Generating Fabric Tensors from ETHZ Data

June 16, 2017

From:

Andrew Stershic, Sandia National Laboratories

To:

Abishek Kumar, Oak Ridge National Laboratories

Srdjan Simunovic, Oak Ridge National Laboratories

In this document, I describe the methods used to develop contact-network fabric tensors from the ETHZ tomography data, as performed in the study published in *Journal of Power Sources*: “Modeling the evolution of lithium-ion particle contact distributions using a fabric tensor approach” (2015).

There are two stages in this analysis:

1. Describing the contact network, namely creating a list of contact vectors
2. Generating fabric tensors representing the contact network.

Several methods are used to describe the contact network from the tomography, including several geometric approximations:

1. Spherical
2. Ellipsoidal
   1. Centroid vector / branch vector
   2. Contact normal
3. Voxel-based
   1. Centroid vector / branch vector
   2. Contact normal

Notes on generating each of these contact networks are given below. The scripts used in the FT generation are posted at: <https://github.com/andybond13/ETHZ_to_FT> .

1.a. For the spherical approximation, the idea is to use ETHZ’s labeled particle statistics files (list of centroid, volume, etc.) to approximate each particle as a sphere (radius = (volume\*3/4)^(1/3) ). Then, contacts are found by computing the distance between every possible pair of spheres and seeing if that’s less than the sum of the radii. I could not find the original file that I had to calculate spherical contacts, so I wrote a new one.

* sph\_approx/sphereContact.py

1.b. For the ellipsoidal approximation, the idea is to use ETHZ’s labeled tomography to mesh each particle, smooth the mesh, find the convex hull of the mesh vertices, and then fit an ellipsoid to these vertices. The contacts are found using an ellipsoid contact finding algorithm. There’s no easy solution for finding ellipsoidal contact. The algorithm works by shrinking or expanding the two ellipsoids by the same volume percentage until they are just tangent. If they must be shrunk, then they are contacting, and the point of tangency when reduced forms the contact vector. The centroid/branch vector is also recorded.

* fixtif.sh (run to correct the \*.tiff tomography)
* ell\_approx/tomography\_to\_simplified\_mesh/Makefile(\_2) (run to generate mesh files)
* ell\_approx/fit\_script.m (run to generate ellipsoids, search for contact, and record contact vector)
* ell\_approx/fit\_script\_centroid.m (run afterward to generate centroid vector)

1.c. For the voxel approximation, the idea is to use ETHZ’s labeled tomography to check if any adjacent voxels (infinity norm) are of different particles. The average contact vector (either the average of all adjacent voxel-voxel vectors, or vector between centroid of group of contacting voxels of each particle, same thing) as well as the centroid/branch vector are recorded. I also wrote code to record clusters of contacts (each pair of contacts might touch more than once), though I don’t remember it being a noticeable difference.

* scanStackRangeContact.pl (process images to make list of contacting pairs)
* cutToPairs.py (takes output above and cuts it to only the contact pairs)
* main.m, main\_cluster.m (creates contact vector list, and I think calculates contact area as well)

2. Fabric tensors are generated using the MMTensor package (<https://es.mathworks.com/matlabcentral/fileexchange/32891-mmtensor-1-0>). I wrote a few scripts to automate launching the fabric tensor generation. I generate the second-order fabric tensor “F2” and the fourth-order fabric tensor “F4” and use the highest order possible that passes the statistical significance test. Depending on application (i.e. mechanical contact, thermal contact), I might recommend using a version weighted by contact area, since that will more accurately reflect the transport ability.

* fabricTensor.m (this is the file I’ve configured to run the analysis and performs the FT comparisons; since this is where the tensors are computed, this would be the place to save them to file)
* fabricTensor\_all.m (this file I used to run all the comparisons)
* fabricTensor\_all\_withArea.m (same thing but weighting by contact area)

Feel free to send me questions on these. It has been a few years since I have used these scripts, so they’re a little fuzzy to me, but I think I described them adequately. I moved the files around in the version I packaged for you, so you might have to check all file-path locations in scripts.

Andy Stershic