49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition
4 - 7 January 2011, Orlando, Florida

AIAA 2011-1319

Introducing SE-FIT: *Surface Evolver* – Fluid Interface Tool for Studying Capillary Surfaces

Yongkang Chen[*], Ben M. Schaeffer[†], and Mark M. Weislogel [‡],
Portland State University, Portland, OR, 97201

Gregory A. Zimmerli[§]
NASA Glenn Research Center, Cleveland, OH, 44135

As an efficient program for studying capillary surfaces, *Surface Evolver* (SE) has a steep learning curve and is without a user-friendly graphical user interface. Here, we describe a new SE-based program, the *Surface Evolver* - Fluid Interface Tool (SE-FIT, pronounced 'see-fit'). SE-FIT has been developed to dramatically broaden the application of SE and provide a user-friendly tool for studying the shape and stability of capillary surfaces. SE-FIT retains the core strengths of SE but adds a Windows-based Graphical User Interface (GUI) to interface with SE's command-line functions. SE-FIT uses an intermediate text file layer to communicate with SE, automatically generating session logs while ensuring seamless communication between SE and the GUI. Groups of pre-defined geometric elements, such as spacecraft propellant tanks, have been created to enable rapid assembly of user-specified configurations. More generally, certain CAD/CAM drawing files generated from third-party software can be imported as well. For run-time processing, a general convergence algorithm has been developed to make the convergence computation a real routine. Features that enable batch process of parameter sweeps have been created. A real-time total energy variation graph and a histogram table are made available as convergence diagnostic tools. SE-FIT features utility tools including a parameter list table and a file explorer. A built-in text editor is included to facilitate editing and debugging of SE geometry and script files. Post-processing features include exporting data such as the container and interface geometry as an image file, and other file formats for further CFD simulations or rendering. A stand-alone SE data-file viewer enables real-time graphical viewing. These and other features make SE-FIT a unique software tool for rapidly calculating the shape and stability of capillary surfaces in complex geometric settings with an efficiency at least an order of magnitude greater than that of existing CFD programs.

I. Introduction

Capillary surfaces have proven a fascinating subject for both appreciation and study since ancient times. They present themselves in numerous forms during daily life in the form of dew drops, bubbles, soap films, and so on. Though mathematical theory of capillary surfaces has slowly developed over the centuries it was not until the last few decades that new technologies have put a more urgent demand on substantially more qualitative and quantitative understanding of phenomena relating to capillary surfaces. So far the new theory successfully predicts the behavior of capillary surfaces confined to certain classes[1]. However, a mathematical quantitative prediction of capillary phenomena related to the shape and stability of equilibrium capillary surfaces remains a significant challenge. As a result, one must often resort to numerical methods to find solutions. As one of many numerical tools, the open-source *Surface Evolver* (SE)[2] has played an important role in the last two decades. This paper is written to convey that a similarly open-source software interface to SE has been and will remain under development for the foreseeable future. The intent of the software is to improve accessibility to SE so that important fluid interface information can be computed in a timelier, more routine, and more frequent manner. These objectives are rather

[*] Research associate, Mechanical and Materials Engineering, AIAA Senior Member.
[†] Graduate research assistant, Mechanical and Materials Engineering and Electronic Engineering.
[‡] Professor, Mechanical and Materials Engineering, AIAA Senior Member.
[§] Research Aerospace Engineer, Propulsion and Propellants Branch.

applied, but they are also expected to aid a variety of research fields in addition to simply improving the design and education process. This paper is written to an audience somewhat familiar with the field of fluid interface computations, and, in particular, SE. However, even this approach may complicate the presentation to the point that would-be users may detect just another barrier to entry, when the intent of SE-FIT is to eliminate such a barrier.

Appearing in late 1980s, SE has since become 'the reigning' tool for studying equilibrium capillary surfaces. SE focuses only on fluid/fluid and fluid/solid interfaces which are simulated using the finite element method. It uses a gradient descent method to compute the minimal energy state of just the interfaces. The approach provides significant advantages over general Computational Fluid Dynamics (CFD) packages, where the entire fluid body must be meshed with the fluid interfaces computed as a part of the fluid body. In most cases, the efficiency of SE is at least an order of magnitude greater than that of CFD packages.

SE is written in C to be cross-platform. On PCs with Windows Operating System[**], SE operates in a DOS console window based mode with an OpenGL window displaying fluid surfaces graphically. It requires the user to manually type commands to perform any task. Alternatively, the user can compile pre-defined scripts that can be imported into SE to perform the same tasks in a more convenient and efficient way. This mode of operation demands substantial knowledge of the SE system including input file setup, mesh management, syntax of C and SQL languages, and so on. Consequently, the learning curve is steep.

