breastCompress Documentation

Release 1.0

Christian G. Graff

Dec 17, 2018

Contents:

Welcome to the VICTRE breastCompress software documentation!

This documentation provides information regarding how to download, install, and use the VICTRE breastCompress software which models the physical compression that breasts undergo during mammographic imaging using a finite-element based linear elasticity solid mechanics simulation. It is designed to be used with anthropomorphic voxelized whole-breast phantoms generated by the VICTRE breast phantom software. This software was developed as part of the Virtual Imaging Clinical Trials for Regulatory Evaluation (VICTRE) project at the U. S. Food and Drug Administration.

The VICTRE tools provide a complete simulated imaging chain for mammography and digital breast tomosynthesis and are available as open source software

CHAPTER 1

Overview

The breastCompress code uses finite element solid mechanics to model the compression of a breast between two planar surfaces. The general procedure is as follows:

1. Load uncompressed voxelized breast phantom file
2. Create tetrahedral mesh for breast volume

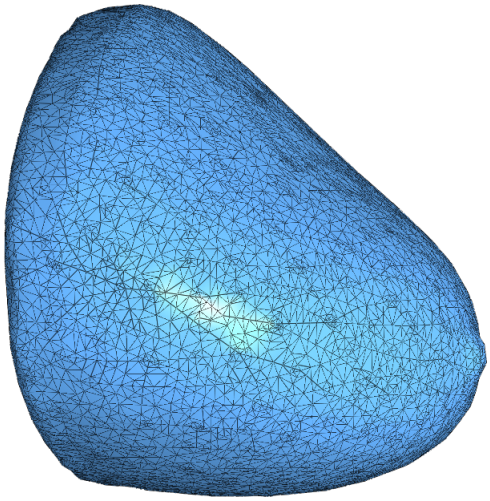

Fig. 1: Tetrahedral mesh

3. Assign linear elasticity properties (Poisson's ratio and Young's modulus) for each tetrahedron. Currently only glandular or fat elasticity parameters are assigned. Tetrahedrons containing muscle serve as a fixed boundary condtion during the compression similation.
4. Virtual compression plates are generated and placed above and below the breast model. EXPERIMENTAL: These plates can be rotated an arbitrary angle to simulate different compressed views.

