



CTViz: A tool for the visualization of transport in nanocomposites



Benjamin Beach^{a,1}, Joshua Brown^{a,2}, Taylor Tarlton^a, Pedro A. Derosa^{a,b,*}

^a Louisiana Tech University, Physics Program, Ruston, LA 71272, United States

^b Grambling State University, Department of Mathematics and Physics, Grambling, LA 71245, United States

ARTICLE INFO

Article history:

Received 12 June 2015

Received in revised form 5 March 2016

Accepted 24 March 2016

Available online 25 March 2016

Keywords:

Visualization

Nanocomposites

Transport visualization

ABSTRACT

A visualization tool (CTViz) for charge transport processes in 3-D hybrid materials (nanocomposites) was developed, inspired by the need for a graphical application to assist in code debugging and data presentation of an existing in-house code. As the simulation code grew, troubleshooting problems grew increasingly difficult without an effective way to visualize 3-D samples and charge transport in those samples. CTViz is able to produce publication and presentation quality visuals of the simulation box, as well as static and animated visuals of the paths of individual carriers through the sample. CTViz was designed to provide a high degree of flexibility in the visualization of the data. A feature that characterizes this tool is the use of shade and transparency levels to highlight important details in the morphology or in the transport paths by hiding or dimming elements of little relevance to the current view. This is fundamental for the visualization of 3-D systems with complex structures. The code presented here provides these required capabilities, but has gone beyond the original design and could be used as is or easily adapted for the visualization of other particulate transport where transport occurs on discrete paths.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

The idea of embedding nanostructures in a continuous material has attracted a great deal of attention, as the resulting composites have interesting and potentially unprecedented properties [1–12]. The addition of fillers with mechanical, electrical, or thermal properties different from those of the matrix, such as hard fillers in soft matrices or conductive fillers in insulators, has been sought as a way to improve the properties of the matrix.

Embedding good conductors into a non-conductive matrix results in larger conductivity than that of the bare matrix, as charges no longer have to bridge the full gap between contacts, but only between neighboring inserts [13–18]. Conduction arises when percolation paths are formed at sufficiently high nanoinert concentrations [13–15,19–24]. A hopping model was developed and implemented into a computer code where carriers are injected at one end of the sample and tracked as they travel through an arrangement of nanotubes (NTs), hopping from NT to NT through a non-conductive matrix [25]. The model does so in a 3-D array where

thousands of NTs are distributed in a variety of configurations. As the code became more complex and the samples to model became larger, the task of making modifications to and finding errors in the code became increasingly challenging, thus triggering the creation of the visualization code that is the subject of this paper.

CTViz was developed to produce visuals of the data output from a 3-D model of a nanocomposite, including the simulation box, stationary visualization of carriers' paths, and animated carriers' trajectories. An important feature of CTViz is the use of shades and transparency to highlight critical information. CTViz was designed for three main tasks: to help debug an existing simulation code, to help with data analysis of the transport processes in the sample, and to help communicate results with high quality visuals. Users have many options for tailoring the visualization capabilities in both stationary images and animations with high quality visuals and great flexibility on visualization modes.

2. Related work

Nanotechnology has pushed research interest towards the atomic scale and larger computer power has triggered an increase in popularity of bottom up approaches to materials modeling. The size and complexity of molecular and coarse-grained systems that can be modeled has grown to the point that visual aids are an unavoidable requirement; as such, several tools have been pro-

* Corresponding author at: Louisiana Tech University, Physics Program, Ruston, LA 71272, and Grambling State University, Grambling, LA 71245, United States.

E-mail address: pderosa@latech.edu (P.A. Derosa).

¹ Current affiliation: Virginia Tech, United States.

² Current affiliation: The University of Colorado, United States.

duced for the purpose of molecular visualization. With no illusion of being fully inclusive, some examples of available visualization tools, which focus mainly on molecular and/or coarse-grained samples, are discussed below.

Several visualization tools exist as a part of, or a complement to, quantum chemistry simulation software. Example of molecular visualization tools include EyeChem [26], Chimera [27], GaussView [28], Spartan [29], and Arguslab [30]. Ptuba [31], VMD [32], BioVEC [33] are or include visualization tools for biosystems. Materials Studio also falls within this category, although it provides a platform to a number of simulation tools [34]. Visualization tools are also important in the simulation of solids, crystalline and amorphous, as they allow the predictions from the simulations to be conveyed much more effectively. MaterialVis [35] and CrystalMaker [36] are just two examples.

