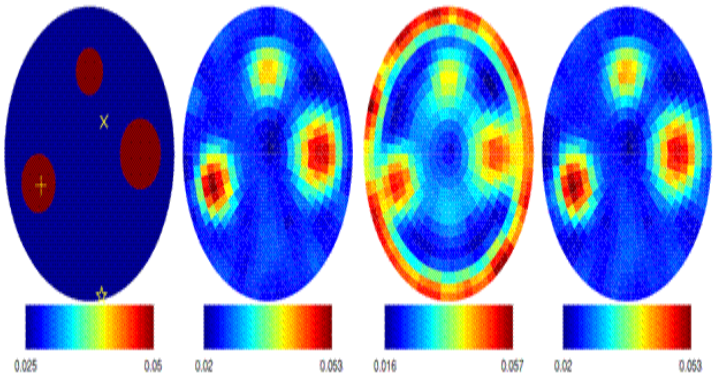


Toast++ and Gmsh installation + DOT simulation instructor

Ron Milutin
Oren Schor

March 2024

TABLE OF CONTENTS

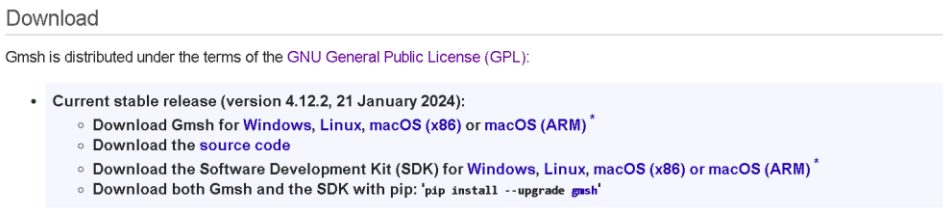
1.0 Installation

A. INSTALLATION

Gmsh

Installing Gmsh is quite simple.

- Go to https://gmsh.info/
- Click on download for your appropriate operation system.

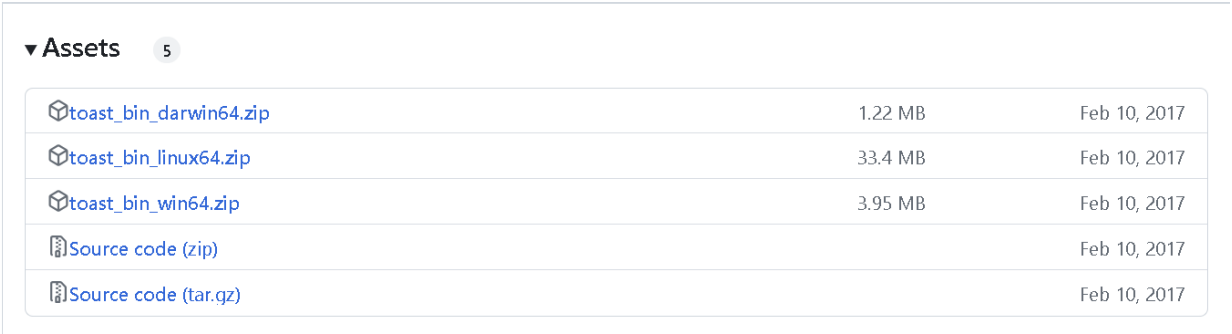


- In your downloads file, the exe file is the Gmsh user application.
- Gmsh is installed.

1.2 Toast++ Installation

Installing Toast++ is quite longer.

- Go to https://github.com/toastpp/toastpp/releases
- Download the two marked files:



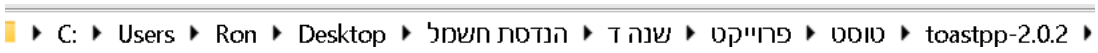
- Unzip them.
- We will run toast++ library on MATLAB so to check whether you have a proper mex compiler you need to run the command "mex -setup" in your MATLAB command window.

- If it still does not work, we will force a mex compiler by attaching the default compiler for MATLAB and then replace it. To do so please follow these steps:
- Log in to the HOME tab in MATLAB, click on Add-Ons:
- In the window that opens, select Get Add-Ons
- The Add-Ons Explorer should open. Write in the search bar "mingw"
- Select MATLAB Support for MinGW-w64 C / C ++ Compiler
- You are supposed to get to the next window:

.