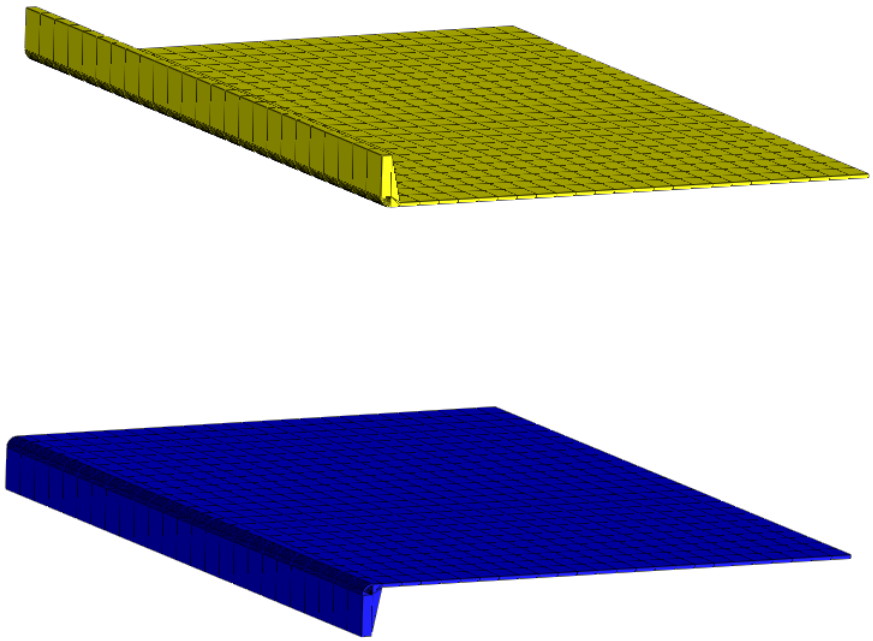

Fig. 2: Paddle mesh design

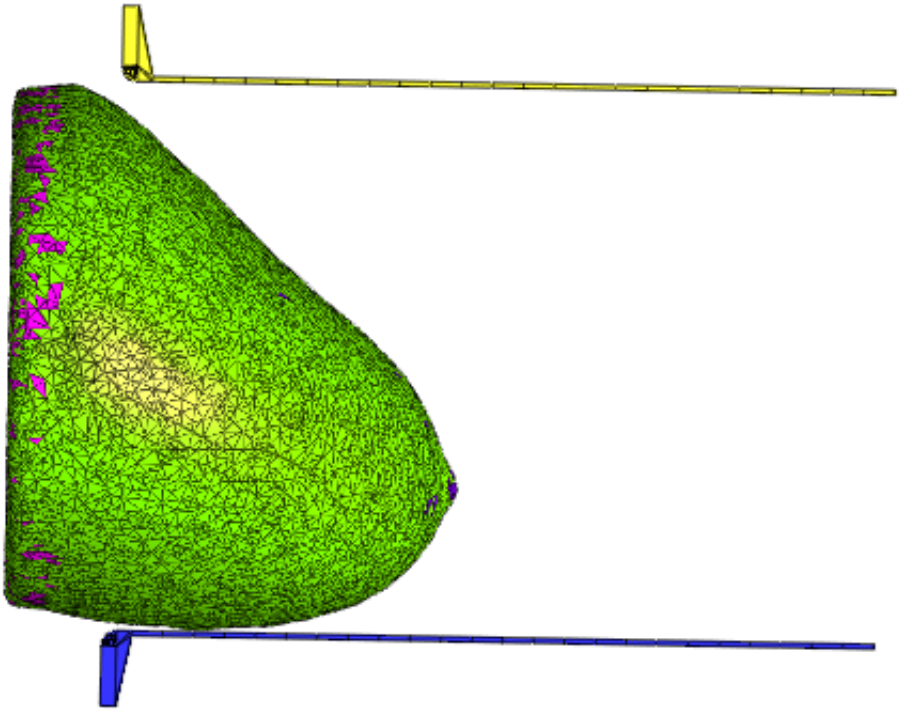

Fig. 3: Initial paddle position

5. An XML FeBiO input file is generated based on the phantom geometry and passed to a spawned FeBiO process that performs the compression simulation.
6. Compression plates are moved towards each other until the desired compressed thickness is achieved.

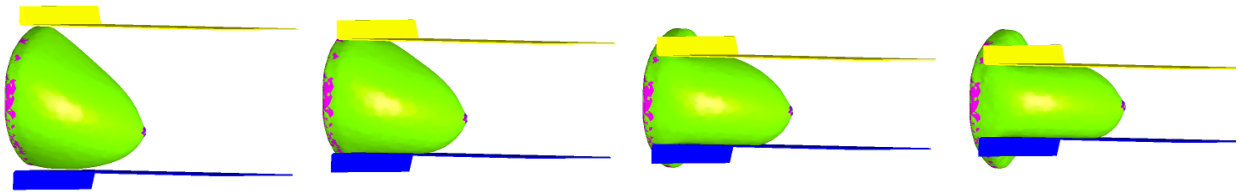

Fig. 4: Compression progression as plates are moved together

7. The obtained displacement maps at each finite element node are interpolated to determine where each voxel in the uncompressed breast moved to during compression and a new voxelized compressed breast phantom, including the compression paddles is generated and saved in the output file.

Note: If the compression simulation fails to converge at the desired compressed thickness, intermediate voxelized phantoms are saved to disk and the current breast shape is re-meshed. Depending on the breast shape, position of the paddles, and desired thickness, multple remeshing steps may be required.

CHAPTER 2

Installation

The breastCompress source code is available here. It is written in C++ and therefore must be compiled for your system before use.