Other properties and processes involving large systems, such as fluid dynamics, heat transport, and mechanical performance, also benefit from graphical tools. Two of those tools that share some of the visualization capabilities of CTViz are VisFluid [37] and VisIt [38]. VisFluid was created to model the fluid flow, a continuous media, in porous materials and uses a simulation engine based on a dynamic percolation method. VisFluid can generate both images and animations of the 3D data and integrates the simulation with the visualization. Similar to CTViz, it makes use of both color and transparency to enhance the visualization of its information by making important regions of the system more visible. VisIt is an open-source general-purpose visualization, animation, and analysis tool that was originally developed to visualize and analyze the results of terascale simulations. VisIt is mostly based on finite elements, but also supports the visualization of scalar and vector field data for 2-D and 3-D structures. VisIt is highly parallelizable, capable of operating on massive systems with thousands of cores, or on a regular desktop workstation. It can produce and save presentation-quality visuals and animations of the simulation data. It was designed with a high degree of modularity featuring plugin architecture for custom readers, data operators, plots, and user interfaces. Users can also develop their own additional plugins to meet their needs; however, as a general purpose tool, specific capabilities for particular applications, such as the one that motivated CTViz, are not immediately available. VisIt was developed at Lawrence Livermore National Lab where many of other software for visualization or with great visual capabilities, were developed. The reader is directed to the national lab web page for a list and description of these software [39].

3. Key features of CTViz

3.1. General features

The simulation box that CTViz visualizes consists of a distribution of cylinders originally intended to represent NTs. Each NT is a sequence of consecutive segments, each defined by a line connecting two points. The coordinates of each point is stored in the coordinate file and are used to generate a map of the NTs in the composite.

The NTs in the sample are plotted in a box and all segments outside the box (Fig. 1a), if they exist, can be hidden for a cleaner view (Fig. 1b). A pair of electrodes can then be drawn at each end of the sample by using a parallel view paradigm (Fig. 1c). Additionally, the plot can be cross-sectioned in any of the three axial directions for a layered view of the sample (Fig. 1d–f). The user can elect to see all NT segments that lie between two values of any of the three coordinates (Fig. 1d and e), or to exclude all NT segments in that range (Fig. 1f). This last feature, hiding a user-defined region, is not commonly available in existing software (although the difference

is subtle, most codes do allow users to select what they want to see and hide the rest). Though it is possible in some software, like VMD, it normally requires custom scripts to execute, but in CTViz, it can be done with relative ease from the control panel. These options are available in all graphical modes, except the animation mode.

One other motivation for this code was to obtain presentation and publication quality visuals. As such, this code was designed to give the user flexibility with the aesthetics. A list of these features is listed below.

In all modes, the user can:

- Choose between black and white for the background color
- Insert or hide electrodes and control their thickness and opacity
- Draw or hide the bounds of the simulations box
- Switch between parallel and perspective views
- Add index labels, with adjustable font size, to individual NTs in the sample to assist with debugging or to pinpoint particular features
- View the sample from any of the +x, +y, and +z directions, or the default viewing angle, with the click of a button
- Dynamically rotate, zoom, and pan the sample
- Control the visual diameter of the nanotubes as a factor of the diameter specified in the file (1 is the actual size and is the default value)

In certain modes, the user can:

- Cross-section the sample in any combination of the three axial directions (does not work with the animation module)
- Hide NT segments located outside the designated box (Works in all modes except the animation module)
- Control the opacity range of NTs visited by carriers (works in modes where charge transport is visualized)

3.1.1. File I/O

CTViz can use 6 different files to generate its plots and animations. Those are the same files that the in-house code that originated this tool either needs as input or produces as output. These files include two configuration files, the *coordinates* and the *neighbor* files, and four output files, *percolation*, *trajectory*, *traps*, and *frequency* files.

In the in-house simulation code that CTViz serves, each nanotube is generated as a set of segments defined by the coordinate of the two end points. These coordinates are stored in the coordinate file with extension “.cor” and read by CTViz to generate a visual of the sample. When carriers progress through the nanocomposite, they do so by hopping from one NT to one of its closest neighbors. A separate file with “.nei” extension, the *neighbor* file, stores for each NT the list of neighboring NTs, the distances to them, and the coordinates of the points defining those distances. An example of these two files is provided as supplementary information.

The frequency file, with extension “.freq”, contains for each NT the number of times that it has been visited by a carrier. A file where percolation paths are stored (with extension.perc) contains only the NTs that are part of a completed percolation path. Further information is stored in the trajectory file (extension “.path”) where the trajectory of all carriers that reached the collection electrode, along with the distance of each hop and the amount of time required to make each hop, are stored. The same type of information is stored in the *trap* file (extension “.trap”) for those carriers that did not reach the opposite electrode (and are thus considered trapped). Not all these files have to be available, only those that are required for the current visualization. The program will even reproduce the information in the *frequency* file from the *path* and *trap* files if not present, although it requires the user to manually select carrier to account for. Nevertheless, having the *frequency* file available makes

Download English Version:

<https://daneshyari.com/en/article/443272>

Download Persian Version:

<https://daneshyari.com/article/443272>

[Daneshyari.com](https://daneshyari.com)