EXPLORING STIRRED TANK CFD SIMULATION DATA VIA TANGIBLE INTERACTION WITH MECHATRONIC INTERFACE COUPLED TO CAVE-BASED VISUALIZATION

Kazim Sekeroglu[1], Sanjay Kodiyalam[2], Brygg Ullmer[3,4], Rajesh Sankaran[3,5], Werner Benger[3],
Somnath Roy[6], Alvin Wallace, Jr.[3], Sumanta Acharya[6] , Amitava Jana[2]
Department of Electrical[1]/ Mechanical[2] Engineering, Southern University
Center for Computation and Technology[3]; Dept. of Computer Science[4]; Dept. of Electrical Engineering[5]; Dept. of Mechanical Engineering[6], Louisiana State University
ksekeroglu@gmail.com, sanjaykodiyalam@engr.subr.edu, ullmer@cct.lsu.edu,
rajesh@cct.lsu.edu, werner@cct.lsu.edu, sroy13@tigers.lsu.edu, awall14@tigers.lsu.edu,
acharya@me.lsu.edu, jana@engr.subr.edu,

Abstract

Computational Fluid Dynamics (CFD) is being used by the chemical/pharmaceutical industry for designing efficient stirred tank systems to mix and react materials. With recent advances in parallel computing and the ability to resolve the flow scales, direct simulations from CFD are generating terabytes of simulated data with a wealth of physics embedded in the spatially and temporally resolved simulation. A key impediment is in the analysis of the large data sets, and tools that are able to represent the essential physics. Towards this end, we envision creating two synchronized stirred tank mechatronic interfaces representing the physical model of the stirred tank with a linear probe representing an emitter of streamlines/pathlines in the coupled CAVE (Cave Automatic Virtual Environment)-based visualization of the CFD data. One of the mechatronic interfaces will be inside the CAVE together with a controlling tangible interface; another will be outside of the CAVE, together with a screen-based display of the same CFD dataset. This setup is expected to handle a large number of users – as in a class of students – that are interested in exploring the dataset. Adopting memory management ideas from previous work [7] it is demonstrated that visualization of streamlines/pathlines in the data with total size beyond the available RAM is possible. These tools will enable a better understanding of the stirred tank flowfields and facilitate the process of improved designs.

Keywords: Computational Fluid Dynamics, Stirred tank, Mechatronic interface, Tangible device, Visualization, CAVE.

1. Introduction

CFD is widely used in industry for prediction, design, and operation. Stirred tanks (Fig.1a) are used in the process industry to mix materials together under a variety of thermal and mechanical conditions[1]. CDF is being used for designing efficient stirred tank models. It is more cost-effective than physical testing and provides more complete information than is possible via testing[2]. Meshes used in CFD are dense and irregular to account for the circular geometry of the stirred tank with regions corresponding to baffles eliminated from the computational space (Figs. 1b & 1c).