2.1 Operating System

This software has been developed and tested on Debian and Red Hat based Linux distributions. Compilation is via cmake and should be possible on any UNIX-like operationg system (including MacOS) with minimal modification to the build configuration in the provided CMakeLists.txt file, however the following distributions are recommended:

- Ubuntu 16.04
- Centos 7.x

The following build instructions are based on Ubuntu 16.04.

2.2 Prerequisites

- FEBiO (required)

  To help perform the breast compression simulation this software relies upon elements of the FEBiO Software Suite, a nonlinear finite element analysis software package. FEBiO is not distributed as part of breastCompress and must be downloaded from the FEBiO website. Registration is required but FEBiO is open source software and free for non-commerical use.

- PostView (optional)

  PostView is another component of the FEBiO Sofware Suite which can be used to visualize the compression process. This is not required to use breastCompress, but may be useful for debugging compression problems or producing graphics, strain maps, displacement maps, etc.

2.3 Compilation

First, install the packages necessary to compile the software (reminder: instructions are specific to Ubuntu, consult your operating system documentation for analagous package names and installation procedures) From a terminal command prompt:

```
> sudo apt-get install cmake vtk6 libvtk6-dev \
                       libblas-dev liblapack-dev \
                       libboost-dev libboost-program-options-dev \
                       libproj-dev libtet1.5-dev zlib1g-dev
```

Next, clone the git repository:

```
> git clone https://github.com/DIDSR/breastCompress
```

From a suitable build directory generate a Makefile using cmake and run make:

```
> cmake <path to repository>
> make
```

where <path to repository> is the directory containing the cloned repository. .. note:: When running cmake you may recieve an error message related to references to files "/usr/lib/x86_64-linux-gnu/libvtkRenderingPythonTkWidgets.so" and "/usr/bin/vtk".

> This is a known Ubuntu VTK bug but since we are not linking to these files, the error can safely be ignored.

2.4 FeBiO Setup

To function properly breastCompress needs to know where the FeBiO executable is installed on your system. This information can be passed to FeBiO as a command line option (see next section) or be specified in the environment variable $FEBIO_PATH. If using the bash shell this variable can be set using:

```
> export FEBIO_PATH=<path>
```

for example:

```
> export FEBIO_PATH=/usr/local/febio-2.3.1
```

breastCompress will expect the FeBiO executable to be located at $FEBIO_PATH/bin/febio2.lnx64.

2.5 Finish

If the previous steps were successful, the executable breastPhantom should be present in the build directory. This file can be copied to a convenient location in your path, or the build directory can be added to $PATH. No additional installation steps are required. Consult the next section which describes how to run the executable.

CHAPTER 3

Running the Software

3.1 Command line options

breastCompress is run from the command line and requires that several command line options be specified:

```
> breastCompress -s <SEED> -t <THICKNESS> -a [ANGLE] -d [DIRECTORY] -f [FEBIOPATH]
```

required command line arguments are

-s SEED the seed number of the uncompressed phantom you wish to compress

-t THICKNESS the desired compressed breast thickness in millimeters

optional command line arguments are

-a ANGLE EXPERIMENTAL: the angle of the compression direction in degrees relative to the transverse plane. The default value is 0 which provides cranial caudal (CC) compression.

-d DIRECTORY the directory containing the uncompressed breast phantom. The output compressed breast phantom will be placed in this directory as well. The default value is '.' representing the current working directory.

-f FEBIOPATH the directory in which FeBio has been installed. This is only optional if environment variable $FEBIO_PATH has not been set (see previous section).

3.2 Example

To compress the phantom p_123456789.raw.gz contained in the directory /tmp with a 15 degree paddle angle to a thickness of 40 mm with FeBiO installation directory /usr/local/bin/febio-2.3.1 (FeBiO application location /usr/local/bin/febio-2.3.1/bin/febio2.lnx64), issue the following command:

```
> breastCompress -s 123456789 -t 40.0 -a 15.0 -d /tmp -f /usr/local/bin/febio-2.3.1
```

CHAPTER 4

File Formats

breastCompress requires several input files that characterize the uncompressed breast phantom. Several types of intermediate temporary files are generated during the compression process in addition to the final compressed breast phantom output files. In the following filename descriptions *nnnnnnnn* represents the seed number specified as a command line option to breastCompress.

