Web-Based Hydrodynamics Computing

Alan Shimoide¹, Luping Lin¹, Tracie-Lynne Hong¹, Ilmi Yoon¹, Sergio Aragon²

¹Computer Science Department
²Department of Chemistry & Biochemistry
San Francisco State University

Abstract
Proteins are long chains of amino acids that have a definite 3-d conformation and the shape of each protein is vital to its function. Since proteins are normally in solution, hydrodynamics (describes the movement of solvent around a protein as a function of shape and size of the molecule) can be used to probe the size and shape of proteins compared to those derived from X-ray crystallography. The computation chain needed for these hydrodynamics calculations consists of several separate programs by different authors on various platforms and often requires 3D visualizations of intermediate results. Due to the complexity, tools developed by a particular research group are not readily available for use by other groups, nor even by the non-experts within the same research group. To alleviate this situation, and to foment the easy and wide distribution of computational tools worldwide, we developed a web based interactive computational environment (WICE) including interactive 3D visualization that can be used with any web browser. Java based technologies were used to provide a platform neutral, user-friendly solution. Java Server Pages (JSP), Java Servlets, Java Beans, JOGL (Java bindings for OpenGL), and Java Web Start were used to create a solution that simplifies the computing chain for the user allowing the user to focus on their scientific research. WICE hides complexity from the user and provides robust and sophisticated visualization through a web browser.

1. Introduction
Proteins are long chains of amino acids that have a definite conformation in three dimensions after the chain of amino acids has been folded in a specific fashion. Proteins are essential constituents of living cells and the shape of each protein is vital to the function of the protein. Many protein structures have been deduced using X-ray crystallography and their atomic structures have been recorded in the Protein Data Bank using the pdb file format. X-ray crystallography measures the shape of proteins as crystals but proteins do not normally exist as crystals. Proteins normally exist in an aqueous environment and the size and shape of proteins can differ between a protein in an aqueous environment and a protein in its crystallized state [Gia92]. The hydrodynamic transport tensors of a molecule directly relate to the diffusive properties of the molecule and can be used to ask questions regarding the applicability in solution of the structures derived from X-ray crystallography. Accurate and effective hydrodynamics methods are usually computationally intensive and the recent advancement of computing power may play a crucial role in the advancement of hydrodynamics.

Computational science, including hydrodynamics computing, consists of many complex computation components developed by different authors on various platforms. As computing power increases, problems or approaches that were not easily tackled before become solvable, and consequently require larger and more complex computation modules to solve. Due to the complexity, tools developed by a particular research group are not readily available for use by other groups, nor even by the non-experts within the same research group. To alleviate this situation, and to foment the easy and wide distribution of computational tools worldwide, we are developing a Web based Interactive Computational Environment (WICE). When the interface is presented via a standard web browser, which is ubiquitous and a part of every researcher’s work environment, researchers take advantage of powerful and possibly remotely located software resources by a few mouse clicks. Hiding the operational complexity from the user promotes the uses of computational resources, collaboration and education.

The current computing chain for web-based hydrodynamics computing consists of three programs: MSROLL, COALESCE, and BEST. X-ray crystallography results only contain the atomic structure and not the molecule’s surface. MSROLL, developed by Connolly, generates a triangle mesh representing the surface. COALESCE, a Fortran program makes the output of MSROLL suitable for boundary element computations by reducing the numbers of triangles in the mesh and eliminating slender triangles. Researchers view the results from COALESCE several times and revise the input parameters for COALESCE until suitable results are obtained. BEST calculates the hydrodynamic
2. Hydrodynamics Background

The molecules that make up a protein structure are of the greatest interest because of their function while immersed in a liquid aqueous environment. The size, shape, and available conformations of molecules may be different in solution compared to a crystalline state [Ric80]. Thus, many methods are developed to probe the structure and dynamics of molecules in solution [BER76] [Ede83] [Str68] [Swa97].