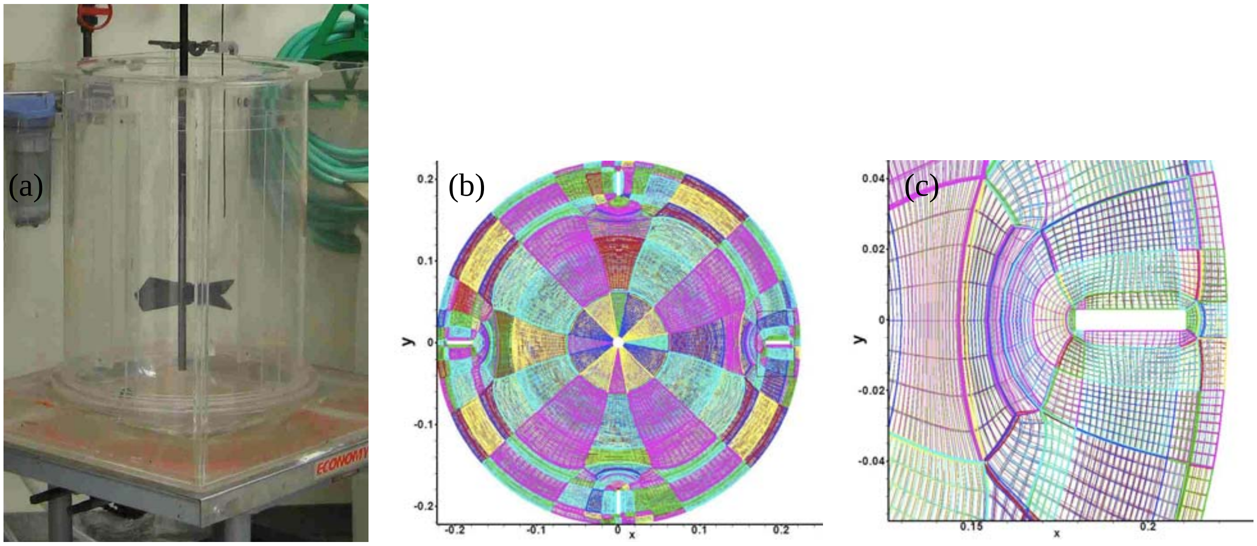

Fig. 1. (a) Stirred Tank Geometry (courtesy: Dow Chemicals). (b) Top view of the finite volume mesh used in CFD[5]. The mesh is composed of 2,088 blocks each of which has nodes along a curvilinear coordinate system. (c) The mesh is complex around the baffles as these regions are outside of the computational space.

One of the ways of exploring CFD data is in a virtual reality (VR) environment. The purpose of VR is to immerse the users in the simulated world and allow them to interact with it in an intuitive way with the goal of enabling users to experience performing tasks in the virtual world[3]. The CAVE at Southern University (Fig. 2) is a projection-based VR system that surrounds the viewer with four displays that are arranged to make a cube with three screen-walls that are rear-projected together with down-projection onto the floor. A user wears stereo shutter glasses with a six-degrees-of-freedom head-tracking device based on which the correct stereoscopic perspective projections are calculated for each display as the user moves inside the CAVE. A second sensor and buttons in a wand held by the viewer provide interaction with the virtual environment[4].

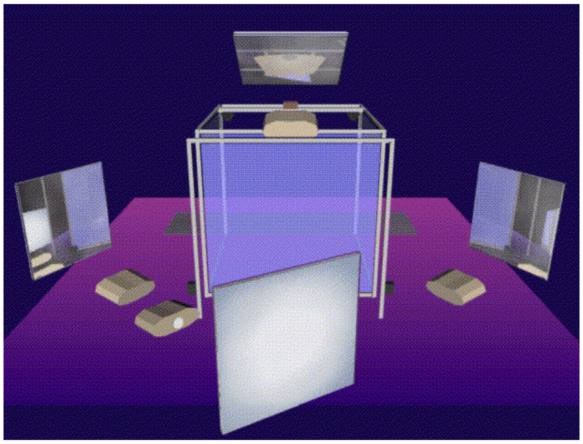

Fig. 2. Schematic of the CAVE at Southern University[6].

In a CAVE the number of users who can simultaneously explore the data is restricted as the displayed perspective projections are exact only for the user wearing the glasses that has the head-tracking device. In order to introduce the physical model to a larger group – such as a class of students - we envision creating two synchronized stirred tank mechatronic interfaces representing the stirred tank with a linear probe representing an emitter of streamlines/pathlines in the coupled CAVE-based visualization of the CFD data. One of the mechatronic interfaces will be inside the CAVE together with a controlling tangible interface; another will be outside of the CAVE, together with a screen-based display of the same CFD dataset. This setup is expected to enable multiple users to have an understanding of the physical model before individually exploring the dataset in the CAVE. The visualization code enables exploration of the large dataset whose total size is larger than the available RAM on the machine driving the displays.

2. Design and Construction: Stirred tank mechatronic interface & Tangible controller

Mechatronic interfaces to provide an interactive means for physically engaging with the CFD dataset from within and outside of the CAVE, in contexts where participants are in an ‘eyes-busy’ mode of engagement, are being developed. The interfaces are an ongoing effort and Fig. 3a illustrates their current design. The interface is a physical model of the stirred tank together with a vertical mast representing an emitter of streamlines/pathlines in the coupled visualization of the stirred tank CFD data. The full dataset can be explored by varying the angular and radial position of this mast. This is accomplished with two motors: Motor-1 drives the angular position, and Motor-2 drives the radial position of the vertical mast. The angular and radial position of the mast represented by the screw at the end of the green line in Fig. 3, is controlled using a closed loop system comprising of the motors, shaft encoders, linear potentiometers and reed limit switches. DC motors and stepper motors are being considered in the current design iterations. A tangible controller is being developed for controlling the mechatronic interfaces and simultaneously steering the visualization in the CAVE.