4.1 Input Files

The breastCompress input files are created using breastPhantom. breastCompress expects the following files to be present in the input directory specified on the command line:

p_*nnnnnnnn*.raw.gz The raw phantom volume stored as 8-bit unsigned integers in a gzip archive. There is no file header.

p_*nnnnnnnn*.mhd A metaImage header file containing information about the phantom data stored in the file p_*nnnnnnnn*.raw.gz

p_*nnnnnnnn*.loc This file lists candidate locations for insertion of masses and calcifications. Each location is where a TDLU has been randomly generated.

4.2 Intermediate Files

These are temporary files generated during the compression process. They are not automatically deleted once the breast compression is complete. Compression is an iterative process and depending on the desired compressed thickness and breast shape, several remeshing steps may be required. Each remeshing step will create new intermediate files.

febio_*nnnnnnnn_m*.feb This is an XML file generated by breastCompress that is passed as input to FeBiO. A new file will be generated for each remeshing step. Minimally m=0 will be present.

febio_*nnnnnnnn_m*.log This is a log file generated by FeBiO. It contains information regarding the convergence of the finite element compression procedure.

febio_*nnnnnnnn_m*.xplt This file contains the results of the FeBiO compression process. The compression process and produced stress/strain and displacement maps can be visualized by opening this file with PostView, which is available for download from the FeBiO website.

p_*nnnnnnnn_m*.raw.gz Intermediate voxelization of the breast phantom during compression at remeshing step *m*

p_*nnnnnnnn_m*.mhd A metaImage header file for the intermediate voxelization contained in the associated .raw.gz file

p_ *nnnnnnnn_m*.loc Intermediate TDLU locations

4.3 Output Files

Final output file have formats indentical to the input files with prefix *pc_* to designate they represent a compressed phantom

pc_*nnnnnnnn*.raw.gz The raw compressed phantom volume stored as 8-bit unsigned integers in a gzip archive.

pc_*nnnnnnnn*.mhd A metaImage header file containing information about the phantom data stored in the file pc_*nnnnnnnn*.raw.gz

pc_*nnnnnnnn*.loc Final post-compression TDLU locations.

CHAPTER 5

Notes

- Both breastCompress and FeBiO are multi-threaded applications which respect the environment variable $OMP_NUM_THREADS
- Currently for the purposes of compression modeling all tissue is treated as either fat, glandular, or muscle
- Several parameters that users may want to adjust cannot be specified as a command line option and must be changed in the source code file breastCompress.cxx. Some of these parameters include
  - Elasticity parameters
    - *fatModulus* - Young's Modulus for fat tissue, default value = 5 kPa
    - *fatPoisson* - Poisson's Ratio for fat tissue, default value = 0.49
    - *fatDensity* - density of fat tissue, default value = 0.000001 kg/mm^3
    - *glandModulus* - Young's Modulus for glandular tissue, default value = 15.0 kPa
    - *glandPoisson* - Poisson's Ratio for glandular tissue, default value = 0.49
    - *glandDensity* - density of glandular tissue, default value = 0.000001 kg/mm^3
  - Paddle geometry
    - *paddleHeight* - thickness of paddle, default value = 1 mm
  - Tetgen parameters - the tetrahedral mesh is generated using the Tetgen library. Mesh parameters are specified in a tetgenbehavior struct. See the TetGet documentatio for more details http://wias-berlin.de/software/tetgen/1.5/doc/manual/index.html
  - FeBiO parameters - all finite element parameters are written to the FeBiO XML input file automatically by the code. To change these parameters consult the FeBiO documentation available at https://febio.org/support/ and the relevant sections of breastCompress.cxx.