The Surface Evolver and the stability of liquid surfaces

By Kenneth A. Brakke
Susquehanna University, Selinsgrove, PA 17870, USA
email: brakke@susqu.edu

The Surface Evolver is an interactive program for studying the shapes of liquid surfaces. Recently added features permit the calculation of the Hessian matrix of second derivatives of the energy. The Hessian can be used for fast convergence to an equilibrium, and eigenvalue analysis of the stability of that equilibrium. This paper describes the use of the Hessian by the Surface Evolver, presents some sample stability analyses, and gives some numerical results on the accuracy and convergence of the methods. It is also shown how one can evolve unstable surfaces.

1. Introduction

The Surface Evolver is an interactive computer program for the study of liquid surfaces shaped by various energies and constraints. Energies include surface tension, gravitational energy, surface contact energy, user-definable integrals, and many others. Constraints may be pointwise, such as a free boundary on a container wall, or global, such as fixed volumes. The surface is represented as a collection of triangles connected in an arbitrary topology, permitting the representation of complicated structures such as foams. Since its initial release as a public domain program in 1989, the Evolver has become widely used for such diverse problems as capillary surfaces (Mittelmann 1993), liquid metals (Racz et. al. 1993), foam rheology (Kraynik and Reinelt, 1995; Reinelt and Kraynik, 1995), cell membranes (Michalet et. al. 1994), elastic surfaces (Hsu et. al. 1991), and fuel in spacecraft tanks (Tegart 1991).

The main use of the Evolver has been to minimize the energy of a surface. The total energy is regarded as a function of the coordinates of the vertices of the triangulation, and the methods of gradient descent or conjugate gradient are available, for which a general reference is (Hackbusch 1994). These are first-order methods, requiring only the calculation of the first derivatives of energy and constrained quantities. While very straightforward, gradient methods can be slow to converge. When the energy decreases very little with each iteration, it is difficult to know whether the surface is near a minimum, a saddle point, or just some small gradient configuration.

These problems can be addressed by considering second-order information, the Hessian matrix of second derivatives of the energy with respect to the vertex coordinates. The Hessian can be used in Newton's Method to converge rapidly to an equilibrium, usually in just four or five iterations once one is reasonably close. Furthermore, the lowest eigenvalues of the Hessian reveal the stability of the equilibrium, and the corresponding eigenvectors are the most unstable modes of deformation. These modes can be visualized using the graphics output of the Evolver.

Section 2 gives a brief review of the mathematics of the Hessian, and describes how key features are implemented in the Evolver. Section 3 analyzes a very simple example, a square surface. Section 4 analyzes a spherical bubble, which has a volume constraint. Section 5 discusses stability bifurcation diagrams and how to navigate around them with the Evolver. Section 6 explores the bifurcation diagram of a catenoid surface on parallel rings, how one would determine the critical separation of the rings, and how one could evolve unstable catenoids. Section 7 is an analysis of a two-dimensional wet foam with periodic boundary conditions. Section 8 does a triply periodic minimal surface. Section 9 discusses accuracy and convergence issues. Section 10 concludes with some remarks on future possibilities.

Operational details of the Evolver are rarely mentioned in this paper, but a few Evolver commands are mentioned for ease of reference. For further information, see the Surface Evolver Manual. The complete Surface Evolver package, including source code, manual, and sample datafiles, is available by anonymous ftp from susqu.edu as /pub/software/evolver/evolver.tar.Z. The source (in C) can be compiled and run on any unix system. There are also separate executables for DOS and Macintosh systems. A separate PostScript version of the manual is also available in the same directory.

2. Hessian, eigenvalues, and eigenvectors

This section presents some general background on the Hessian matrix, its eigenvalues and eigenvectors. For a good general introduction, see (Strang 1988), especially chapter 6.

The fundamental variables in the Evolver are the coordinates of the vertices of the triangles that constitute the surface. In Evolver's *linear* mode of operation, each triangle is a flat triangle determined by three vertices, but in *quadratic* mode, each triangle is a quadratic surface also depending on vertices at the midpoints of edges. It is convenient to represent all of the coordinates together in a single vector X, which may have thousands of components. When considering perturbations of a particular surface, we will take X to represent displacements from the current positions. The energy of the surface may be expanded in a Taylor series,

$$E(X) = E_0 + G^T X + \frac{1}{2} X^T H X + \dots,$$

where G is the gradient vector, whose components are the first partials of $E(X)$, and H is the Hessian matrix, whose components are the second partials of $E(X)$. Note that H is a real symmetric matrix, and usually very sparse.

Each iteration of ordinary gradient descent uses a motion

$$X = -\lambda G,$$

where λ is chosen by a line search to minimize $E(X)$. This is guaranteed to reduce the energy each iteration, until a local minimum is reached (or a saddle point, if one is unlucky). Convergence is slow, but can usually be sped up by the conjugate gradient method. However, much more rapid convergence can be obtained by retaining the quadratic Taylor approximation,

$$E(X) = E_0 + G^T X + \frac{1}{2} X^T H X,$$

and solving for the critical point,

$$grad\, E(X) = G^T + X^T H = 0,$$

or

$$X = -H^{-1} G.$$

Using such an iteration is called *Newton's method.* It requires the calculation of H and the solution of the sparse matrix equation $HX = -G$. One such iteration may be done in the Evolver with the `hessian` command. Note that `hessian` seeks only an equilibrium point, $G = 0$, not necessarily a minimum.

One drawback of using `hessian` is that the surface must be near enough to a critical point for the quadratic approximation to be a good approximation. Using `hessian` too far from an equilibrium will give nonsense results, sometimes wrecking the surface. Hence some form of gradient descent usually must be used to get a reasonable approximation of the critical point.