- Click "Install"

- Click "I accaept" and then "Next" and at the end "finish".
- After the installation is complete Restart MATLAB.
- Type again in the command "mex -setup", it should work now.
- Under the bin file->win->x64, copy x64 directory into toastpp-2.0.2\win directory.
- In order to make an m file work using Toast++, ensure you are under the appropriate directory:

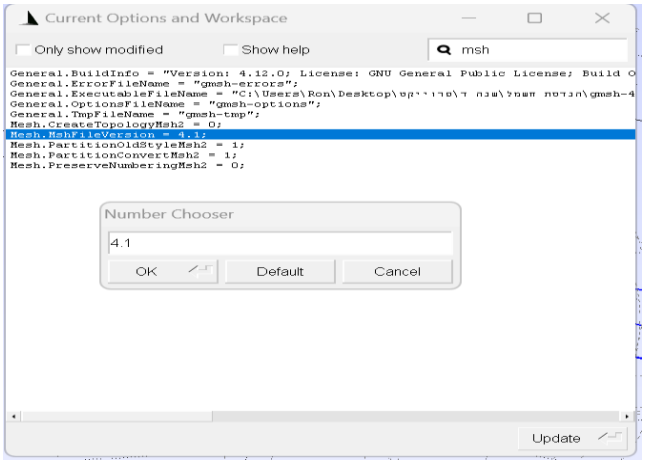


- TOAST++ installation complete

1.3 Integrating Gmsh with Toast++

The Following example is using Toast++ 2.0.2 MATLAB toolbox and Gmsh 4.12.0. Note that MSH file should be saved in an older revision (2.x) as Toast++ last update was in 2017 while Gmsh is updated monthly (But thankfully has a backward compatibility).
Integration process:

- Click "Help"-> "Current Options and Workspace"
- Change here from 4.1 to 2.2



- click update.
- Now, when you save a mesh file as .msh, it is formatted correctly to be compatible with Toast++.

2.0 SIMULATION

B. SIMULATION

We are excited to present a 3D simulation using Toast++ and Gmsh. To create a realistic simulation resembling a genuine Diffuse Optical Tomography (DOT) scan, we acquired a breast mesh from Digi breast. This mesh serves as the foundation for our simulation, accurately replicating the shapes of real breasts, including the presence of a tumor inside.

2.1 Files Overview

After downloading all the files put them in a directory under Toast 2.0.2 directory as explained earlier.

Five files are needed for this simulation:

a. Base_Mesh_struct.mat – This file signifies an array comprising elements and nodes extracted from the Digi Breast dataset.
b. conv_nodes_elements_arrays_to_msh.m - This section defines a function to convert node and element arrays to a Gmsh .msh file format.
c. stretch_mesh.m - This script defines a function to stretch a mesh along specified scaling factors.
d. project_meshrefine.m - This script defines a function to refine a mesh within a spherical region and calculate element centroids.
e. test_3d_scan.m - This section initializes the script, loads a base mesh, selects a random node, refines the mesh around the selected node, and simulates a DOT scan.

The script utilizes a breast mesh, refines it, and simulates a DOT scan. The functions conv_nodes_elements_arrays_to_msh, stretch_mesh, and project_meshrefine contribute to converting the refined mesh to Gmsh.msh format, stretching the mesh, and refining it within a specified region, respectively. The integration is achieved through the call and utilization of these functions in the main script. More specific description about the scripts can be found as comments in the files.

Users can customize various parameters to simulate different Diffuse Optical Tomography (DOT) scans with distinct characteristics. The customizations are in the stretch_mesh.m and test_3d_scan.m. These parameters include:

- Source and Detector Locations:
  Modify the number of sources and detectors above the breast mesh. User can change them by adjusting nqm.

- Breast Mesh Scales:
  In the stretch_mesh function, users can change the scaling factors (scaleX, scaleY, scaleZ) to stretch or compress the breast mesh along different axes. These factors control the size of the breast mesh.

- Tumor Mesh Size and Location:
  When calling the project_meshrefine function, users can specify the parameters for refining the mesh within a spherical region, such as centroid, radius, and maxvol. These parameters determine the size and location of the simulated tumor within the breast mesh.

- Optical Parameters for the Breast and Tumor:

The optical properties of the breast and tumor tissues are defined in the mua_bkg, mus_bkg, and ref_bkg variables. Users can adjust these values to simulate different optical characteristics for the breast. Additionally, specific optical properties are assigned to tumor nodes, like mua (tumor_nodes_idx) and mus (tumor_nodes_idx), allowing users to model variations in the tumor's optical properties.

By adjusting these parameters within the provided MATLAB code, users can create diverse scenarios for DOT scans, exploring different configurations of source-detector pairs, breast mesh scales, tumor characteristics, and optical properties.