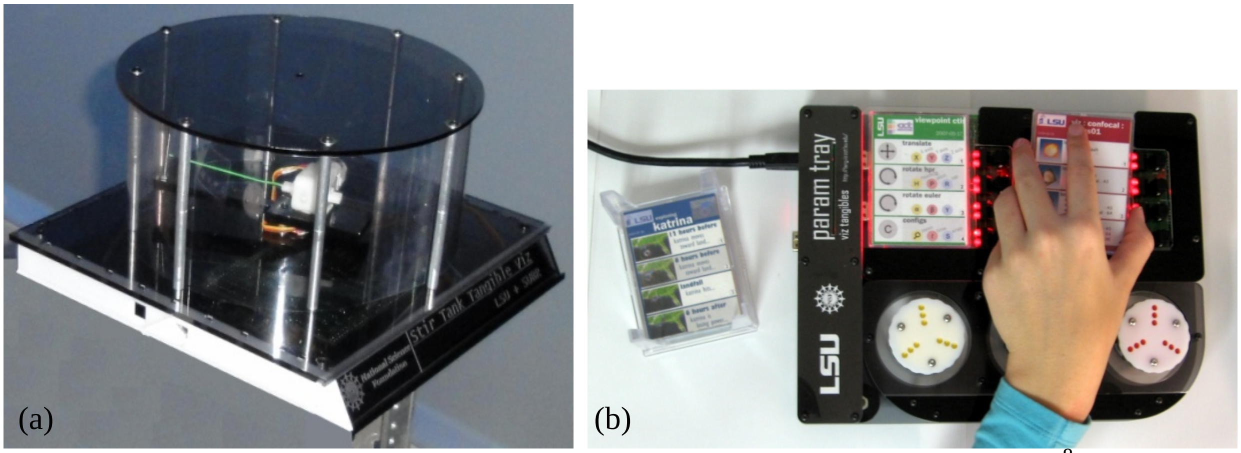

Fig. 3. (a) The Mechatronic interface, and (b) The Tangible controller[8].

Fig. 3b shows one embodiment of tangible controller. The position of the vertical mast in the mechatronic interfaces is made to coincide with a vertical line-emitter of streamlines/pathlines in the visualized data. The underlying electronics in the mechatronic interfaces and the parameter tray will be implemented using Blades and Tiles[8]. Blades and Tiles are modular and flexible electronic building blocks aimed at facilitating the design and development of a variety of tangible and embedded interfaces in a decoupled fashion. The parameter tray and the mechatronic interfaces will interact with the visualized data in the CAVE through their respective embedded Linux gateways. The electronic hardware for the controller has been realized and we are currently focused on the mechatronic interface and associate software.

3. Visualization application

A Visual Studio based C/C++ project, using the CAVE Library 3.2, is developed for visualizing streamlines/pathlines in the finite volume based CFD data of the stirred tank. To enable quick exploration of the large stirred tank CFD dataset (~2100 curvilinear blocks with a total of ~2.3 million nodes at which fluid velocity (3D) and pressure are specified for ~30,000 time steps) the following steps, inspired by algorithms used in classical atomistic simulations and exploiting the time independence of the CFD mesh, are implemented:

3a. Interpolation of nodal velocity based on node-specific spherical cutoffs:

While the stirred tank CFD data is output only at nodal positions streamline and pathline computation requires the fluid velocity to be defined over all of space. Therefore a scheme to interpolate the velocity from nodal values is constructed with each node having a spherical region of influence beyond which the particular nodal value does not contribute to the interpolated velocity. This cutoff distance is chosen to be the maximum distance between the node of interest and any of its neighbors in the curvilinear blocks – inclusive of neighbors along the body diagonal of the finite volumes defined by this node. The interpolated velocity at a point of interest is chosen to be a weighted sum of all the nodal values corresponding to nodes that are within their respective cutoffs from the point of interest. The distance based weight factor is zero beyond the cutoff distance and has zero first and second spatial derivatives at the cutoff so that the same derivatives are continuous for the interpolated velocity. In addition, the weight factors are normalized such that the interpolated velocity is equal to the nodal velocity at every nodal position.