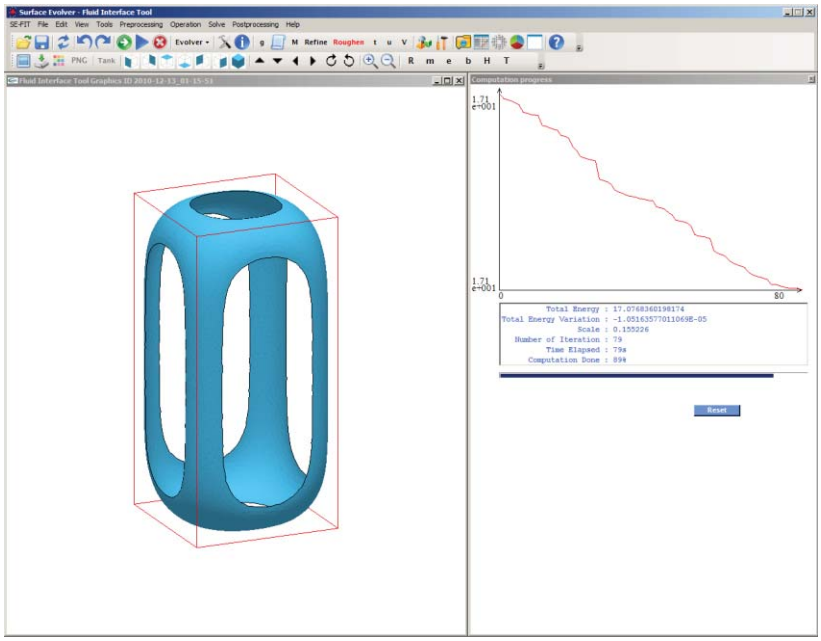


Figure 1. *Surface Evolver* – Fluid Interface Tool

Despite the steep learning curve, SE has seen substantial growth in its application over the last two decades. A search on ISI Web of Knowledge produces a total number of 157 journal articles. A subscription to Google Scholar Alerts with the keyword '*Surface Evolver*' generates 40 alerts, from May 20, 2010 to the time of this writing, most of which consists of citation information of one or multiple journal articles. The number of citations reveals the trend of a growing need for SE in subjects such as microgravity tankage[3], inkjet printing[4], nanotechnology[5], interface shape[6], capillary stability[7], porous media[8], self-assembly/self-alignment[9,10], micro-scale wicking structures[11], liquid foam[12], capillary forces[13,14], and more. Across all publications it is clear that some are generated by well-trained users whereas others are not. In fields such as aerospace engineering there seems to be an acute need of a user-friendly interface between SE and users. For example, it is still necessary to predict the fluid interfaces in liquid fuel tanks. We believe a friendlier user interface would further promote the use of SE and expand opportunities to discover fluid interface configurations that may be difficult to foresee without experimental

[**] In order to describe procedures adequately, it is occasionally necessary to identify commercial products by the manufacturer's name or label. In no instance does such identification imply endorsement by NASA or Portland State University, nor does it imply that the particular product is necessarily the best available for the purpose.

observations. Such configurations can be exploited for enhanced control. Conversely, it is certain that a lack of such knowledge can pose a serious threat to successful critical system function[15].

The *Surface Evolver* - Fluid Interface Tool (SE-FIT) was developed to meet such a need, see Fig. 1. The goal of SE-FIT is to provide a supplemental set of Graphical User Interface (GUI) elements that significantly reduce the training for beginners and simplify the operation of SE for advanced users, ultimately broadening the use of SE while maintaining its core capabilities.

The ultimate goal of current SE-FIT development is to build a fully integrated front-end with a set of graphical user interface (GUI) elements. Such a front-end enables the access to functionalities that are developed along with the GUI to deal with pre-processing, convergence computation, and post-processing. In other words, SE-FIT is not just a front-end consisting of GUI elements, but an integrated environment that can (or will be increasingly able to) perform sophisticated computation tasks employing minimal interaction with the user. These functions are created using commands and procedures provided by SE. They form the foundation for a high performance front-end that substantially simplifies using SE.

As will be demonstrated herein, the real power of SE-FIT lies in its automated pre-processing that includes geometry import functions, pre-defined geometries, convergence computation operation, computational diagnostic tool, and crash-handling capabilities to sustain extensive computation tasks such as *Parameter Sweeping* and *Parameter Critical Value Searching* batch processes.

II. File Layer Mechanism

Since SE is written in C, the SE-FIT GUI front-end is developed with Microsoft Visual Basic .Net, and the two parts are compiled separately which generates separate executable files. As a result, a smooth flow of communication between the GUI front-end and back-end with SE must be created and maintained to ensure the high-performance of the whole system.

During the early stages of development, it appeared that maintaining seamless communication between the front-end and SE was not trivial. It became necessary to modify the original SE code in order to obtain capabilities that are required for an enhanced and synchronized communication. To this end, a file layer was created and became the most essential addition to the original SE code. A schematic view of the overall architecture is shown in Figure 2. The file layer serves as a command buffer to ensure a continuous and sequential execution of commands sent from the front-end to SE. It works in both Windows and various UNIX/LINUX systems. The file layer logs input commands and SE output; it also supports user interruption requests. It enables the front-end to load any input log file so as to reenact the operations of a previous session for the purpose of debugging.

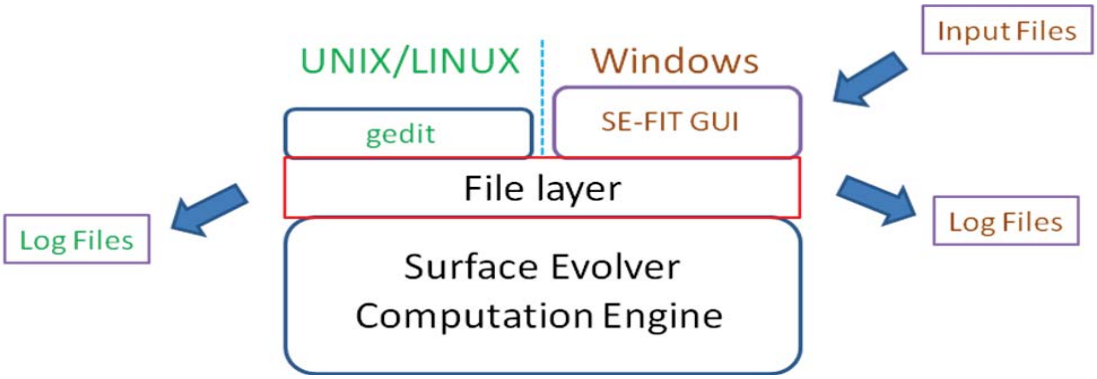


Figure 2. File layer architecture.

In this way the modified SE extends the capability of the original SE. For example, SE is capable of executing a series of commands, but this operation is limited to one input file and treats both errors and script completion as a condition to return to the standard command input mode. The modified SE, however, can abort the execution of the troubled command and continue to the next command in the queue. Such capability makes it possible to load different files and perform computations in a batch process mode, even without a front-end GUI. It also makes it possible to create front-end functions that perform sophisticated tasks such as the *Parameter Sweeping* and the *Parameter Critical Value Searching* batch processes, which significantly boosts the efficiency and productivity of SE applications.

III. GUI Features

A list of the SE-FIT GUI elements is provided below.