The simplest (and slowest) of the dynamical motions available to molecules in solution involves translation and rotation. These two motions have the closest correlation to the overall molecular size and shape. The characteristic time constants for the slower process are often described as diffusive in origin and thus can be directly related to the hydrodynamic friction tensors of the molecule in question. Therefore, a hydrodynamic description of macromolecules is of interest when interpreting a wide class of experimental measurements.

For macromolecules, considering the solvent as a continuum is an excellent approximation. Thus, the governing equations for the computation of the hydrodynamic transport properties are the Navier-Stokes equations of fluid flow. Because most molecules do not have such high symmetry or smooth contours, hydrodynamics cannot be solved in such a way that an exact answer is returned. Therefore, a method that takes into account the precise molecular shape is needed. Methods to do this based on a set of hydrodynamically interacting beads were pioneered by Bloomfield and coworkers in the late 1960’s [Blo67]. The work based on bead models progressed rapidly in the decades following its introduction and it achieved a reasonable measure with regards to extracting shape and size information from the intrinsic viscosity and sedimentation coefficient measurements [Tel79]. The intrinsic limitation of course bead models is that in general, if the model parameters are set for fitting the translational diffusion coefficient, the rotational diffusion coefficient is not accurately predicted or vice versa [Gar77] [Wil79] [Gar81] [Ant89].

In the mid-70’s Youngren and Acrivos (YA method) presented an effective method for the calculating the numerical solution to the exact surface integral representation of the velocity field for the creeping flow equations [You75]. This expression was used to calculate hydrodynamic transport coefficients for spheroids to a high accuracy. This method lends itself easily to doing a numerical analysis of stresses distributed over the wetted surface of the body. Despite its accuracy, the YA method has not been extensively used for a variety of reasons. First, the integral equation approach, when compared to bead
methods, is mathematically obscure. Bead methods allow for simplified calculations that can be performed by hand while still returning very useful results. Integral equation methods, on the other hand, require powerful computers to ensure a successful implementation. With the advancements of personal desktop computers, supercomputer power is now available to the common user, which means the time for a widespread application of the YA method is now here. Aragon, recently showed that the YA method is the most general hydrodynamic method and is capable of producing unprecedented accuracy in the computation of transport properties of macromolecules which surpasses the accuracy previously attained through bead methods [Ara04].

3. Implementation

This section focuses on the three main components of WICE: current computation chains, 3D visualization, and current design implementation. A brief description of the current computation chains used in WICE will be discussed first, followed by a discussion on the 3D visualization tool developed for the WWW and the technology involved with this tool, and finally a discussion of the WICE design as an Internet Application.

3.1 Current Computation Chains

A triangulation of the surface of the molecule in question is necessary for hydrodynamic calculations using the YA method. PDB files, which are the results from X-ray crystallography, contain only the atomic structure of a protein and not its molecular surface. Connolly has developed a program named MSROLL which calculates the solvent accessible surface by first simulating a ball about the size of the solvent molecule, and then rolling it over the Van Der Waals surface of the atomic structure. MSROLL then generates a triangulation of the molecular surface which has typically been visualized using common mathematical packages. However, this triangulation is not suitable for boundary element calculations since it contains both slender triangles as well as triangles with small areas, which increases the error introduced into the calculations.

Aragon has created a Fortran program, COALESCE, which reduces these problematic triangles as well as reduces the total number of triangles using during surface triangulation. Researchers can view each of the triangulated surfaces generated from COALESCE as they revise their input parameters until suitable results for boundary element calculations are obtained. Once a suitable surface triangulation is generated, Aragon’s Fortran program BEST is used to calculate the transport sensors. Using the boundary element method for stick boundary conditions, BEST calculates the protein’s hydrodynamic interaction sensors. Figure 1 demonstrates this flow of data through the computation chain. Notice that at each step within the process, iterations can occur to generate a suitable output.