3**b**. Reorganizing nodes as belonging to a uniform mesh of cubic regions:

In order to minimize the search over nodes that may have a non-zero weight for the interpolated velocity at a particular point the nodes are organized as belonging to cubic regions that span a cuboidal region of space containing the stirred tank. This is analogous to earlier work mapping curvilinear block cells to a uniform grid[9]. The search for nodes that can contribute to the interpolated velocity at the point of interest can then be restricted to a few such cubic regions neighboring the point of interest. For any point in a given region, all the neighboring regions that can possibly contribute to the interpolated velocity at the point are determined and listed as neighbors of the given region. This is done for all the regions of the regular mesh. As a first attempt the size of the regions is chosen to be the root-mean-squared value of the nodal cutoffs.

3c. Reorganized nodal data written to files in binary format:

To access data rapidly during the computation of streamlines/pathlines the reorganized nodal data is written to binary files. Two files are constructed. The first has nodal cutoff and position information organized as in **3b** together with the list of neighbors for all regions. While this is constructed only for the first CFD time step it is valid for any time step due to the time independence of the original CFD mesh. The second file has the nodal velocity and pressure information organized in exact correspondence to the information sequence in the first file – with data corresponding to the CFD time steps written out in sequence. These file formats are analogous to the F5 format for a uniform grid[10].

3d. Computation and display of streamlines and pathlines:

The previous steps (**3a-3c**) are completed before exploring the stirred tank dataset in the CAVE. The position of the linear emitter of streamlines/pathlines is currently determined by user input via the joystick that is also used for navigation: Translation, rotation, and scaling of the display of the curvilinear blocks with streamlines/pathlines. Streamlines are computed by integrating the interpolated velocity corresponding to a single CFD time step of interest while pathlines are computed by integration across the CFD time steps of interest. All of the nodal cutoff, position and required neighborhood information for these computations (first file in **3c**), that is common to all of the CFD time steps, is read in at startup. Velocity data corresponding to the each of the CFD time steps of interest (second file in **3c**) may be loaded at startup for a limited number for CFD time steps or read in dynamically[7] – with memory allocated for only the data read in / required for the computations. Both streamline and pathline computations use the 4th order Runge Kutta integrator with a step size equal to the CFD simulation time step– with pathlines composed from small streamlines corresponding to single time steps. As the CAVE library spawns a new thread for each of the displays, the main thread is used for these computations to allow for the display threads to execute at the standard rate of 60 frames per second. For rapid execution of the displays, the same display list is compiled in each of the display threads as required by the CAVE library at the end of every new streamline/pathline computation.

Readin time for nodal cutoff, position and neighborhood information (131 MB) ~0.4 sec Readin time for nodal velocity and pressure information (69 MB/CFD time step) ~0.16 sec			
Objects	Velocity data read in:	Compute time (sec)	Display list compile time (sec)
1,000-step streamlines	Dynamically	2.68 ± 0.67	0.169 ± 0.016
1,000-step streamlines	At start up	2.76 ± 0.76	0.167 ± 0.017
1,000-step pathlines	Dynamically	4.85 ± 0.60	0.166 ± 0.013
1,000-step pathlines	At start up*	2.44 ± 0.45	0.177 ± 0.013
20,000-step pathlines	Dynamically	82 ± 10	6.26 ± 0.85

Table 1. Timing for various operations on the display-machine in the CAVE. The mean and standard deviation for 10 streamlines/pathlines are emitted from the vertical line emitter at 34 different locations in the stirred tank is shown. *See text for actual amount of data loaded.