A. Command Input and SE Output

The Command Input and SE Output window replaces the SE command-line interface window while retaining the original means of interaction with SE.

B. File Explorer

The File Explorer enables users to browse only the SE input files, such as FE files and DMP files. These files can be loaded in a sequential manner to produce a slide show. Selected files may also be launched to the Text Editor from within the File Explorer.

C. Text Editor/Script Debugger

The Text Editor enables viewing, editing, and debugging of SE related files. In addition to standard features that one can find in any text editor, the Text Editor features a keyword highlighting scheme. Three types of keywords are highlighted: SE keywords, standard C and script language keywords, and SE-FIT keywords. A primitive SE script debugging capability is implemented which is yet to be fully developed.

D. Total Energy Graph and Computation Progress Bar

The Total Energy Graph displays a plot of total energy versus the number of iterations. It serves as a diagnostic tool to identify the convergence of the free surface. The Computation Progress Bar indicates the estimated percentage towards completion for the current computation task.

E. Utility Window

The Utility Window includes three Tab windows. The first displays the parameter list. The parameter list enables the user to modify parameter values throughout a SE-FIT session. The second tab displays a histogram of the edge length and facet area. The third tab displays statistics of the free surface**.**

IV. Convergence Operation

The SE iteration command **g** performs a significant amount of operations and computations behind the scenes to evolve the fluid interface to a lower energy state. The user, however, must oversee mesh refinement and grooming while continually judging the convergence of the computation. This approach provides flexibility, but often requires significant interaction from the user to achieve convergence.

In SE-FIT, a convergence scheme is developed which automates the operation of mesh refinement and grooming and identifying convergence. It consists of **g** commands followed by algorithms to identify mesh irregularities and perform grooming operations when necessary. It includes a method to examine the variation of the total energy and identify convergence once a prescribed criterion is satisfied. The method is still under development to address computations with total energy oscillations.

The convergence scheme serves as the core upon which automated procedures can be developed. For example, procedures are developed to automatically refine the mesh during run-time. To achieve a degree of mesh resolution at the end of the computation it is generally a good practice in SE to refine a relatively well-converged surface and continue the computation, repeating this process until the desired resolution is reached and convergence criterion satisfied. The final resolution of the mesh can be specified in terms of either a vertex number or a maximum edge length.

Procedures are also developed to handle situations where it is necessary to identify if the total energy is truly at a minimum or at a saddle point. These procedures employ SE commands that examine the existence of negative eigenvalues of the Hessian matrix and perform operations accordingly. These procedures are especially useful for studying the stability of capillary surfaces.

The automation of mesh refinement and convergence operations comes at the price of added computation time. The great advantage however is that it frees the user to further focus on the problem, or entirely different tasks. Furthermore, the automation enables more sophisticated operations such as a *Parameter Sweeping* and *Parameter Critical Value Searching* batch processes. The *Parameter Sweeping* (PS) batch process enables computations of equilibrium capillary surfaces in certain geometries. An example is to study ullage bubble configurations in a liquid fuel tank[16] for a set of parameters with ranges specified by user. Typical parameters are liquid fill level, contact angle, and acceleration, e.g. gravity. This process is not impossible using SE, but it is fully automated in SE-FIT.

The *Parameter Critical Value Searching* batch process (PCVS) is created for determining critical values of parameters when studying the stability of capillary surfaces (i.e., Bond number). This feature is a manifestation of the combined capabilities of the file layer and the GUI front-end. A case study presented in section VIII-D below will demonstrate this capability.

V. Pre-processing

In general, preparation for SE computation is implemented in an input FE file that defines geometry, constraints, and energies. Creating such FE files can be painstaking for complicated geometries. An automated process for creating geometries, constraints, and the initial free surface is highly desirable. Note that the word *geometry* or *geometries* is used here to refer to a geometric entity that only consists of solid walls that serve as constraints that will be applied to the free surface for SE computation. Two mechanisms have been developed to this end. In one, pre-defined geometries are created that can be tailored to match a particular application (aka, 'pre-built models'). In the other, the geometry is imported from a file that is created using CAD software.

A typical example of a pre-defined geometry is a closed cylindrical container with elliptical ends, a common geometry used for spacecraft fuel tanks, as shown in Fig. 3. This container type includes all spheroidal tanks as well as right circular cylinders with flat ends. The user may assign the dimensions of the container, define internal structures such as baffles, and set the initial free surface with potential energies that may include both gravitational and centrifugal contributions.

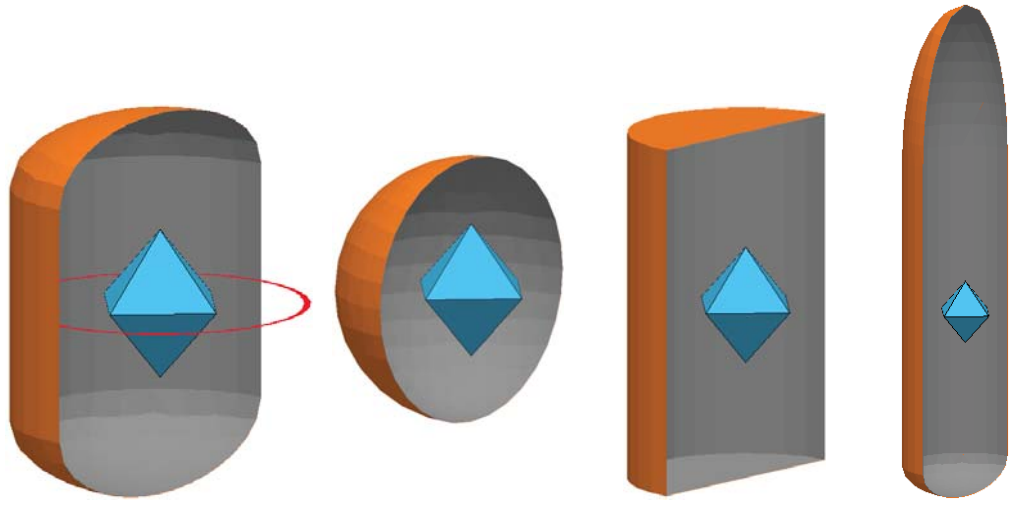


Figure 3. Typical spacecraft tanks generated from one of many possible pre-defined features. The blue diamond shape in the center is the initial fluid interface; (a) Cylindrical tank with a single ring baffle; (b) Spherical tank; (c) Cylindrical tank with flat ends; (e) 'Space shuttle' external tank.