3.2 3D Visualization Tool on the WWW

Having an interactive and intuitive 3D visualization tool is an important component of WICE (Web-based Interactive Computing Environment). Several cross-platform technologies exist for delivering and visualizing 3D data over the WWW: VRML/X3D, proprietary technologies such as Macromedia Shockwave, Java Applets with Java3D, and JOGL. VRML/X3D is an open standard from the WEB3D Consortium that is primarily used as a model or scene description language for facilitating the browsing of 3D data across the Internet. VRML/X3D
is good for browsing the geometry of our molecular data, however, extending it to include other GUI components other than browsing would need to be developed using either JavaScript or External Authoring tools which are notorious for their instability (e.g. causing the browser to frequently shut down).

Macromedia Shockwave is a more flexible solution than VRML/X3D. A Shockwave viewer could be built with additional viewing functionalities, and 3D data could be delivered to the Shockwave viewer using an intermediary XML format. Though Shockwave is more flexible than VRML/X3D, it is not an open standard. Shockwave requires users to have a Shockwave plug-in and currently there is no implementation for Linux. In addition, certain desired functionalities may not be supported, and the decision as to which components are supported is decided solely by the proprietary company [Yoo04].

JOGL combined with Java Web Start is a cross-platform solution that works seamlessly with both Internet Explorer and Netscape, and it offers extensible solutions for developing a vertex viewer. JOGL is a Java wrapper for the OpenGL API, and it provides hardware-accelerated 3D on systems that support OpenGL. JOGL, when paired with Java Web Start, offers a sophisticated GUI with powerful 3D graphics rendering. It can also resolve many immature problems found when using Java Applets and Java 3D, as well as problems with usability and performance. Java Applets require the download of all necessary classes whenever it is executed whereas Java Web Start allows the caching of unchanged classes to the local machine so it does not need to be downloaded again. Also, Java Applets can support up to JDK 1.1 while Java Web Start can support the most up-to-date version which means that developers can leverage powerful Java Packages. JOGL also has better overall performance than Java3D since it is a lower level API than Java3D. Of the several technologies available for delivering 3D content on the Web, JOGL paired with Java Web Start is not only platform independent, but also the most flexible, and it offers the best performance for developing the vertex viewer needed for hydrodynamics computing.

The JOGL based vertex viewer was developed so that researchers can visualize intermediate vertex data from MSROLL and COALESCE over the Web while at the same time providing users with customizable shading and lighting functionalities to aid in both the researcher’s perception of the 3D geometry and the discovery of problematic triangles. The JOGL viewer has three main components: a vertex file reader that reads in vertex data from different file formats, a SWING based user interface for the viewer, and the JOGL viewer itself. The vertex file reader is able to read vertex data in .C3P (MSROLL), COALESCE, and ASC (ASCII 3D Studio) file formats. The JOGL viewer uses the vertex reader to read in each file format and to store the data into a common data format for each triangle. The common triangle data format stores the area of the triangle, a normal for each vertex, and its color(s), edge lengths, and...
respective vertices. For file formats that do not include the vertex normal, the vertex normal is calculated by averaging the normal for all adjacent surfaces. The vertex reader calculates the area of the triangle by using either Heron’s formula or the magnitude of the cross product of two edge vectors. The vertex reader also determines the minimum and maximum X & Y coordinates so that the molecule can be scaled properly for the screen.

The SWING user interface provides platform independent controls of the visualization options the JOGL viewer provides as seen in Figure 2. All options can be controlled with sliders, combo boxes, and buttons that GUI users are familiar with. Appropriate default options are set for each molecule so that researchers using the tool for the first time can use the program effectively. Clicking and dragging the molecule will rotate or translate the molecule depending on which option is selected.

