figure 32: View of the skeleton used for the open cavity

3.12 parabolic_antenna.msh (adaptive_flag=1)

Parabolic reflector with support for a small feed antenna at its focus. Same geometry inserted in the warship (last example). No refinement study yet, accuracy obtained for $n_{refinement} = 0$ and n_order_sf=45 is $Err = 4.8 \cdot 10^{-6}$

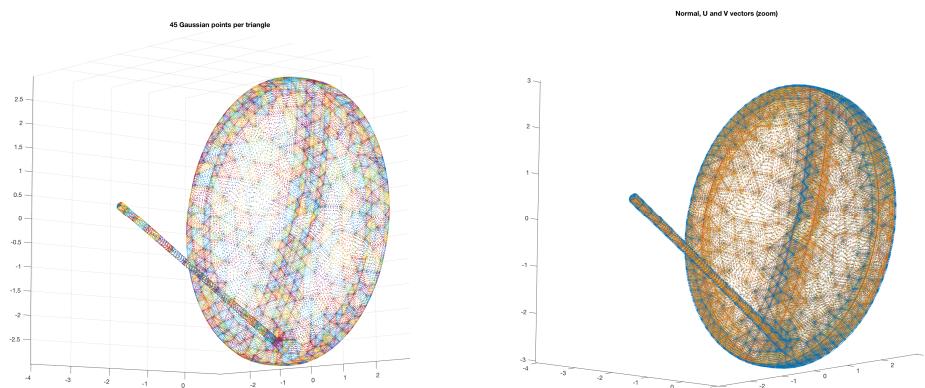


Figure 33:

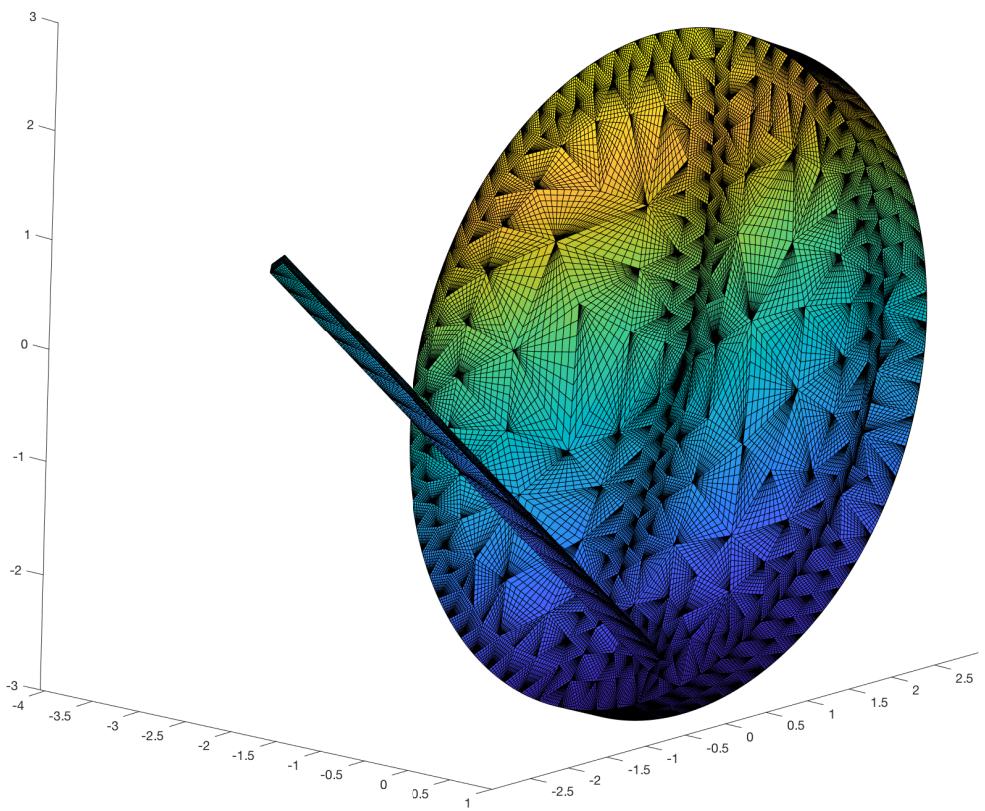


Figure 34:

3.13 two_cavity_filter_1.msh (adaptive_flag=1)

Two cavity filter with two tuning impedances at the center of each cavity. No refinement study yet, accuracy obtained for $n_{refinement} = 0$ and n_order_sf=45 is $Err = 6 \cdot 10^{-8}$