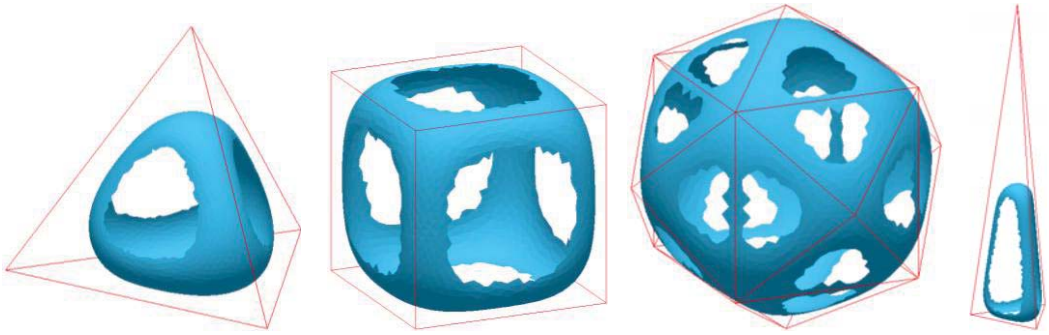
An automated pre-process is developed to import the geometry of a closed container with planar walls from an STL file, with examples shown in Fig. 4. It then sets up constraints, creates an initial free surface, and sets up potential energies for the free surface. STL is a common file format that can be generated with standard CAD/CAM software. It includes pure mesh data which is sufficient for geometries that posses planar walls. Because curvature information is compromised in STL files, the STEP file format is preferred for geometries with curved walls.

The initial free surface is created using a closed surface (or 'bubble') representing the interface between a liquid and gas. The pros and cons of the 'bubble method' is discussed in next section, but here it provides a simple means of setting up the free surface and works perfectly for cases where the wetting or 'contact angle' of the liquid is zero degrees. For non-zero contact angles a more sophisticated manipulation of the interface employs an *Artificial Velocity Method* introduced below.

VI. The Contact Line Integral and Artificial Velocity Methods

In SE formulations of wall constraints which represent solid walls that are partially wetted by a fluid are facilitated with integrals along edges that represent contact lines to accurately account for the surface energy between the fluid and the walls. The *Contact Line Integral Method* (CLIM) employed by SE removes the necessity of including facets to represent the liquid/solid interface and tackles the troublesome problem of near-zero velocity of vertices

constrained on solid walls. For example, if facets representing the fluid/solid interface are included instead of using CLIM, the vertices on a receding contact line may run into the vertices on those facets so as to stall the computation prematurely. CLIM also saves the computer memory usage which in turn reduces the computation time. However, it requires setting up integrands which can become cumbersome if the geometric configuration is complex. For example, see Abkarian[17] for the case of liquid drop configurations with surface particles.



a b c d

Figure 4. Fluid interfaces in zero gravity inside (a) a tetrahedron (only wire mesh shown); (b) a cube; (c) a spherical polyhedron container of 24 faces; (d) a tapered triangular section. The computation is set up using the automated pre-processing functions to import the container geometries and set up constraints and energies for the free surfaces.

To completely automate the preprocessing of creating and/or importing geometries as well as set up constraints without intensive user participation is a goal of the SE-FIT development, which CLIM cannot achieve easily. A more practical approach is created in SE-FIT with the purpose of addressing the problem of near-zero velocity of vertices on the liquid/solid interface, which we call the *Artificial Velocity Method* (AVM).

In AVM, all of the vertices on the constraints are first grouped into layers based on their association to the vertices on the contact lines, as shown in Fig. 5. For example, the vertices connected to at least one vertex on the

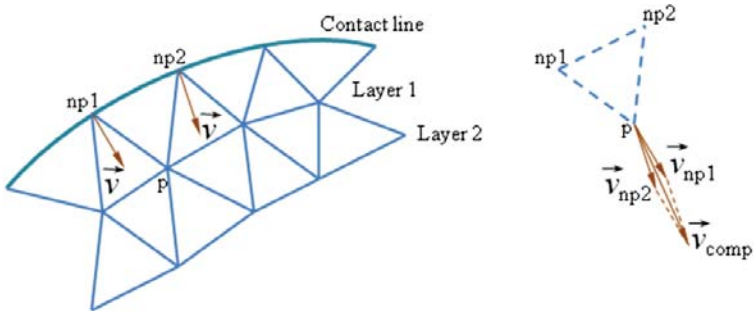


Figure 5. Contact line and layers of vertices on constraints (left); the vector diagram shows compound velocity vectors on vertex 'p' of layer 1 (right).

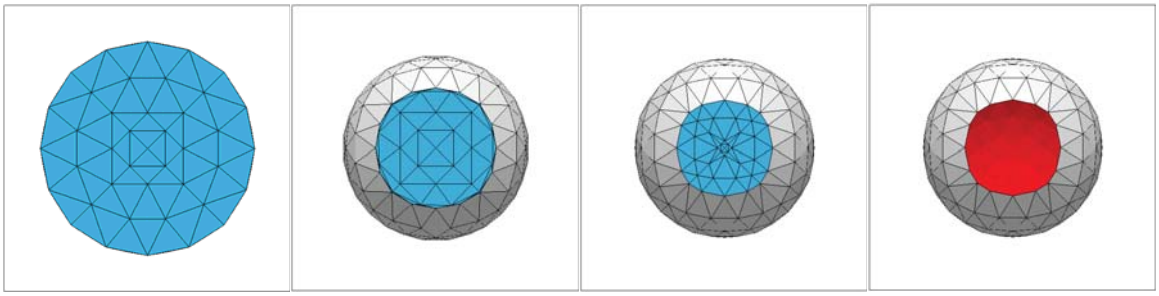
contact line through an edge are grouped into layer 1. The vertices that are connected to at least one vertex on layer 1 are grouped into layer 2, and so on. The direction of the artificial velocity vector on any inner vertex 'p' on layer 1 can be determined by the addition of the velocity vectors of its neighboring vertices 'np1' and 'np2' on its upper layer. The contact line is treated as the upper layer for the vertices on layer 1. A weighted average of the velocity magnitudes of the neighboring vertices can be taken as the magnitude of the artificial velocity vector such that

$$\vec{v}_{\mathrm{p}} = \psi\left(\left|\vec{v}_{\mathrm{np1}}\right| + \left|\vec{v}_{\mathrm{np2}}\right|\right)\frac{\vec{v}_{\mathrm{comp}}}{\left|\vec{v}_{\mathrm{comp}}\right|},$$

where ψ is an optimization factor.