The JOGL viewer has several functions that can help the user correctly understand a molecule’s complex 3D geometry. The JOGL viewer provides several colored lights and two white lights that can be turned off and on. These lights can be positioned independently of each other around the molecule as desired by the researcher (Figure 3). By providing a multitude of lights as well as the ability to move any light along its respective X, Y, and Z axis, researchers are able to shine lights into the nooks and crannies of a molecule while at the same time maintaining lights on the entire molecule or on other areas of interest. Such varied lighting options are not available in common mathematical packages that are currently being used for visualization. The JOGL viewer can rotate, translate, and scale the molecule which allows the researcher to focus on either on an area of interest or the overall shape. These lighting, scaling, rotating, and translating functionalities allow researchers to quickly and correctly pick out errors even in the most complex 3D molecular geometry.

The JOGL viewer provides several shading options to help researchers distinguish triangles while enhancing their 3D perception of the molecule. The JOGL viewer can display the molecule with either smooth or flat shading of the triangles. Smooth shading allows the researcher to gain an overall picture of the molecule’s geometry but sacrifices the ability to distinguish between individual triangles (Figure 4.a). Flat shading, on the other hand, provides a good overview of the individual triangles across the molecule’s surface but must sacrifice in the area of geometrical perception (Figure 4.b). The JOGL viewer can also superimpose a wire frame of the triangles on top of the shaded molecule. By doing this, researchers receive the best of both worlds in that they gain the overall picture of a molecule’s geometry that is provided through smooth shading, as well as the outline of individual triangles found through flat shading (Figure 3). To further enhance the ability of the researcher to see the individual triangles across the molecule’s surface, the JOGL viewer can change the colors of triangles displayed, along with the ability to assign a different color to an individual triangle (Figure 4.c). These shading options are not currently available to researchers using common mathematical packages, thus the JOGL viewer provides researchers with a tool to further distinguish the triangles that make up the surface of the molecule in question.

![Figure 4](image-url):
(a) Smooth shading of triangles with long skinny triangles colored in green, (b) Flat shading of triangles with long skinny triangles colored in red, (c) Triangles colored individually.
The JOGL viewer also provides functions to help the researcher distinguish and focus on problematic triangles. To help the user distinguish small triangles from the other triangles, the JOGL viewer can color triangles with small areas differently with respect to the largest area found (Figure 4 and 5). By using a combo box, researchers can quickly color triangles with areas that are 10%, 20%,... 90% of the maximum area found in the molecule. To distinguish slender triangles on the molecule surface, the JOGL viewer can also color triangles based upon their percentage from being perfectly equilateral. Again, using a combo box, the researcher can quickly color triangles that are 10%, 20%,... 90% of being perfectly equilateral. No tool that has previously been used or is currently being used, has the options of coloring triangles in such a way that it facilitates the spotting of errors. By performing a simple analysis of triangles in the model, coloring triangles with specific characteristics, and providing proper shading and lighting options, the JOGL viewer provides a profound new tool to help researchers visualize the source of errors in the triangulation of the molecule.

3.3 WICE Design

Prior to the development of WICE, researchers looking to generate the structure of a molecule would have to execute a series of programs, oftentimes more than once, through the command line. WICE was designed to simplify this process by providing researchers with an easy to use interface to these same programs that can be accessed from any standard web browser. The layout of WICE is clean, simple, and intuitive which allows even the novice user to easily navigate through the system, configure, and execute their files, as seen in Figure 5. Buttons and links to important features of WICE remain static and its position does not change. This provides users with a static layout regardless of the dynamic content contained within it. In order to support a consistent layout across web browsers, WICE uses the platform-independent Java technologies of JSP, Java Servlets, and Java Beans as well as a MySQL database to store its information.

While developing WICE, special attention was paid to the needs of those who would be using the application. Two features that researchers designated as being of high importance were support of virtual work spaces, and a configuration pipeline to store all configurations and results that have been saved or executed. Implementation of these features requires that users provide their user name and password before being given access to any files. This is done through a simple login page where, upon account validation, researchers can access any of their own files which have been previously saved, execute new files, or visualize previous program executions.