Table. 1 shows the time required for various operations as measured on the machine driving the displays in the CAVE: an Opteron 4-way SMP with 64 GB of RAM. These tests use only 100 CFD time steps converted to the format in 3c. However, the tests are carried out for a larger number of steps as streamlines require data from only single time steps while pathlines are computed by repeated cyclic read in of the 100 time steps as necessary. Pathlines with data read in at start up correspond to reading in 700 time steps followed by cyclic referencing of the same for time steps beyond 700. The timing data shows that there are no distinctions between the cases of streamline computations and pathline computations with read in data. This is consistent with the small time to read in the entire velocity data for a single CFD time step. While pathline computation with dynamic read in of data is slower that other cases by a factor of ~2, it illustrates that data with total size (1000 x 69 MB ~ 67 GB) larger than the available RAM (64 GB) can be explored via dynamic allocation of memory for only the read in / required data. The time required for pathline computation with dynamic read in of data scales roughly linearly up to 20,000 time steps demonstrating that any subset of up to 20,000 CFD time steps can be explored using this application.

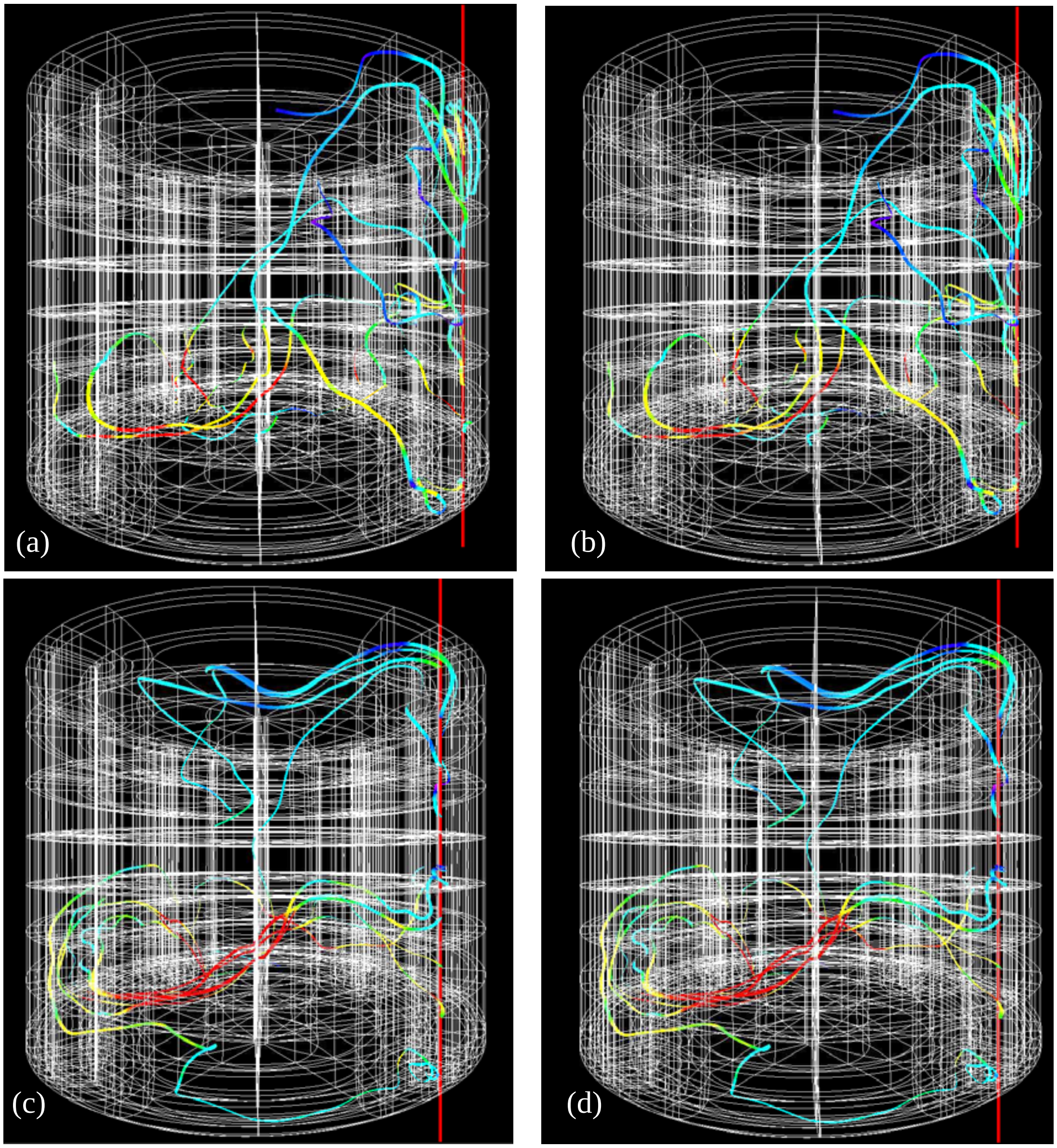

Fig. 4. Perspective right eye (a,c) and left eye (b,d) views of the stirred tank's CFD mesh together with streamlines, visualized as ribbons, originating from a vertical line emitter (red line) close to a baffle. The ribbons are color coded based on the magnitude of velocity. Stereoscopic viewing can be enabled via overlapping the above images by appropriate squinting of the eyes.

Figure 4 shows perspective views of the stirred tank's curvilinear mesh with streamlines originating from different positions of the vertical line emitter (red line)– a parameter that will be steered by the tangible controller (Fig. 3b). The immersive nature of CAVE-based visualization can be seen via overlapping the images by appropriate squinting of the eyes.

4. Limitations and ongoing work

The accuracy of the interpolation scheme with spherical regions of influence for nodal values needs to be addressed as a nodal value is extrapolated beyond the cells that the node defines – even into regions that are not included in the original curvilinear mesh. The visualization application needs to be optimized to avoid building display lists that currently cause a noticeable freezing of the displays – use of vertex buffer objects is being explored. To enable comparison with previous[9] and ongoing