The surface of a liquid drop on a solid surface is taken as a model problem for which a preliminary study provides encouraging results. The interface, including facets on the solid wall, is initially converged with a 90^{o} contact angle as shown in Fig. 6a. The contact angle is then changed to 150^{o}. For the first two cases the tension of the facets on the wall (Fig. 6b and c) is changed to 0.866 to produce an approximately 150^{o} contact angle. AVM is

applied in case of Fig. 6c. CLIM is used in the case of Fig. 6d. A 200 iteration run without mesh grooming is performed. The interfaces in the case of Figs. 6c-d are well-converged to almost the same energy level in healthy conditions whereas in Fig. 6b the scale factor has dropped significantly and tiny edges are present near the contact line such that the evolution of the surface is stalled. This example shows that AVM is a promising approach in handling non-zero contact angle to avoid the troublesome situation presented in the case of Fig. 6b. With AVM the setup of wall constraints is reduced to simply identifying analytical formulas for the walls since there is no need of setting up integrals along the edges of contact lines as in CLIM. This will potentially make it possible for the automated pre-processing to import complex geometries and handle setting up constraints.



a **b** **c** **d**

Figure 6. Bottom view of a liquid drop on a solid wall: a) initial fluid interface with 90^o contact angle, which is changed to 150^o in other images, b) interface after 200 iterations without AVM, c) interface after 200 iterations with AVM, d) interface after 200 iterations with CLIM. The outer side of the fluid interface lying on the solid wall is colored with light blue while the inner fluid interface is colored with red.

VII. Post-processing

At present, SE-FIT post-processing features are less extensive than pre-processing features, but they do include saving a snapshot as an image file of the OpenGL graphical window. There are existing scripts in SE that export the surface mesh in different formats for further data processing or rendering. SE-FIT makes it easy to access these functionalities. A stand-alone SE data-file viewer is created to enable real-time graphical viewing without launching SE-FIT. As an alternative, the open-source program Paraview[18] is an adequate tool for post-processing.

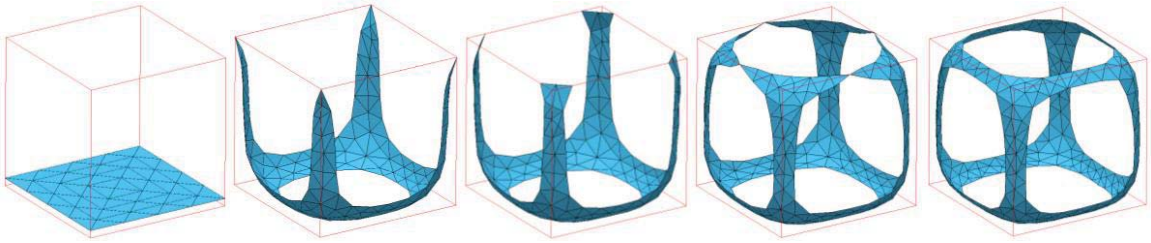
VIII. Performance/Case Study

A. A Comparison with OpenFOAM CFD Package

As mentioned earlier, in SE, a fluid interface is of zero thickness and represented by a mesh consisting of a collection of triangles, and the bulk fluid is not of concern except that its volume is conserved. The computation based on a gradient descent method moves the interface toward a lower energy state until equilibrium is reached. No fluid dynamics equations are solved. In typical CFD the entire bulk fluid is represented by a mesh and the fluid interface has a finite thickness (i.e. VOF method). The computation solves the fluid dynamics equations and the fluid interface reaches either steady state or static equilibrium. Using CFD to compute equilibrium capillary surfaces is far more computationally intensive than using SE. A comparison between the performance of SE and a standard CFD package demonstrates the advantages of SE for computing capillary equilibria.

For example, the equilibrium shape of a fluid interface inside a cubic container in zero gravity is chosen as a trial problem. Fig. 7 shows a series of snapshots from the SE-FIT computation. The cubic container is 5% filled with fluid that initially covers the base of the container (Fig. 7a). In zero gravity, with the Concus-Finn[19] critical wetting condition satisfied, fluid wicks along the side edges and advances toward the top corners of the container (Fig. 7b). Once fluid reaches the top corners it bifurcates and wicks along the top edges of the container (Fig. 7c). As a result, along each top edge there is a pair of advancing menisci that move toward each other (Fig. 7d) and eventually merge near the center of the edges (Fig. 7e).