Because the area of hydrodynamic computing is a highly competitive field of study, researchers in

![Figure 5: Screenshot of the main WICE interface.](image)
this area want to have their findings protected from others, while at the same time, have the flexibility to share their data in order to promote research and community growth. In order to provide support for this ideal, WICE provides each of its users with individual virtual work spaces. These workspaces allow the researcher to specify which files, if any, he or she would like to make available to the outside world. For example, suppose a researcher generates the structure for a molecule that has not been generated previously. He/she may not want to publicly share the parameters used to create this structure, but he/she might want to share the basic data used for this generation. WICE would allow this researcher to designate which files to share and which to protect; thus giving him/her full control over his/her files in much the same way as storing the files on his/her own computer and executing the program from the command line would.

Having this work space for user files is only helpful if the user is able to save pertinent data into it. Previous to WICE, researchers would need to execute each program from the command line and even then, they oftentimes had to execute a program multiple times with different parameters before generating the correct result. Researchers had to manually store their program parameters and results because no system existed to handle this for them. WICE solves this issue by automatically storing both the results and the parameters used to generate the results into the researcher’s individual work space through our configuration pipeline.

Configurations defined by the researcher are passed through the pipeline in two simple steps. First, the pipeline checks to see if this configuration is a duplicate of any other configuration which the researcher has already sent through it. If it is, then the results of this particular configuration are sent back to the user without the need to execute the given program. However, if this is a new configuration, the proper program will be called to execute. After execution, both the configuration and its results will be saved within the researcher’s work space before being sent back through the pipeline. There are many advantages to using this type of pipeline, the first being that it saves on computation time. Because all configurations and their results are saved by the pipeline, the need to execute a program for each incoming configuration is eliminated. Duplicate configurations by the same user can have their results returned without any computational expense. Secondly, since all configurations and results are saved, users can easily pull up these records and see what they have done in the past. There is no longer a need to manually keep track of all past configurations and their results because it is now a mouse click away. This saves the researcher time in that he/she no longer needs to write everything down; instead he/she can use the extra time to develop better representations of the molecule’s structure.

Although many other features have been built into WICE such as file uploading; aside from the protein visualization, individual work spaces and history of configuration and execution are the two most important features of WICE. These two features make WICE a powerful tool for researchers looking to do hydrodynamic computing.

4. Future Work

Web-based hydrodynamics computing implementation was started as a prototype of feasibility and usability study for more flexible and configurable framework, Web-based Interactive Computing Environment. Therefore, there are many improvements and features that need to be added. WICE should be expanded to include other computational components easily. Doing so will allow WICE to reach a wider audience and make it an even more powerful tool. The consolidation of various computational components into one easy to use interface which can be accessed from any standard web browser will not only take advantage of advancements made in computer power, but it will also provide researchers with a wider range of tools and methods to continue their work.

With the inclusion of other computational components, the JOGL viewer will also need to be altered to provide visualizations for output files other than .C3P and .ASC. It may be wise to consider altering the current WICE architecture into a plug-in architecture, so as new computational components are added to the environment, they can easily be integrated with minimal code changes. This will cut down on the work needed to maintain the application as well as make it a much more flexible environment.

We are also researching the feasibility of implementing WICE as a distributed application. This means that instead of having the application and all its data stored on one server, we are looking into implementing the application environment on one server which saves and accesses all its data from another location. Ultimately, this would be a good feature to include because as more and more computational components are added and more researchers are given accounts to WICE, the amount of data stored will grow exponentially and another computer will need to be integrated into the system regardless.
Finally, some form of usability testing would be beneficial to WICE development in the future. We are developing this application in response to researchers who are looking to perform all their computations in an easy to use environment. Usability testing will ensure that we are heading in the right direction and that our system is interactive and intuitive enough for the common user. It will bring up issues that we may not have previously thought of, and will, in the end, help us to develop a better product.

References