Another problem that arises is that the Hessian H may be singular. There are two common causes for this. One is symmetries of motion, i.e. rotations and translations. If X represents one of these, then $E(X) = E_0$ and $HX = 0$. The other common cause is vertices in flat parts of the surface. Moving such vertices tangentially does not change the energy, again causing a singular Hessian. The Evolver is able to cope with singular Hessians as long as the gradient G is in the range space of H, so that $HX = -G$ is solvable.

However, tangential motions in curved surfaces give rise to nearly singular Hessians. Furthermore, it is observed experimentally that it is the tangential motion of vertices that prevents successful use of `hessian` very far from an equilibrium. The vertices want to form exactly the right triangulation, even if the improvement in energy is miniscule. To avoid this, the Evolver has a mode of operation called `hessian_normal` in which vertices are constrained to move only along a vector perpendicular to the surface. Use of `hessian_normal` greatly relaxes the needed proximity to an equilibrium, and should always be used unless there is a definite reason not to. It will be used in all examples below unless otherwise stated.

Once at an equilibrium point ($G = 0$), the stability of the equilibrium may be analyzed via the eigenvalues of the Hessian. Recall that an eigenvalue λ is a scalar and an eigenvector X is a vector that satisfy

$$HX = \lambda X.$$

Any negative eigenvalue $\lambda < 0$ corresponds to a perturbation which decreases energy:

$$E(X) = E_0 + \frac{1}{2} X^T H X = E_0 + \frac{1}{2} \lambda X^T X < E_0.$$

Since H is a real symmetric matrix, there is an orthogonal basis of eigenvectors. Hence the equilibrium is stable if all eigenvalues are positive (H is positive definite), and unstable if there are any negative eigenvalues (H is indefinite). The eigenvectors associated with an eigenvalue form a subspace, and the dimension of the subspace is called the *multiplicity* of the eigenvalue. The total multiplicities of all the negative eigenvalues is the *index* of H, and is essentially the number of independent ways the energy may decrease. The triplet of total negative, zero, and positive multiplicities is called the *inertia* of H. By Sylvester's Law of Inertia, any matrix A that is *congruent* to H, i.e. $H = P^T AP$ for some nonsingular matrix P, has the same inertia as H. Hence the inertia of H may be found by factoring $H = U^T DU$, where D is diagonal and U is upper triangular, without explicitly finding any eigenvalues. The Evolver's implementation of Newton's method involves such a factoring, so knowledge of the index is essentially a free byproduct.

The Evolver's eigenvectors may be interpreted as physical modes of vibration of a membrane if mass is introduced to make dynamics applicable. Let X be position and $\dot{X}$ be velocity. Then the total energy is

$$E = E_0 + \frac{1}{2}X^T HX + \frac{1}{2}\dot{X}^T M\dot{X},$$

where M is a symmetric matrix that plays the role of mass. Standard dynamics gives the corresponding equation of motion

$$-HX = M\ddot{X},$$

which is Newton's Law of Motion, $F = ma$. If X is a *generalized eigenvector*,

$$HX = \lambda MX,$$

then a positive λ gives an oscillatory solution,

$$X(t) = \cos(\sqrt{\lambda}t)X,$$

and a negative λ gives an exponentially growing solution,

$$X(t) = e^{\sqrt{-\lambda}t}X.$$

It can easily be shown that the generalized eigenvalues are the ordinary eigenvalues of a matrix congruent to H, so the inertia is the same and the choice of M does not affect stability. Standard eigenvectors take M to be the identity matrix, which corresponds to a unit mass at each vertex. Since triangulations can be very irregular, this is not a very good approximation to the mass distribution of a smooth surface, and the eigenvalues will change with the triangulation. A more informative choice is to take an M that does approximate the mass distribution of a smooth surface. This is done through defining $X^T MX$ by interpolating X over the facets and integrating with respect to area. Recall that X is composed of a vector at each vertex, and for each facet, the vectors at its vertices can be interpolated across the interior of the facet to give a vectorfield $\hat{X}$. Then we define

$$X^T MX = \int \hat{X}^T \hat{X}\, dA.$$

In linear algebra terminology, M is called a metric, since it is used to define the inner product of vectors, $< X, X >= X^T MX$. In Evolver's linear mode, it has been found that more accurate eigenvalues result from concentrating half of the mass at the vertices instead of having it all evenly distributed over the surface.

Table 1. *Eigenvalues for square surface at several refinements.*

		Linear mode		
ideal	41 vertices	145 vertices	545 vertices	2113 vertices
2	2.0365492403543	2.0099742163701	2.0025688911659	2.0006483680965
5	4.9597203065508	4.9994990426854	5.0005039961225	5.0001661112756
5	4.9597203065508	4.9994990426854	5.0005039961225	5.0001661112756
8	7.7646341433346	7.9853849437890	7.9990889953386	7.9999430908325
10	10.0987980693166	10.0901564190790	10.0268080202750	10.0069731552839
10	10.4997170691025	10.1865249154711	10.0519390576227	10.0134019138280
		Quadratic mode		
ideal	41 vertices	145 vertices	545 vertices	2113 vertices
2	2.0143130550007	2.0009834270978	2.0000639512383	2.0000040664883
5	5.2654046952594	5.0174195838750	5.0011615965936	5.0000744115833
5	5.2654046952594	5.0174195838750	5.0011615965936	5.0000744115833
8	9.4936356610833	8.0566039391621	8.0039725121210	8.0002582433143
10	10.5839769876880	10.1052358235150	10.0078121665966	10.0005196327577
10	11.0713290521096	10.1082736933458	10.0080916241102	10.0005313858573

Quadratic mode does not benefit from this, so the quadratic metric uses only pure quadratic interpolation. The net result of using the metric is that eigenvalues and eigenvectors are nearly independent of the triangulation, and approach those of the continuous surface as the refinement increases.

When there are constraints on the surface, such as a volume constraint, all perturbations X obey the constraint, and the proper Hessian is the energy Hessian projected to the constraint tangent space. Further, each nonlinear constraint (such as volume) contribute an adjustment to the Hessian that is proportional to the Lagrange multiplier for that constraint. The Evolver automatically takes care of all this, and automatically solves for Lagrange multipliers during a `hessian` iteration.

3. Square surface

The first example is a membrane spanning a square boundary of side length π. A basis of eigenfunctions of the continuous surface is $\{\sin mx \sin ny : m, n = 1, \ldots\}$, and the eigenvalues are $m^2 + n^2$ for $m, n = 1, 2, 3, \ldots$. Hence the lowest eigenvalues are 2,5,5,8,10,10,.... This provides a test for the accuracy of eigenvalue calculations. Table 1 shows results for various levels of refinement and different modes. See section 9 for a discussion of the accuracy of the eigenvalues.

Notice that the multiple eigenvalue 5 remains multiple in the discrete approximation, while the multiple eigenvalue 10 splits. Multiple eigenvalues are usually due to symmetries, here the symmetry of the square. Figure 1 shows two eigenvectors for eigenvalue 10, one with full square symmetry and one without, which have different discrete eigenvalues. One can use this effect to advantage to get a nice-looking set of basis vectors for an eigenspace. If the discrete eigenvalue is multiple, one can perturb the problem slightly (make the square slightly rectangular, say) in order to break the symmetry and perhaps get a more comprehensible

Table 2. *Eigenvalues for sphere without* `hessian_normal` *mode, linear model.*

50 vertices	194 vertices	770 vertices
0.0000000000000	-0.0000000000001	-0.0000000000003
0.0000000000000	-0.0000000000001	-0.0000000000003
0.0000000000000	0.0000000000000	0.0000000000000
0.0000000000000	0.0000000000000	0.0000000000000
0.0000000000000	0.0000000000000	0.0000000000000
0.0000000000000	0.0000000000003	0.0000000000005
0.1268571808241	0.0326492834861	0.0081786702183
0.1268571808241	0.0326492834861	0.0081786702183
0.1542354144874	0.0392714891707	0.0098453504444
0.1542354144913	0.0392714891708	0.0098453504444

set of eigenvectors. Left to its own devices, the Evolver will produce a random orthogonal basis of eigenvectors for an eigenspace.

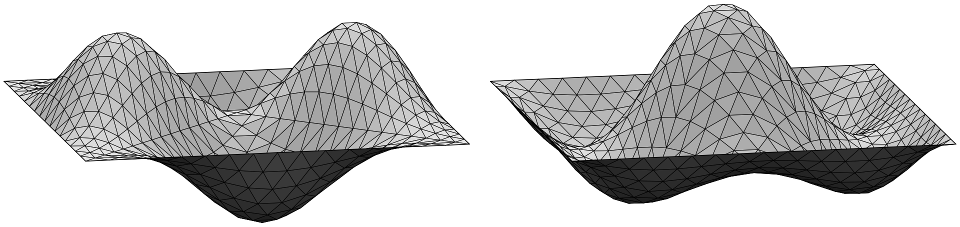

Figure 1. Symmetry breaking of multiple eigenvalue due to discretization.

4. Spherical bubble

This example is a spherical surface of radius 1 enclosing a fixed volume of $4\pi/3$. This is another case where the eigenvalues are known, being $n(n+3)$ with multiplicity $2n+3$ for $n = 0, 1, \ldots$, but the main purpose of this example is to show how `hessian_normal` mode affects singular and near-singular Hessians.

Without `hessian_normal` mode, when every vertex has complete freedom of movement, the lowest eigenvalues are listed in table 2 for several refinements. The first six eigenvalues in table 2 are three translational and three rotational degrees of freedom. The next eigenvalues represent tangential deformations. There is a plethora of these small eigenvalues, which cause problems in convergence, and which disguise the eigenvectors we are really after. The problem gets worse with successive refinements. In linear mode, for an N-vertex triangulation, there are about $2N$ eigenvalues of tangential modes more or less evenly distributed between 0 and 2, while the first interesting eigenvalue is at 4. In quadratic mode, the problems are even worse. All these problems are cured, however, in `hessian_normal` mode.

Table 3 shows eigenvalues for a sphere with `hessian_normal` mode in effect. By permitting only normal motion, all the rotational and tangential modes have been eliminated. The spherical eigenvalues are more apparent. One might wonder why the translational eigenvalues have become nonzero. The answer is that a

Table 3. *Eigenvalues for sphere with* `hessian_normal` *mode, linear model.*

ideal	50 vertices	194 vertices	770 vertices	3074 vertices
0	0.1002018756377	0.0262263658891	0.0066227945708	0.0017502955852
0	0.1002018756377	0.0262263658891	0.0066227945708	0.0017502955853
0	0.1002018756377	0.0262263658891	0.0066227945708	0.0017502955854
4	4.0212279424347	4.0310354860744	4.0093668727492	4.0021201809839
4	4.0212279424347	4.0310354860744	4.0093668727492	4.0021201809839
4	4.0212279424347	4.0310354860744	4.0093668727492	4.0021201809839
4	4.0937433036690	4.0380985434588	4.0104226479940	4.0043427060926
4	4.0937433036690	4.0380985434589	4.0104226479940	4.0043427060926
10	9.1896788636549	9.9557516924035	9.9987386011587	9.9986176742591
10	9.5106749007852	9.9996538541661	10.0081604521679	10.0049321180465

translation of the sphere has to be done with individual normal motions, leading to a distortion of the triangulation, which raises the energy. Figure 2 shows such a deformation.

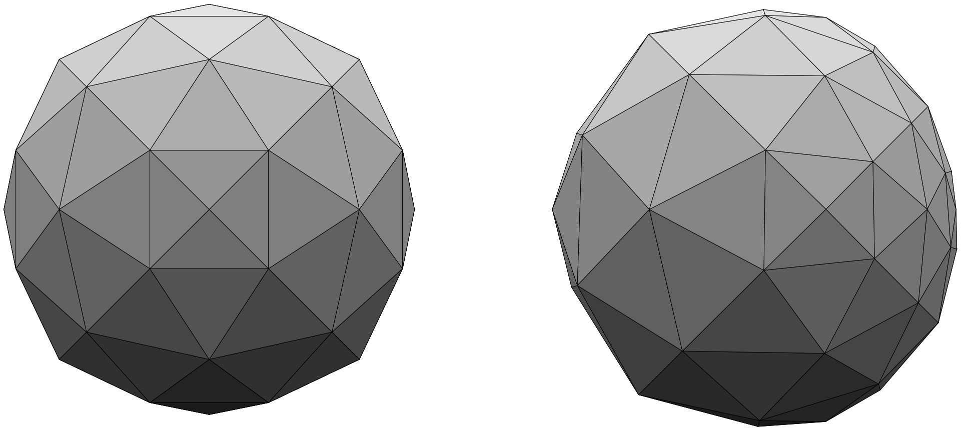

Figure 2. Triangulation deformation due to translation.

5. Bifurcation diagrams

A convenient way to visualize the stability properties of a surface is with a bifurcation diagram, such as figure 3. The independent variable is some control parameter, such as liquid volume or physical dimension. The vertical axis is some interesting shape parameter describing the response of the surface to the control parameter. Usually the interesting shape is along the mode of the lowest eigenvalue. On the diagram are plotted the possible equilibrium configurations, which form curves. Curves of stable equilibria are shown solid, and unstable equilibria are dashed. In figure 3, for low values of the control parameter there is one stable equilibrium, and for high values there are two stable equilibria separated by an unstable equilibrium. The most interesting point is where the equilibrium curves intersect, called a *bifurcation point*. At the bifurcation point, or anywhere there is a transition from stability to instability, the Hessian must have a zero eigen-

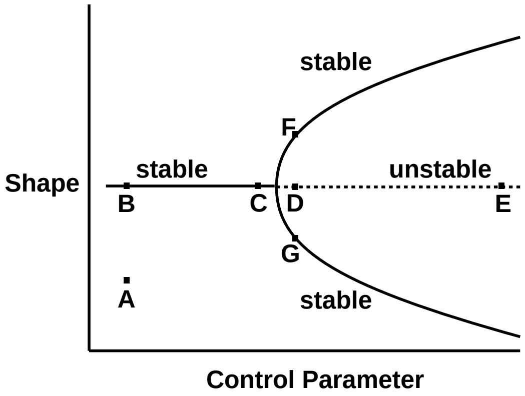

Figure 3. "Pitchfork" bifurcation diagram.

value. Hence one can find the bifurcation point with the Evolver by following the stability curve toward the bifurcation point and watching the lowest eigenvalue. For more background on stability and bifurcation diagrams, see (Arnol'd 1992) or (Hale and Kocak 1991).

It is possible to use the Evolver to navigate around the bifurcation diagram. The user designs the problem so the control parameter can be changed manually in the Evolver to move horizontally on the diagram. Gradient descent moves vertically on the diagram, toward a stable equilibrium. `Hessian` iteration moves vertically toward the nearest equilibrium of any type. Moving along the eigenvector of the lowest eigenvalue is another way to move vertically. This can let the user control which branch of the diagram the surface is on. It is possible to trace along an unstable equilibrium curve by changing the control parameter slightly and using `hessian` iteration only. Controlling one's position around a bifurcation point is trickier, but generally can be done by using the control parameter to move horizontally and the eigenvector of the lowest eigenvalue to move vertically.

As an example of the technique, consider navigating around figure 3. The horizontal stable-unstable line might represent a symmetric shape, and changing the control alone parameter may preserve symmetry. Suppose one starts with a non-equilibrium surface at point A of figure 3. Gradient descent and then `hessian` gets the surface to point B. Then the control parameter is increased gradually, reconverging the surface each time, until point C is reached. The next increase of the control parameter lowers the lowest eigenvalue below zero, so one is in the unstable zone. Using `hessian` iteration only, the surface gets to point D, an unstable equilibrium. One could keep changing the control parameter gradually and use `hessian` iteration only until the surface gets to point E or beyond. Or, from point D, one could use the eigenvector of the lowest eigenvalue as a direction of motion and seek in that direction for the lowest energy. This would wind up near points F or G, and gradient descent and `hessian` could be used to converge exactly to F or G. From F, one could likely use the eigenvector of the lowest eigenvalue to reach the neighborhoods of D or G.

6. Catenoid

The catenoid is the soap film that forms between two parallel rings that are not too far apart. The control parameter in this system is the height between the rings. For rings at less than a critical height H_c, there are three equilibrium surfaces: a stable thick-necked catenoid, an unstable thin-necked catenoid, and a stable pair of disks. For rings farther apart, only the pair of disks exist. The bifurcation diagram is sketched in figure 4. This diagram illustrates a *catastrophe*. If one starts with a stable catenoid and pulls the rings farther apart, the surface moves to the right along the stable branch. But as the height passes the critical height H_c, there is no continuous path along a stable branch available. The surface falls down to the two-disk stable branch, i.e. the catenoid pops.

From the form of the stability diagram near the catastrophe point, it may be expected that the height parameter is a quadratic function of the lowest eigenvalue, i.e. the neck radius is monotone with respect to the lowest eigenvalue.

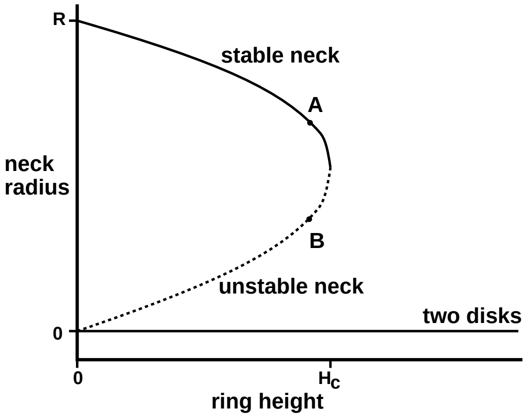

Figure 4. Catenoid stability diagram.

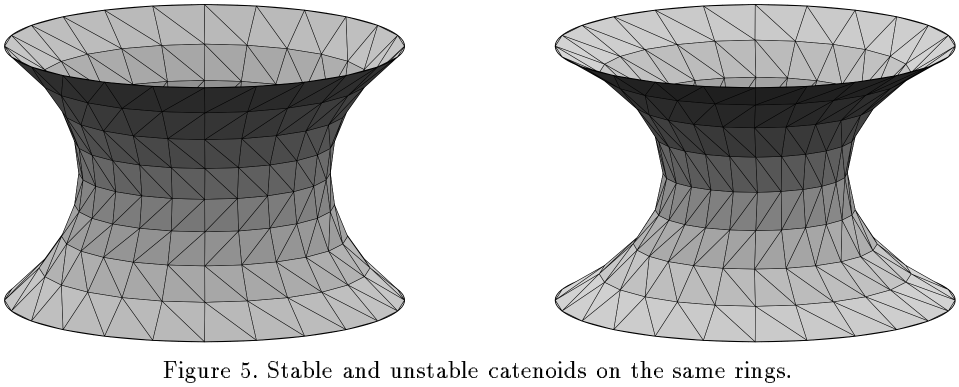

Figure 5. Stable and unstable catenoids on the same rings.

A stable and unstable catenoid for the same rings are shown in figure 5, where the ring separation is $0.98H_c$. The unstable catenoid was generated by first evolving the stable catenoid (point A in figure 4), finding the eigenvector of the lowest eigenvalue, moving along that eigenvector in the thin-necked direction to somewhere near the unstable catenoid (which is a maximum of energy in the search

Table 4. *Eigenvalues for catenoid near instability, with extrapolated values of the critical height.*

Linear model					
60 vertices		216 vertices		816 vertices	
height	lowest eigenvalue	height	lowest eigenvalue	height	lowest eigenvalue
0.95	0.7843072020538	0.97	0.6794680620506	0.995	0.2613684324282
0.97	0.4975872956970	0.98	0.5285333398723	0.997	0.1821410586137
0.98	0.2157674752854	0.99	0.3181538066139	0.998	0.1236805711725
0.9811	0 (extrapolation)	0.9964	0 (extrapolation)	0.99878	0 (extrapolation)
Quadratic model					
216 vertices		816 vertices		3168 vertices	
height	lowest eigenvalue	height	lowest eigenvalue	height	lowest eigenvalue
0.997	0.2354964551417	0.997	0.2286534698147	0.997	0.2282417834338
0.998	0.1960813689387	0.998	0.1871625778573	0.998	0.1866111210436
0.999	0.1465675481787	0.999	0.1331531751143	0.999	0.1322838713261
1.00029	0 (extrapolation)	1.000008	0 (extrapolation)	0.999991	0 (extrapolation)

direction), then using `hessian` iteration to converge to the unstable catenoid (point B). The unstable catenoid has index 1, and the corresponding mode leads to the stable catenoid in one direction and the pair of disks in the other direction.

Table 4 shows some eigenvalues for discrete surfaces around the bifurcation point. At each refinement there is a critical height slightly different from H_c. To find this critical height, I have chosen to use the quadratic relation between the control parameter and the lowest eigenvalue to extrapolate the critical height. At the critical height H_c there seems to be no discrete equilibrium in the linear model. Further experiments at H_c verified this. Attempts to use `hessian` iteration just bounce back and forth, like trying to find a root of x^2+1 with Newton's Method. Sometimes the Hessian is positive definite, and sometimes it has index 1. So even getting a positive definite Hessian does not mean the surface is near a minimum. In the quadratic model, sometimes a discrete equilibrium does exist at H_c, and sometimes it does not. At a particular refinement, sometimes the existence of an equilibrium at H_c depends on the particular history of the evolution.

When an equilibrium does not exist (for example, immediately to the right of the catastrophe point in figure 4), one can find a near-equilibrium in the sense of minimizing the square of the gradient of the energy. The Evolver can find such surfaces by using `hessian` iteration with adjustable step size as a gradient descent method for square-gradient.

Note that due to the quadratic relation between control parameter and eigenvalues at the catastrophe point, very slight shifts in the control parameter can give a large change in the eigenvalue. Furthermore, discretizing the surface shifts the equilibrium curve slightly, further damaging the relation between discrete eigenvalues and smooth eigenvalues. Hence a general a priori estimate on relative accuracy of eigenvalues is not possible. However, the presence and location of a catastrophe or bifurcation point are relatively stable, and that is the real target, rather than exact eigenvalues.

7. Wet foam

Foams are collections of bubbles. Wet foams contain enough liquid to form appreciable "Plateau borders," which are thickenings of surface junctions. Foams behave as non-Newtonian liquids, with an intricate stress-strain relationship and topological changes during movement. The Evolver has been used to investigate foams in two and three dimensions (Kraynik and Reinelt, 1995; Reinelt and Kraynik, 1995; Phelan et. al. 1995). To simplify matters, the foams are usually taken to have periodic boundary conditions with a managable number of bubbles. The Evolver has a flat torus mode which enforces periodicity. The fundamental cell may be any parallelogram, and its shape may be varied at runtime (which is useful for putting strain on a foam). Thousands of bubbles are possible in two dimensions, and a few hundred in three.

As foams evolve, they can pass through unstable critical points, and it is reasonable to assume that the further evolution follows the most unstable mode. As a very simple example, consider a two-dimensional wet foam with periodic boundary conditions and just a few cells in a regular square lattice.

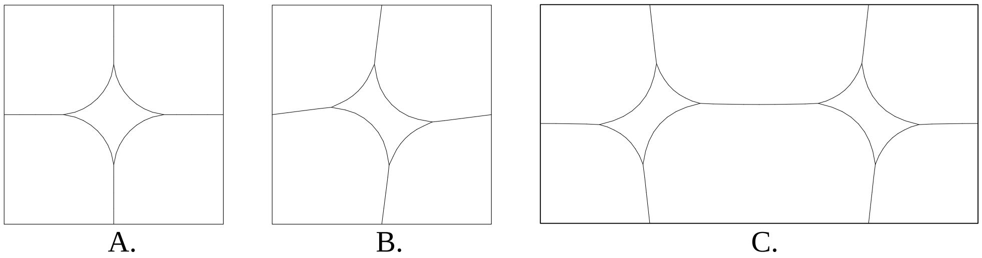

Figure 6. Foam and instability modes.

The fundamental cell of a single bubble foam is shown in figure 6A. This configuration has index 1 for a liquid fraction less than a critical value (which in this simple case can be calculated to be $(4-\pi)/16 = 0.0536...$), and the corresponding unstable mode is shown in figure 6B. However, if the fundamental cell is doubled, then the index becomes 2, and a second unstable mode appears, shown in figure 6C. This second mode has a lower eigenvalue than the first, and is unstable at all liquid fractions. For larger fundamental cells, this remains the most unstable mode. This mode represents a sliding of columns of cells towards a hexagonal configuration. However, the Evolver cannot follow the evolution all the way to the hexagonal configuration because (at the time of this writing) it lacks detection of films running into other films.

8. Triply periodic minimal surfaces

A triply periodic minimal surface is a minimal surface that is periodic in three directions. Such surfaces have been observed when two materials form interpenetrating labyrinths and in cell membranes. As area minimizing surfaces, they are necessarily unstable, so they must owe their existence to some other phenomenon. Whatever the reason for their natural existence, their instability poses a problem for modelling them with the Evolver. There are several techniques available:

(i) Use symmetry to work with a piece of surface small enough to be stable. This will work if the modes of instability do not share the symmetry of the surface.

(ii) Use a volume constraint to eliminate one mode of instablility. Unfortunately, this will eliminate only one mode. Eliminating more modes would require more constraints, and I am not aware of any that would be naturally occurring.

(iii) Define the energy to be the integral of the squared mean curvature, rather than area. This will be a minimum by definition for a minimal surface. In theory, one can always define the square of the gradient of the original energy to be a new energy, which is minimized at any equilibrium of the original energy. There is recent evidence that this type of energy does actually account for some biological membrane shapes (Michalet and Bensimon 1995).

(iv) Use only `hessian` iteration, which converges to an equilibrium regardless of stability. In fact, `hessian` iteration may be viewed as a gradient descent step for a certain form of squared gradient energy.

As an example, consider Schwarz' P surface, the cubical unit cell of which is shown in figure 7A. This surface divides space into two congruent labyrinths. With periodicity enforced by Evolver's flat torus mode, it turns out that the P surface has only one unstable mode, which is a more or less uniform expansion of one labyrinth, shown in figure 7B. Hence a volume constraint is enough to suppress this mode and make the P surface stable, since the gradient of the volume is approximately in the direction of the unstable mode.

The symmetry approach can also be used to construct the P surface. Mirror planes can divide the fundamental cube into 48 isometric pieces, each in a tetrahedron with perpendicular contact with the tetrahedron faces. However, such a piece in a tetrahedron is still subject to the instability and needs a volume constraint. There is a further symmetry of 180° rotation about a line, and with this symmetry a piece is stable. Few triply periodic minimal surfaces are lucky enough to have this much symmetry.

In nature, the perturbations of a triply periodic minimal surface are not required to be periodic. To handle perturbations of longer wavelength, the cubic fundamental cell can be replaced with a larger fundamental cell made by replicating the cube in each direction. To facilitate this, the Evolver has a command for duplicating the fundamental cell along any one of the periodic directions. Table 5 shows the lowest ten eigenvalues for several sizes of fundamental cells without a volume constraint. With a volume constraint, the only difference is that the lowest eigenvalue is missing. The others shown remain the same since their modes are all volume preserving.

Figure 7C shows a 2x1x1 fundamental cell deformed by an unstable mode with a volume constraint. The eigenspace actually has dimension 2, with an independent mode formed by translating the perturbation by half the diagonal of the cube. Figure 7D shows a most unstable mode of a 4x1x1 fundamental cell with volume constraint. It appears that for any size fundamental cell, the low eigenvalue modes can be viewed as normal motion multiplied by eigenmodes of the ordinary Laplacian on a flat torus domain. As the fundamental cell gets larger and larger, the lowest modes become locally more and more like the pure normal mode, and hence the lowest eigenvalue approaches the lowest unconstrained eigenvalue.

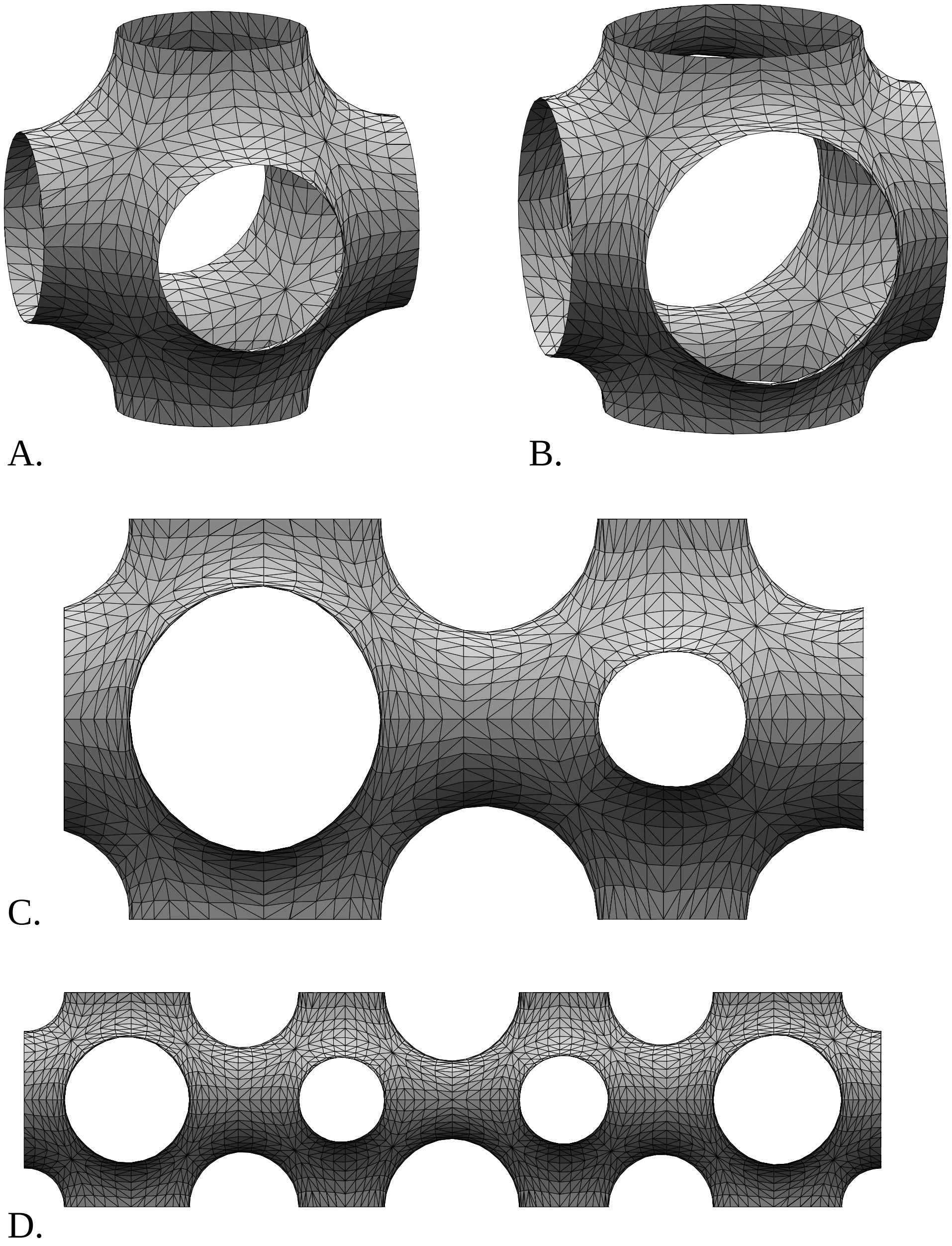

Figure 7. Schwarz' P surface. A. Fundamental cell at equilibrium. B. Mode of instability without volume constraint. C. Mode of instability with volume constraint for doubled fundamental cell. D. Most unstable mode with volume constraint for quadrupled fundamental cell.

9. Accuracy of eigenvalues

It is difficult to give a general statement on how close the eigenvalues of a discrete surface are to those of a smooth surface. Several types of behavior may occur:

Table 5. *Eigenvalues for Schwarz' P surface for various sizes of the fundamental cell without volume constraint. Linear model, 764 vertices per cube unit.*

1x1x1	2x1x1	4x1x1	8x1x1
-1.3751552395274	-1.3751552395274	-1.3751552395274	-1.3751552395274
0.0050557043828	-0.9707120740664	-1.2733834999456	-1.3496765588438
0.0050557043828	-0.9707120740664	-1.2733834999456	-1.3496765588438
0.0050557043828	0.0050557043827	-0.9707120740664	-1.2733834999456
0.5144097462895	0.0050557043828	-0.9707120740664	-1.2733834999456
0.5144097462895	0.0050557043828	-0.4807931744404	-1.1467446537502
1.0859420131168	0.3735205562288	-0.4807931744404	-1.1467446537502
1.0859420131169	0.3735205562288	0.0050557043892	-0.9707120740664
1.0859420132500	0.3735205562352	0.0050557044219	-0.9707120740664
2.1301296386849	0.3735205568893	0.0050557050477	-0.7471956960614

1. The discrete equilibrium may not exist when a smooth solution does. The linear mode catenoid is an example of this.

2. A discrete equilibrium exists, but a smooth equilibrium does not. This happens in some cases of the quadratic mode catenoid.

3. Both discrete and smooth equilibria exist, but their eigenvalues are not close due to sensitive dependence on the control parameter. The catenoid slightly below its bifurcation point is an example.

4. The eigenvalues of the discrete surface are close to those of the smooth surface. This is the usual behavior away from bifurcation points, and is seen in the square and sphere examples. In linear mode, eigenvalues will have errors on the order of h^2, and in quadratic mode errors will be on the order of h^4, where h is the triangle side length. Each refinement of the surface quadruples the number of vertices and halves the edge lengths, so one generally sees a factor of 4 improvement in the linear mode eigenvalues each refinement, and 16-fold improvement in the quadratic mode.

The exact value of the control parameter at a bifurcation point is often the item of most interest. It is clear from the catenoid example that discreteness can change the location of the bifurcation point, and not always in a predictable manner. Hence one should never trust too much in the bifurcation point found at one level of refinement, but should use several levels of refinement to get an idea of the errors involved.

The eigenvectors seem much better behaved than the eigenvalues around a bifurcation point. In all the catenoids, the lowest mode was always rotationally symmetric and changing the diameter of the neck. This is understandable in the sense that the bifurcation really involves only the lowest mode, so all the other modes are slowly changing. Since eigenvectors of a symmetric matrix are mutually orthogonal, that means the lowest mode must also be slowly changing.

The history of evolution can have an effect on the exact values of the eigenvalues, since in `hessian_normal` mode the triangulation does not get to adjust laterally. As an example, table 6 shows eigenvalues for the catenoid with two different histories: evolving with height 1 first and then reducing the height, and evolving with height 0.97 first and then raising the height.

Table 6. *Effect of evolution history on eigenvalues of catenoid, in quadratic model at lowest refinement level.*

height	Lowering height	Raising height	true
0.97	0.7756484301625	0.8016978545934	0.732
0.98	0.6477910223247	0.6764366167894	0.592
0.99	0.4980292782834	0.5313773392421	0.416
1.00	0.2964353302531	0.3403379336293	0

10. Conclusion

We have seen that it is possible to use the Surface Evolver to explore eigenvalues, eigenvectors, stability, and bifurcation diagrams. It is hoped that the Evolver will become a widely used tool for such investigations.

Some future developments of the Evolver are clear. There is always a demand for handling more complicated surfaces, for example realistic foams. Today's workstations can handle tens of thousands of facets, and supercomputers perhaps a million. Parallel versions of the Evolver are available, and will be developed further. For increased accuracy, higher order finite elements are being added. Further additions to the Evolver's capabilities will be made in response to users' experiences.

The really interesting developments will be the unforeseen uses people find for the tools the Evolver makes available.

This work has been partially supported by The Geometry Center, University of Minnesota, Minneapolis, and by the University of Massachusetts, Amherst.

References

V. I. Arnol'd, *Catastrophe Theory*, 3rd ed., Springer-Verlag, New York, 1992.

K. A. Brakke, "The Surface Evolver", *Experimental Mathematics* **1** (1992), 141-165.

W. Hackbusch, *Iterative solution of large sparse systems of equations*, Springer-Verlag, New York, 1994.

J. Hale and H. Kocak, *Dynamics and Bifurcations*, Springer-Verlag, New York, 1991.

L. Hsu, R. Kusner, and J. Sullivan, "Minimizing the squared mean curvature integral for surfaces in space forms," *Experimental Mathematics **1*** (1991) 191–208.

A. Kraynik and D. Reinelt, "Simple shearing flow of a dry Kelvin soap foam", submitted to *J. Fluid Mech.*

X. Michalet, B. Fourcade and D. Bensimon, "Fluctuating vesicles of nonspherical topology", *Phys. Rev. Lett. **72*** (1994) 168-171.

X. Michalet and D. Bensimon, "Observation of stable shapes and conformal diffusion in genus 2 vesicles", *Science **69*** (4 August 1995), 666-668.

H. Mittelmann, "Symmetric capillary surfaces in a cube", *Math. Comp. Simulation **35*** (1993) 139-152.

R. Phelan, D. Weaire, and K. A. Brakke "Computation of equilibrium foam structures using the Surface Evolver", preprint, 1995.

L. M. Racz, J. Szekely, and K. A. Brakke, "A General Statement of the Problem and Description of a Proposed Method of Calculation for Some Meniscus Problems in Materials Processing", *ISIJ International **33*** (1993), 328-335.

A. Kraynik and D. Reinelt, "The linear elastic behaviour of a bidisperse Weiare-Phelan soap foam", submitted to *Chem. Eng. Comm.*

G. Strang, *Linear Algebra and its Applications*, 3rd ed., Harcourt Brace Jovanovich, 1988.

J. Tegart, "Three-dimensional fluid interfaces in cylindrical containers," AIAA paper AIAA-91-2174, 27th Joint Propulsion Conference, Sacramento, CA, June 1991.