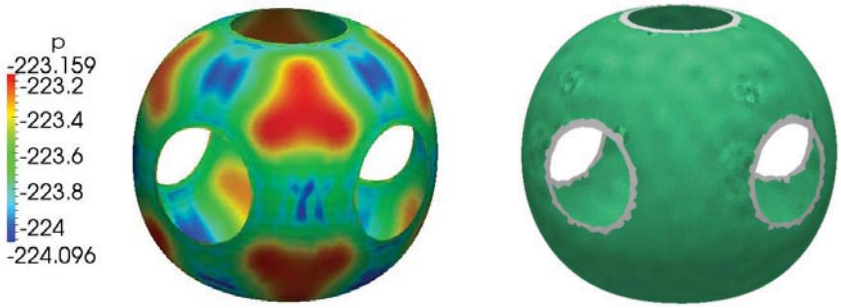
A comprehensive performance comparison between OpenFOAM and SE-FIT is beyond the scope of this study, but even a cursory comparison quickly conveys the SE-FIT niche. Sticking with the partially filled cubic container problem, for a side length of 1 cm, a 30% fill level, zero degree contact angle, and 2 cSt silicon oil the interface is computed using both methods. In the OpenFOAM run, a flow of 10 seconds is computed. It is observed that after 3 seconds, the pressure difference between the liquid and vapor continues to approach level, but the absolute pressure of both phases does not. The OpenFOAM computation for those 3 seconds requires 3288 seconds. For the same



a **b** **c** **d** **e**

Figure 7. Fluid interface evolution inside a cubic container computed with SE-FIT.

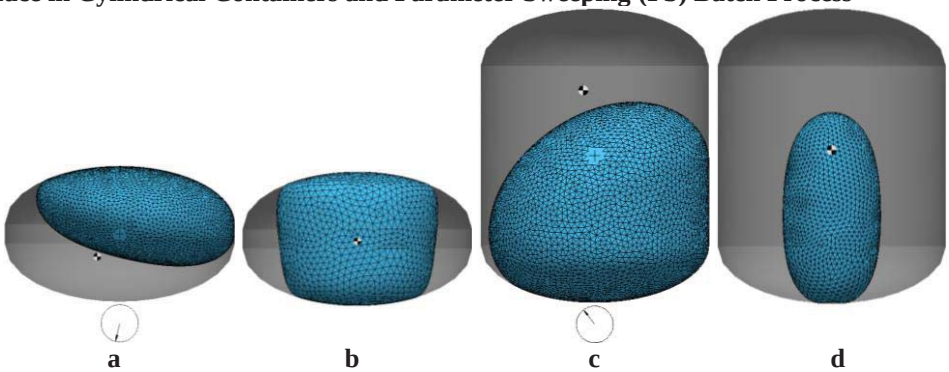
resolution of 7000 surface vertices extracted from the OpenFOAM result, the computation time with SE-FIT is 155 seconds—a 20-fold reduction. But this improvement does not consider accuracy which argues decisively in SE-FIT's favor and several arguments are lightly noted here prior to a deeper treatment elsewhere. For example, the fluid interfaces with their respective pressure distributions are shown in Fig. 8 with a pressure scale provided for the OpenFOAM calculation only. In both cases the pressure is narrowly confined, signifying an approach to equilibrium, but the distributions are telling that the CFD pressure is clearly not the Young-Laplace pressure. Both methods conserve liquid volume to < 1%, but observations of the dry circular regions on the cube faces in Figure 8 reveal that the OpenFOAM dry surface area is 32% greater than that of the SE-FIT calculation. Lastly, a most critical metric of an interface is its mean curvature. In OpenFOAM, the mean curvature must be acquired from the pressure difference between the vapor and liquid phases, which is found to be 18% lower than that of SE-FIT result. Such comparisons raise the arresting question of whether it is appropriate to use CFD tool to study equilibrium capillary surfaces at all. Further investigations along these lines are merited.



a. OpenFOAM **b. SE-FIT**

Figure 8. A comparison between OpenFOAM CFD and SE-FIT computations. The unit of the pressure in (a) is g/cm.s^2. In (b), the facets are colored based on the average of mean curvature of the vertices; the pressure along the contact lines is significantly different from that on the rest of the surface due to the mesh grooming operations, which does not have observable impact on the rest of the surface.

B. Fluid Interface in Cylindrical Containers and Parameter Sweeping (PS) Batch Process



a **b** **c** **d**

Figure 9. Equilibrium bubble (ullage) configurations in fuel tanks for a given gravitational acceleration (a, c) or centrifugal acceleration (b, d). The direction of the gravitational acceleration is shown with an arrow within a circle below the tank and the fluid center of mass is represented by symbol.

Cylindrical containers with elliptical ends are the first pre-built geometries in SE-FIT to meet a need to study fluid interface configurations in spacecraft fuel tanks. Computed free surfaces in two selected tanks[3] are shown in Fig. 9. In this type of study it is required to compute interface configurations at different values of relevant parameters such as fill level and acceleration vector. To meet this requirement, a *Parameter Sweeping* batch process feature is implemented. As mentioned earlier, this feature is built on the file layer mechanism, which makes it routine to perform intensive computations that cover a test matrix with up to five parameters.

C. Interface in an N-Prism

Fluid interface shapes in an n-sided prism in zero gravity is of both fundamental and practical interest. With the Concus-Finn condition[19] satisfied, the interior corners along the edges of the container provide passageways for the fluid to extend throughout the container, connect, and form a network of fluid interfaces. If the computation begins with a simple flat fluid interface that covers the base of the container, the interface will likely go through several topological changes such as bifurcation at vertices and merging along edges as demonstrated in Figures 7 and 10. Procedures have been developed within SE-FIT to treat topological changes to generate contact lines when the fluid interface meets the base of the container forming a dry region and handle interface bifurcations in corners where three planar walls meet and mergers along edges where two planar walls meet.

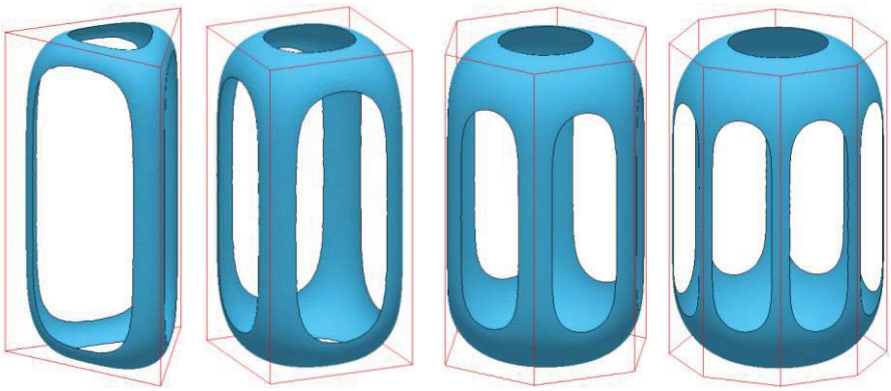


Figure 10. Interface in a 3, 4, 6, 8-prism. 3, 4, 6-prism, 0 degree contact angle; 8-prism, 10 degree contact angle.

D. Parameter Critical Value Searching (PCVS) via Batch Process

The problem of the corner-edge bound drop (Fig. 11) concerns the stability of a capillary surface as a function of parameters such as Bond number. In this study, it is of interest to identify the critical Bond number as a function of gravity vector for a fixed corner angle and contact angle. For Bond numbers great than the critical value the drops become unstable and 'fall off the edge.' This particular study is an extension of the so-called wall-edge bound drop[20] problem. Due to the difficulty in mathematical prediction of the stability of three dimensional capillary surfaces, numerical computation is so far the most efficient way to predict such stability. The problem serves as a case study to demonstrate the PCVS batch process function of SE-FIT. This capability allows extensive computation and sophisticated parameter studies because of the file layer. Further details of the study will appear elsewhere[21].

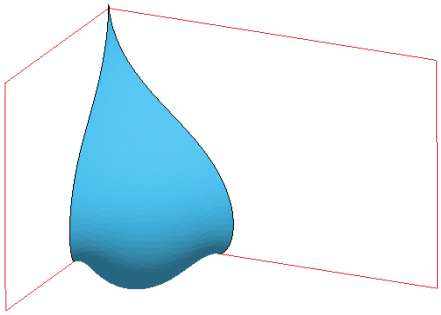


Figure 11. A corner-edge bound drop.

Fig. 12 shows a comparison of the critical Bond number computed before and after the creation of the file layer. Without the file layer, each case represented with one data point in Fig. 12a was computed separately, with each

data point being in turn a manually managed batch process that sweeps cross a certain range of Bond number. On average, it requires one work day to compute one data point with one machine. Each data line in Fig. 12b consists of 33 data points with a 5° increment. The computation time is not reduced for each data point, but because of the *Parameter Sweeping* batch process, the computation can be performed in a continuous manner such that the total computation time is reduced by at least a factor of two.

Constrained by the computational resources, the increment for the gravity orientation angle (ϕ) is relatively large in Fig. 12a. As a result, the trend in variation of the critical Bond number versus ϕ is not clear and even confusing. The confusion abates with new results obtained after the creation of the file layer in SE-FIT, as shown in Fig. 12b. The latter computation is carried out with the Bond number incremental computation batch process embedded in the angle ϕ incremental computation batch process. In this process, the user is still involved in identifying the critical Bond number by studying the data produced. This involvement is eventually removed with the creation of PCVS batch process.

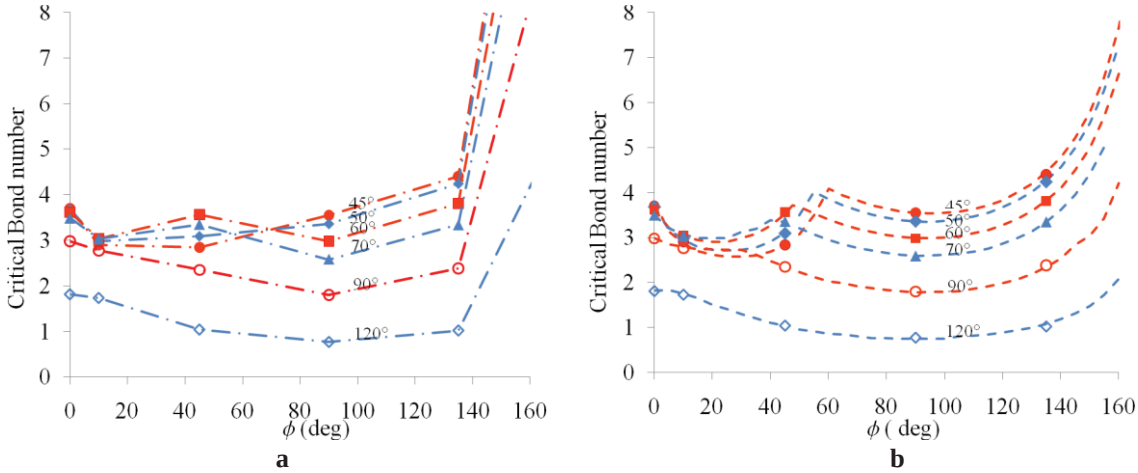


Figure 12. A comparison of computational results shown in (a) before and (b) after the creation of the file layer; in (a) the data points with symbols are computed results which are connected with dash-dot lines; in (b) the dashed lines are computed results with 5° increments with data points from (a) shown as well. The numbers above the data lines are contact angle values, ϕ is the orientation angle of the gravity, and the data is for 45° corner half angle.

PCVS is a feature created in the SE-FIT front-end that monitors the abrupt change in the computation (e.g. the dimensions of the fluid interface) so as to identify the critical Bond number value at which the fluid interface becomes unstable. PCVS is also able to use the information from the computation history for further computations. For example, PCVS uses the critical Bond number value identified for the previous ϕ value as the starting point for searching the critical Bond number for the new ϕ value. This process continues until it sweeps across all the values specified. The process identifies the 'crash' or 'hanging' of SE and restarts SE automatically to resume the batch process. In the end, this feature significantly reduces by at least a factor of 3 the computation time consumed to acquire the data for an entire 'data line' in Fig. 12b.

IX. Conclusion

SE-FIT is based on the existing program *Surface Evolver* and is introduced herein with select key features demonstrated. SE-FIT employs a collection of Windows-based graphical user interfaces to create a user-friendly tool for the study of capillary surfaces. A file layer approach is adopted to ensure seamless communication between the GUI front-end and SE back-end. The file layer enables the creation of other features that handle extensive computation tasks such as *Parameter Sweeping* and *Parameter Critical Value Searching* batch processes, access to a catalog of widely variable and configurable pre-built models, and ability to import geometries produced by third party software. Such features can dramatically increase the efficiency and productivity of SE applications in science and engineering. It is hoped that SE-FIT will benefit both beginning and advanced users alike.

SE-FIT has been developed as an open-source program to be released to general public. A website dedicated to SE-FIT has been created (http://se-fit.research.pdx.edu) and users are invited to contribute to the ongoing development of the program.

Acknowledgments

SE-FIT was initially developed by authors Y.C. and M.M.W. as a means to speed employment of SE as a tool to teach capillary phenomena in under/graduate courses at Portland State University. The work was accelerated thanks to support and input by NASA through the Exploration Technology Development Program Office, Cryogenic Fluid Management project, under contract NNC08QA05P. The authors would also like to thank Lihong Yang and Ben Semerjian for early contributions and to the many PSU students who served as the first users of SE-FIT in its many versions. Thanks especially to Ken Brakke for SE and assistance compiling it and to Steven Collicott for teaching Y.C. the tricks and tips of SE. Support for OpenFoam computations performed herein was provided by Lars Kiewidt.

References

[1]Finn, R., "Capillary Surface Interfaces," *Notices of the AMS*, Vol. 46, No. 7, 1999, pp. 770, 781.

[2]Brakke, K. A., "The *Surface Evolver*," *Experimental Mathematics*, Vol. 1, No. 2, 1992, pp. 141, 165.

[3]Zimmerli, G. A., Asipauskas, M., Chen, Y., and Weislogel. M. M., "A Study of Fluid Interface Configurations in Exploration Vehicle Propellant Tanks," AIAA-2010-1294, *AIAA 48th Aerospace Sciences Meeting*, Orlando, FL, 2010.

[4]Shin, D.-Y., "Fabrication of an Inkjet-printed Seed Pattern with Silver Nanoparticulate Ink on a Textured Silicon Solar Cell Wafer," *J. Micromech. Microeng.*, Vol. 20, 2010, pp. 125003:1, 10.

[5]Drozdov, M., Kauffmann, Y., Carter, W. C., and Kaplan, W. D., "Shape-controlled Nanopores in Single Crystals," *Nanotechnology*, Vol. 21, 2010, 475301:1-6.

[6]Prabhala, B., Panchagnula, M., Subramanian, V. R., and Vedantam, S., "Perturbation Solution of the Shape of a Nonaxisymmetric Sessile Drop," *Langmuir*, Vol. 26, no. 13, 2010, pp. 10717, 10724.

[7]Manning, R. E. Jr. and Collicott, S., "Liquid Plugs in Rectangular Channels under a Traverse Gravity Field," AIAA-2010-1476, *AIAA 48th Aerospace Sciences Meeting*, Orlando, FL, Janurary 2010.

[8]Louvet, N., Hohler, R., Pitois, O., "Capture of Particles in Soft Porous Media," *Physical Review E*, Vol. 82, 2010, pp. 041405:1, 5.

[9]Lewandowski, E. P., Cavallaro, M. Jr., Botto, L., Bernate, J. C., Garbin, V., and Stebe, K. J., "Orientation and Self-assembly of Cylindrical Particles by Anisotropic Capillary Interactions," *Langmuir*, Vol. 26, no. 19, 2010, pp. 15142, 15154.

[10]Berthier, J., Brakke, K., Grossi, F., Sanchez, L., and Di Cioccio, L., "Self-alignment of Silicon on Wafers: A Capillary Approach," *Journal of Applied Physics*, Vol. 108, 2010, pp. 054905:1, 10.

[11]Xiao, R., Enright, R., and Wang, E. N., "Prediction and Optimization of Liquid Propagation in Micropillar Arrays," *Langmuir*, Vol. 26, no. 19, 2010, pp. 15070, 15075.

[12]Weaire, D., Barry, J. D, and Hutzler, S., "The Continuum Theory of Shear Localization in Two-dimensional Foam," *J. Phys.: Condens. Matter*, Vol. 22, 2010, pp. 193101:1, 22.

[13]Chau, A., Régnier, S., Delchambre, A., and Lambert, P., "Theoretical and Experimental Study of the Influence of AFM Tip Geometry and Orientation on Capillary Force," *Journal of adhesion science and technology*, Vol. 24, 2010, pp. 2499, 2510.

[14]Chini, S. F. and Amirfazli, A., "Understanding Pattern Collapse in Photolithograpy Process due to Capillary Forces," *Langmuir*, Vol. 26, no. 16, 2010, pp. 13707, 13714.

[15] Kossom, R., Brown, R., Ungar, E., "Space Station Heat Pipe Advanced Radiator Element (SHARE) Flight Test Results and Analysis," AIAA-1990-0059, *AIAA 28th Aerospace Sciences Meeting*, Reno, NV, 1990

[16]Zimmerli, G. A., Asipauskas, M., Wagner, J. D., and Follo, J. C., "Propellant Quantity Gauging Using the Radio Frequency Mass Gauge," to appear in the AIAA 2011 Aerospace Sciences Meeting Proceedings.

[17]Abkarian, M., Subramaniam, A. B., Kim, S.-H., Larsen, R. J., Yang, S.-M., and Stone, H. A., "Dissolution Arrest and Stability of Particle-Covered Bubbles," Physical Review Letters, Vol. 99, 2007, 188301.

[18]URL: http://www.paraview.org/

[19]Concus, P. and Finn, R., "On the behavior of a capillary surface in a wedge," *Proceeding of National Academy of Science of U.S.A., Applied Mathematical Sciences*, Vol. 63, no. 2, 1969, pp. 292, 299.

[20]Chen, Y., Bacich, M., Nardin, C., Sitorus, A., Weislogel, M. M., "The Shap and Stability of Wall-bound and Wall-edge-bound Drops and Bubbles," *Microgravity Science and Technonlogy*, Vol. XVII, no. 4, 2005, pp. 14, 24.

[21]Chen, Y., Baker, J. A., Jones, J. L., Schaffer, B., Semerjian, B., Tavan, N., Weislogel, M. M., "Shape and Stability of Corner-edge Bound Drops," in preparation.