work on the stirred tank data use of the F5 format[10] for the binary files is being considered. Alternative means of reducing the streamline/pathline compute time need to be explored via (a) better use of the "master" machine in the CAVE that is at present redundantly doing the same computations as the display node, and (b) use of display threads for parallel computations while keeping the frame rate above a minimum of 30 fps. Construction of the mechatronic interfaces and the software development for the tangible controller are in progress.

Acknowledgement

This work is supported by NSF/EPSCoR EPS-0701491 (CyberTools), NSF MRI-0521559 (Viz Tangibles) and NSF DGE-0504507 (IGERT). We thank Farid Harhad, Bidur Bohara, Nathan Brener, and Marcel Ritter for their insights from related work on visualizing the stirred tank data[7] and their supportive interest in CAVE-based visualization. The Center for Energy and Environmental Studies at Sothern University and Scott Woodall are acknowledged for supporting and designing hardware upgrades to the CAVE.

References

1. J.L. Ditter and C.W. Hirt: A scalable model for stir tanks. Flow Science Technical Report TN38 (1993)
2. Reaction Engineering International: Introduction to Computational Fluid Dynamics (CFD) for Combustion.
3. JI-SUN KIM: Tangible User Interface for CAVE based on Augmented Reality Technique. Master of Science Thesis in Computer Science and Application, Virginia Polytechnic Institute and State University, (2005)
4. Dave Pape, Carolina Cruz-Neira, Marek Czernuszenko: CAVE Library version 2.6. Electronic Visualization Laboratory University of Illinois at Chicago, (1997)
5. S. Roy, S. Acharya: Stirred Tank Dataset, Unpublished.
6. http://www.cs.uic.edu/~kenyon/Conferences/GILKY/CAVE_DOD.html
7. B. Bohara, F. Harhad, W. Benger, N. Brener, S. Iyengar, M. Ritter, K. Liu, B. Ullmer, N. Shetty, V. Natesan, C. Cruz-Neira, S. Acharya, S. Roy, and B. Karki. Evolving time surfaces in a virtual stirred tank. In Vaclav Skala, editor, 18th International Conference on Computer Graphics, Visualization and Computer Vision'2010 (2010)
8. R. Sankaran, B. Ullmer, J. Ramanujam, K. Kallakuri, S. Jandhyala, C. Toole, and C. Laan: Decoupling Interaction Hardware Design Using Libraries of Reusable Electronics. In Proc. of Tangible and Embedded Interaction (TEI), pp 331-337, (2009)
9. W. Benger, M. Ritter, S. Acharya, S. Roy, and F. Jijao: Fiberbundle-based visualization of a stir tank. In 17th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, pp 117-124 (2009)
10. W. Benger: On Safari in the file Format Jungle – Why Can't You Visualize My Data? Computing in Science and Engineering-11-6, pp 2-6 (2009)