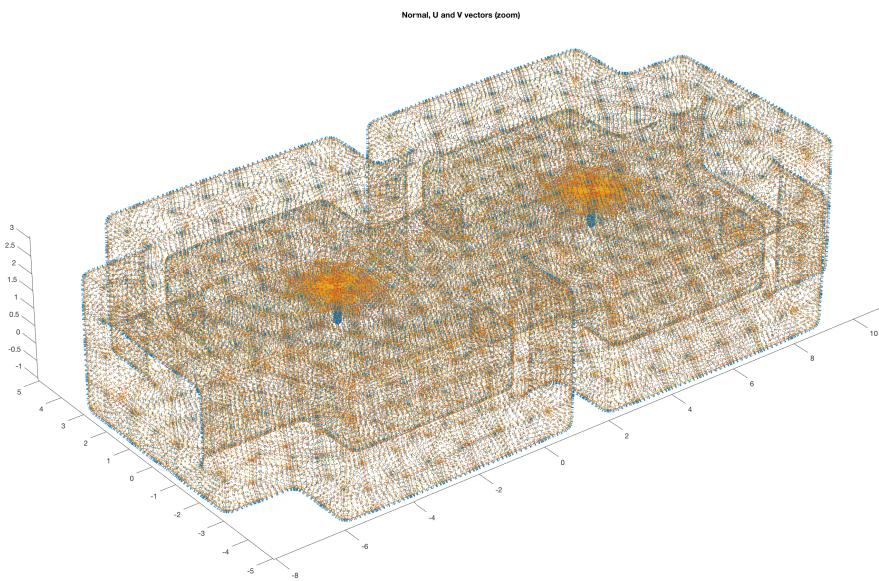


Figure 35:

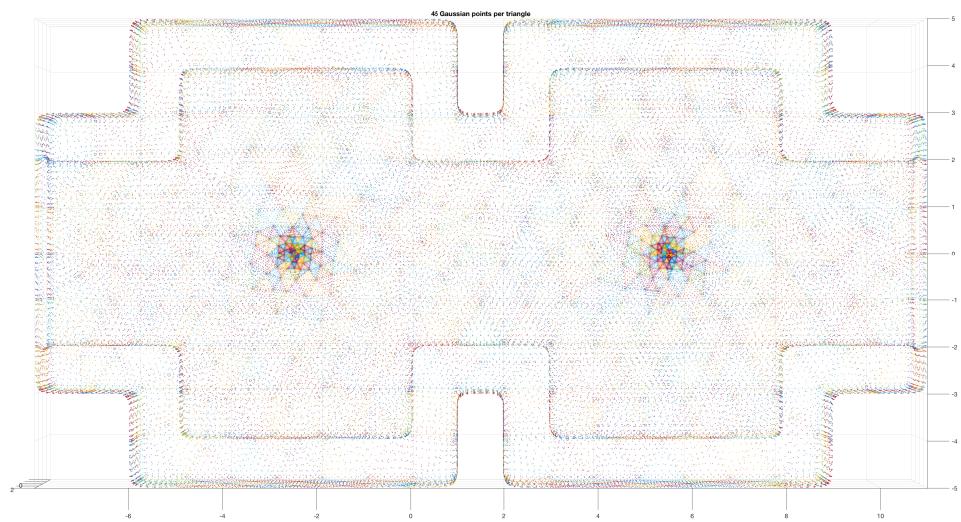


Figure 36:

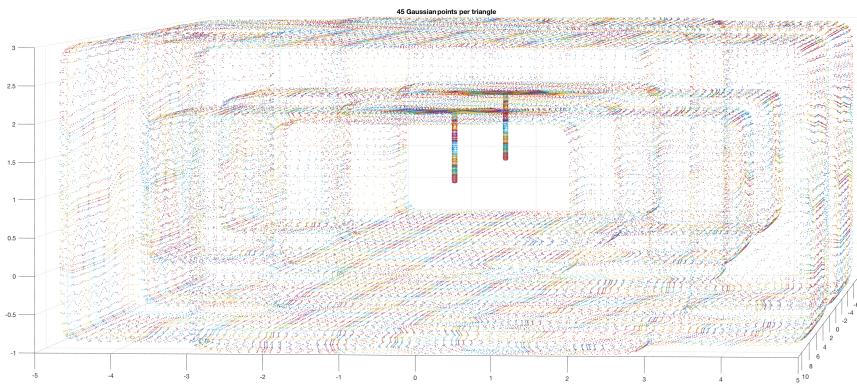


Figure 37:

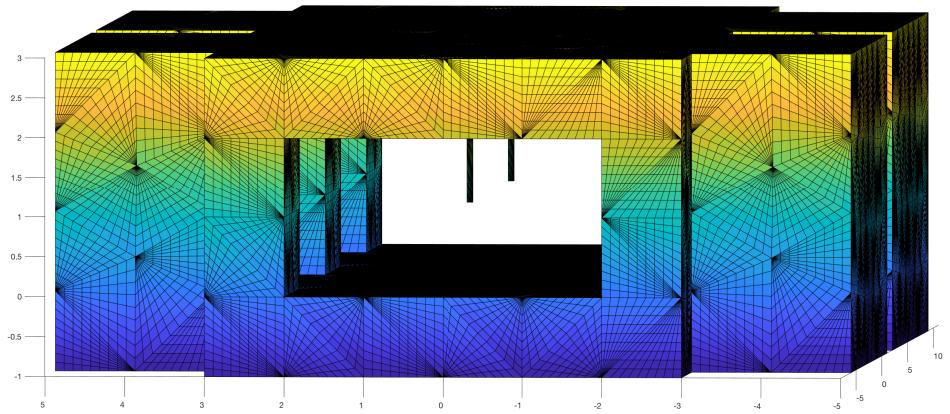


Figure 38: View of the skeleton. Detail of the tuning impedances

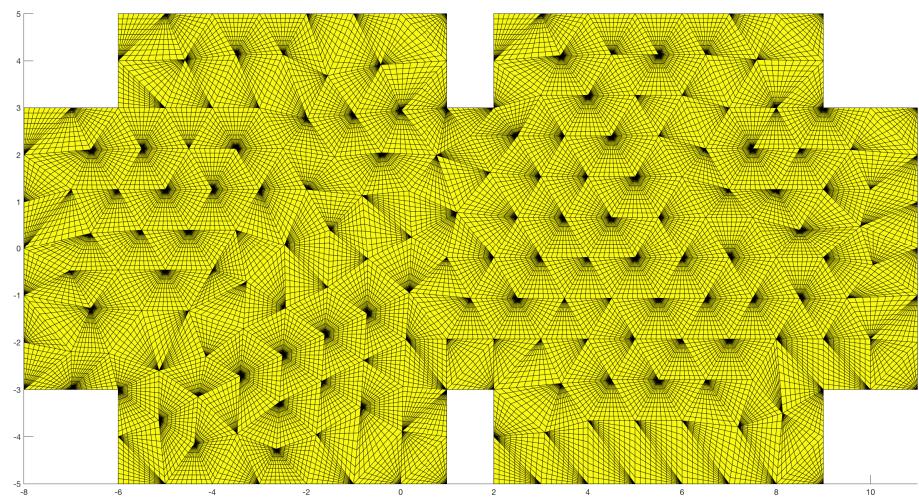


Figure 39: Top view of the skeleton

3.14 mi_barco_simple_13.msh (adaptive_flag=1)

Complex geometry with some small details (monopole antennas and parabolic reflector). No refinement study yet, accuracy obtained for $n_{refinement} = 0$ and n_order_sf=45 is $Err = 2.58 \cdot 10^{-7}$. I tried some refinement but no accuracy was gained. I have to study in detail which part of the ship is causing troubles. I'll study the convergence with earlier versions of the ship with fewer details. Geometries with this degree of complexity are still challenging (with this version of the code).

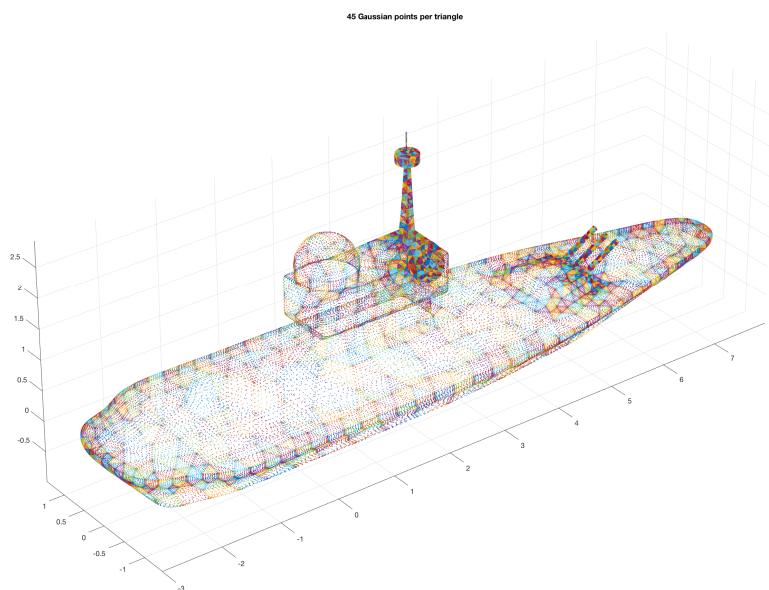


Figure 40:

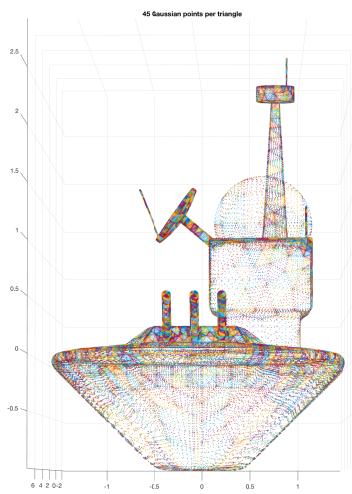


Figure 41:

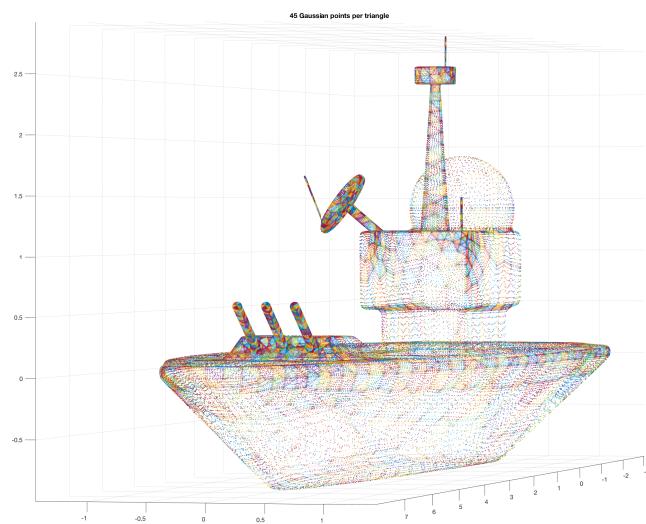


Figure 42:

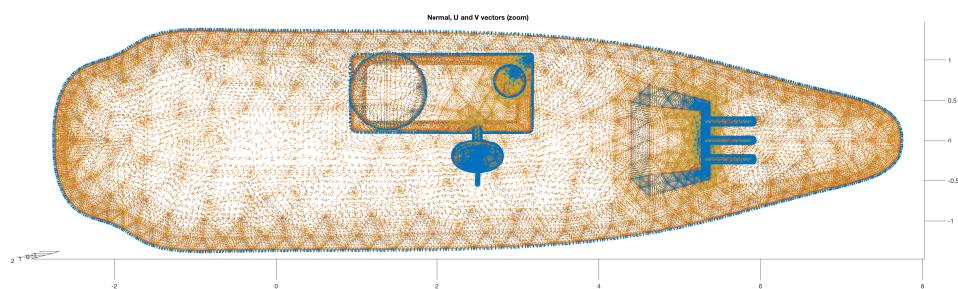


Figure 43:

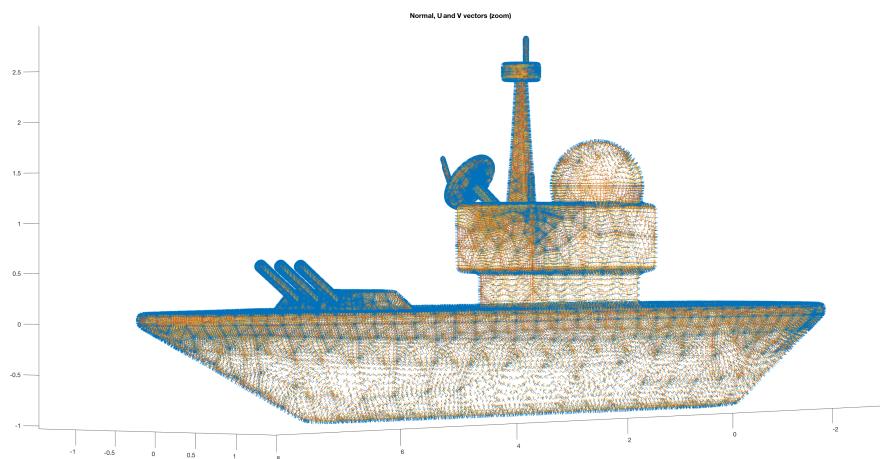


Figure 44: