

Tools for designing programmatic self-assembling systems.

Saul Griffith.

6.978 final paper, Dec 2002

“...until an adequate theory of automata exists there is a limit to the complexity and capacity of the automata we can fabricate”.

– Von Neumann.

Abstract

It is of interest to add more control and complexity to the processes of self-assembly that we are just beginning to understand. Existing self-assembling systems are essentially meso or macro-scale versions of crystallization. I wish to expand the tool-box of self-assembly to include dynamic components that emulate the behaviour of allostery exhibited in bio-macromolecules. An allosteric molecule can be considered abstractly as a state machine where each stable conformation is a state. Designing the components for self-assembly gets increasingly complicated as part numbers increase, and as the components have re-configurable 'states'. Essentially the problem in designing self-assembling components is to avoid undesirable meta-stable states, and to make the desired assembled geometries the lowest energy conformations of the system. To that end, this work describes the development of simulation tools for the modeling, and eventual evolution of mechanical state machines for programmatic, or predicated, self-assembly.

1.Introduction.

Biology uses many examples of programmed, and self, assembly processes to achieve a remarkable array of structure and morphologies at many scales. Whilst our understanding of self assembling processes is slowly proceeding, there are no non-biological examples of on-the-fly programmable assembly processes for three dimensional structure at any scale. Sub-cellular, uni-cellular, and multi-cellular biological systems all display controlled programming of 3D structure. Commonalities between these biological systems and a “mechanical morphogenesis” are being sought. Self-assembly is oft touted as the route to building next-next generation computers with components at the nanometer scale. I would contend that without more adequate knowledge about more complex self-assembly, and quantitative rules for system information, this will not be possible.

1.Background and Goals.

1a. Self Assembly.

The goal of this work is to expand the current knowledge of 'self-assembly' and to introduce concepts of state and program size complexity into the components such that aperiodic and non-crystalline structures can be produced 'automatically'.

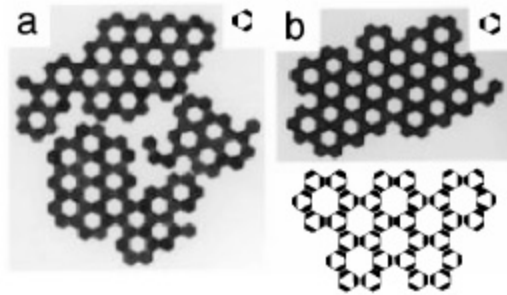


Figure 3. Regular crystalline lattices by self assembly.

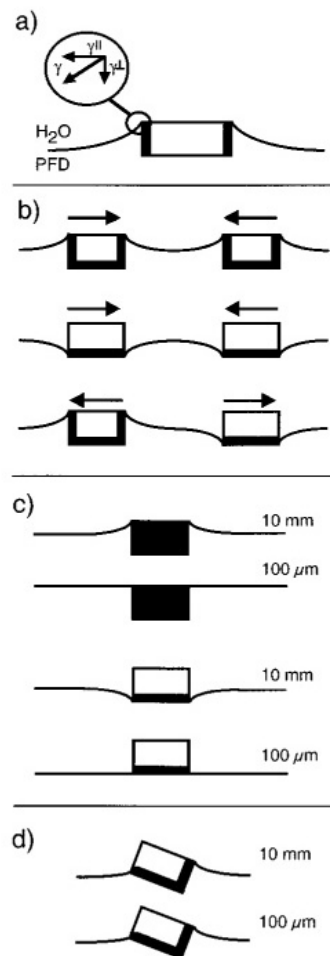


Figure 2. Attractive and repulsive forces by meniscus shape.

The physical system employed borrows from the work of Whitesides et.al. and comprises PDMS components at fluid/fluid interfaces and the minimization of surface energies to converge on an assembled state. An illustration of the forces involved and the menisci generated can be seen in figure 2. (Bowden et.al). The typical self-assembling constructions from such parts can be seen in figure 3. To date this type of self-assembling system has dominated the literature. I see it as limited in some very important ways. Firstly all such systems have significant 'program' limitations in that the individual units have a set state determined prior to assembly. In essence the only thing being programmed is the basic attractive and repulsive forces between given faces. These systems all build periodic crystalline type structures which have inherently limited complexity. In essence all of these systems are seeking out a global energy minima and hence a static equilibrium.

1b. Biology.

Biological systems do much more than this simple type of self-assembly, though crystallization and periodic arrays are also within it's toolkit. Perhaps the most important difference is that biological systems are capable of being non-equilibrium or dynamic systems, and that some biological components may also be thought of to have more than one pre-programmed state. Biology employs the allosteric function heavily in the control of self-assembly and morphogenesis, particularly in sub-cellular (bio-macro-molecule) systems (Ptashne). A schematic diagram of allostery can be seen in figure 1. The various allosteric configurations of a part can in some respects be considered states of a simple state machine. Biology also uses the concept of co-operative binding heavily, but I will not go into much detail on that in this paper.

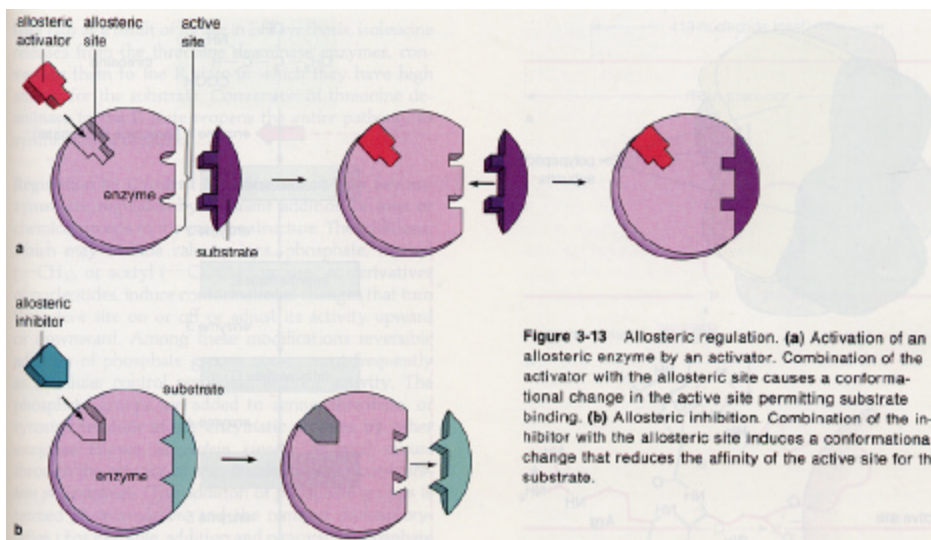


Figure 3-13 Allosteric regulation. **(a)** Activation of an allosteric enzyme by an activator. Combination of the activator with the allosteric site causes a conformational change in the active site permitting substrate binding. **(b)** Allosteric inhibition. Combination of the inhibitor with the allosteric site induces a conformational change that reduces the affinity of the active site for the substrate.

Figure 1. Schematic of allosteric 'state'. Ptashne.

By way of this introduction I have wanted to demonstrate how non-static, 'state' machine type components are an interesting topic of research in self-assembling systems.

1c. Predicated, or programmatic self-assembly and my previous work.

Where the goal of this work is to programmatically assemble 3D structure from a solution or collection of parts, two challenges can be identified.

The first is the set of questions that relate to system design: How many components of what complexity are capable of manufacturing what collection of assemblies with what complexity? How many states required at each node? What is the assembly complexity space for a given set of components? I believe we can borrow heavily from automata theory and CA's in answering these questions, however the systems to be built are inherently non-digital and the concepts of program and state as traditionally defined, are not entirely adequate.

The second set of questions relates to the individual component design: How does one build the required number of states and inter-component interactions into a unit.

I hope to make progress on questions in both of these statements, but in the particular work described in this paper I will risk putting the egg before the chicken and focus on the design of the individual components for assembly.

Every component could have a minimal micro-chip that controls electrostatic or electro-wetting interactions with other parts, and given enough memory and processor for each component an infinite number of assemblies could be produced. Power, expense, and other reasons make this an undesirable method of manufacturing our component parts, and in many ways the redundancy of having a powerful processor in every unit is un-interesting. I have chosen instead to focus on limited state machines as every unit can conceivably be manufactured in very high numbers. This is more akin to sub-cellular assembly of macro-molecules than to multi-cellular assembly.

Previous to this work I developed a 2 bit mechanical state machine that programmatically self assembles at the interface between water and poly-fluro-decalin (Figure 4). The mechanical state machine is a PDMS component with a mechanical flexure that acts as the 'switch' in the state machine. The components can be most easily envisaged as a mechanical allosteric enzyme.

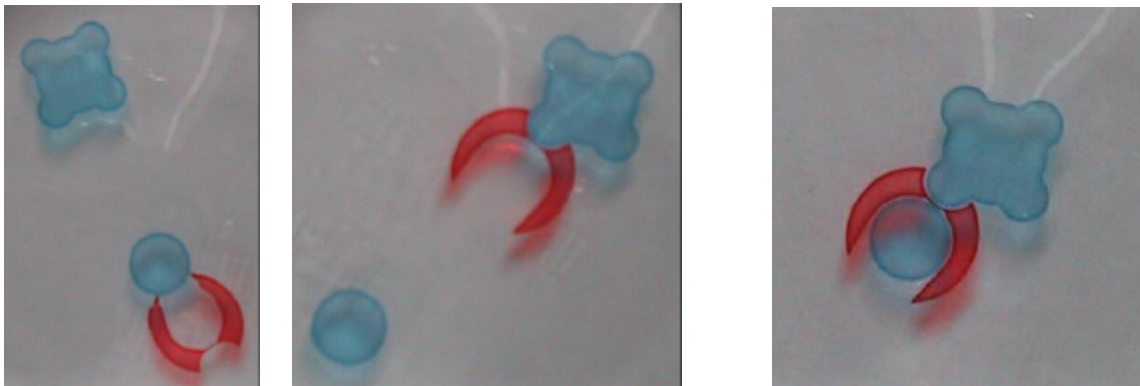


Figure 4. PDMS parts at water/PFD interface, coming together and interacting due to meniscus forces. The allosteric part is the horse-shoe shaped red component that is 'opened', or activated when the catalyst (4-lobed blue part) binds it's functional site. The bound blue circular part can only fit into the horseshoe once the catalyst is bound.

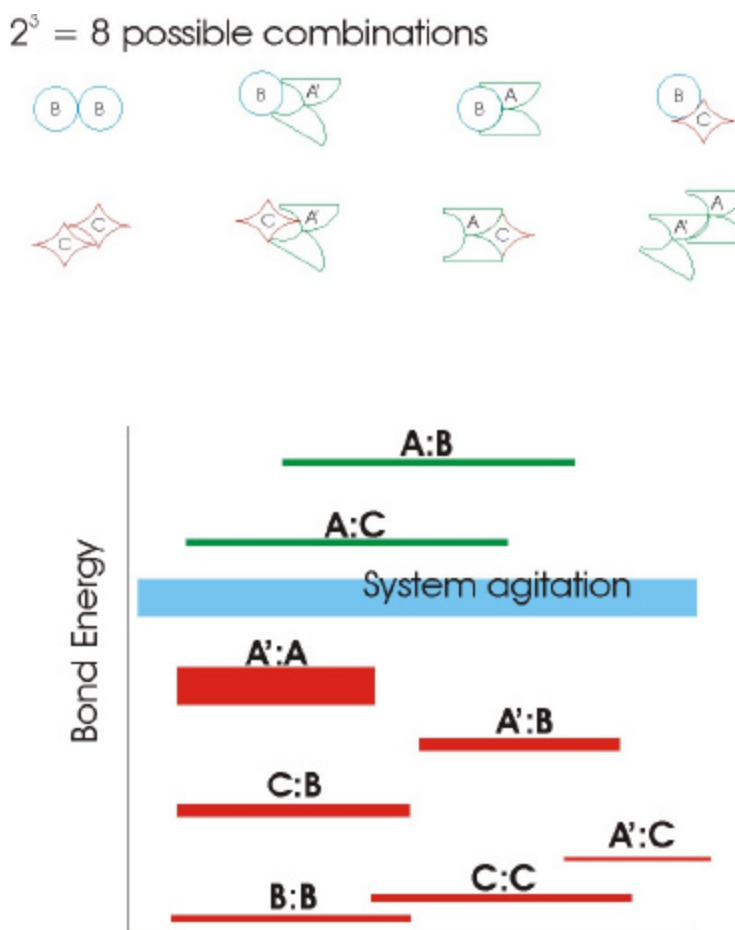


Figure 5. System energy design and component interactions.

I have quickly discovered that the extra complexity of designing state and flexible parts into the self assembling systems dramatically increases the complexity of their design. You will observe in figure 4. that there are no straight edges on the

components to avoid local energy minima on their collision. The essence of designing mechanisms for assembly in this system is designing an energy diagram of the different components geometric alignment with respect to one-another, such that the desired states sit in the deepest possible local energy minima. I have successfully done this for the 2 state system above and am making progress on a 3 state system, however it is a problem of increasing difficulty, and as more component types are added to a system the challenge is to avoid any undesirable local energy minima.

Figure 5 illustrates this point. For the example of a 3 component system where one component has two states, A & A', and the other two components single state, fixed parts, B & C, there are 8 possible combinations of parts. For each combination of parts in a 2D surface interaction there is a continuum of rotational orientations, with a subsequent bond energy relative to orientation curve. The key is to design the components such that desirable interactions A:B and A:C have stable conformations whilst the other bond energies are below the level of the system agitation energy (akin to KT in chemical systems).

I believe there is utility in having modeling tools that can facilitate this design process by automating the generation of the rotational energy diagram for any pair of components. From here it is possible to imagine using genetic algorithms or some other virtual evolutionary mechanism to evolve sets of parts with more optimal interactions and more complex interactions by performing a mechanical mutation on the part shapes at each generation. This paper describes the development of some of these tools for modeling the self-assembly of simple components at the interface of a liquid.

3.The Physical Model.

The goal for developing a working physical model of the self assembling systems I am interested in is to have an engine for designing and analyzing components and their interactions. I have employed heavily the excellent program 'Surface Evolver' of Kenneth Brakke. This program was developed for modeling soap films, mathematical geometries and solder wetting and bond formation. It uses the gradient descent method to evolve the minimal energy surface for a given and defined starting geometry. Surface Evolver may be found at: <http://www.susqu.edu/facstaff/b/brakke/evolver/evolver217.html> There is excellent documentation at that site and I refer those interested to read more.

3.1.Defining and importing geometry.

Surface evolver utilises a vertice, edge, face, body, hierarchy of geometric definition. 2D, 3D cartesian, 3D polar, string, and 4 and higher dimensional coordinate systems may be employed. Each edge, face, and body must carefully be defined within a convention such that surface normals are aligned and faces are oriented and bodies given mass. In the early part of this work I was able to

manually generate simple test geometries to execute in evolver. As part of the project I developed MATLAB code for automatically importing geometry from 3D cad programs (eg. Rhino) via the .raw (Raw Triangles) data export format. This code generates vertices, edges, and faces, in a surface evolver readable format, however the geometry files still require editing to constrain appropriate parts, and to define interfacial surface energies, and body masses. This is further work to do.

3.2.The datafile.

Surface evolver operates on a data file. This datafile includes the geometric definition of the parts, constrained geometry, contact angles between faces (a constraint), surface tensions, gravity, mass and centers of mass, optimizing parameters for running, and calls for functions (.cmd files) that implement the updating of position etc. as they file is run. An example datafile, duohex4.fe, can be found in Appendix A.

3.3 The .cmd files.

The command file which is called from the datafile is essentially a subroutine for recalculating the forces, volumes, geometric constraints, or other components of the datafile. These command files look very much like the C programming language. An example command file, xyztorque.cmd, can be found in Appendix B. In this example (xyztorque.cmd is the file called from duohex4.fe), the x,y,and z forces, and the torque, on the floating components, are calculated by the central difference method. Each physical system being modeled requires an appropriate command file which appropriately simulates the physical behaviour that one is wishing to model.

3.4 Validating the Surface Evolver physical model.

Surface evolver is typically used to model the equilibrium surfaces of, for example, a soap film on a wire frame. A lot of massaging of the datafiles and especially the command files was required to model this system. Before using the model as a predictive tool for designing components I have verified it by modeling interactions in a simple well constrained case which I can compare to physical experiments. For this simple case I am modeling the assembly of two hexagonal components on a petri dish surface with different contact angles, masses, and wetted faces.

Example 1 (figure 6) is the case of two parts which only wet on their bottom surface (top surface is not shown in this model). The components can be seen to come together iteration by iteration until their edges touch. Quantitatively the resulting minimum energy surface of the petri dish appear similar to those for mm sized hexagons at a water, air interface.

Example 2 (figure 7) is the same model modified slightly such that the sides of the hexagonal prisms also wet and hence the top face is that in contact with the air water interface. Again, the components can be seen to come together as the surface iterates. This is again reflective qualitatively of a similarly patterned system in a real dish.

Figure 8 is the same as example 1 except that the inter-component spacing has been increased. These parts do not converge as their menisci do not interact.

Figure 9 is the same example again, with the mass of the component increased beyond a point where surface tension will buoy it. It illustrates the point that the model does not rebuild the surface after the component sinks. It can be used however to model the scale vs weight issues for parts to get an idea of the instabilities in real systems.

Changing the scale of the components has similar effects such that when they are very large they converge slowly relative to their size and positions.

These examples are enough to give confidence in the physical model as representative of the physical system and therefore as a useful engine in design of more complicated parts.

Another detail of the model is illustrated well in figure 9. The facets between parts do not disappear as the parts approach each-other in the current model. In one respect this is good as it effectively does edge detection of the parts and prevents them physically overlapping. The problem is that these facets are fixed and end up pinning the parts at an orientation prematurely. In the real physical system, these two parts will converge until these two faces line up. In future iterations of the model I will regenerate these facets as they approach each-other to allow them to converge further to the expected structure.

So far I have described the generation of dynamic models that allow the user to observe the interactions as the model runs. I believe in retrospect that this is not the optimal way to generate the energy diagrams that I wish to have for one component relative to another as a function of total system energy vs. angular orientation of the two parts. To do this I have generated MATLAB code for the importing of raw triangle files from a 3D CAD program. (Appendix C). One can generate a series of these files by rotating the parts automatically within the CAD environment and upon import evolve them to an equilibrium point and sum the energy over all facets in the system to get a quantitative measure of surface energy for each orientation. These can be used to generate energy vs. theta curves. So far I can implement this manually on hand coded geometries, but have not as yet completely automated the import process.

4. Results, Conclusions.

Surface evolver was used as the scaffolding for building a physical model of self assembly at a 2D interface by meniscus forces. Using a well studied system I was able to produce qualitative results indicating that the model is valid and representational of real systems. I determined the principal limitations in the current system. In retrospect I think the dynamic model attempted is of less utility than a static model that merely evolves and measures the equilibrium energy for a system of components and their relative orientations. By running a series of these models I will have energy vs. orientation diagrams for pairwise part interactions that can either inform the manual design of the next generation of components, or in the future for the computer evolution of these parts. In summary useful tools were developed for modeling these systems and as aids to design of future mechanical implementations of the ideas outlined earlier in this paper.

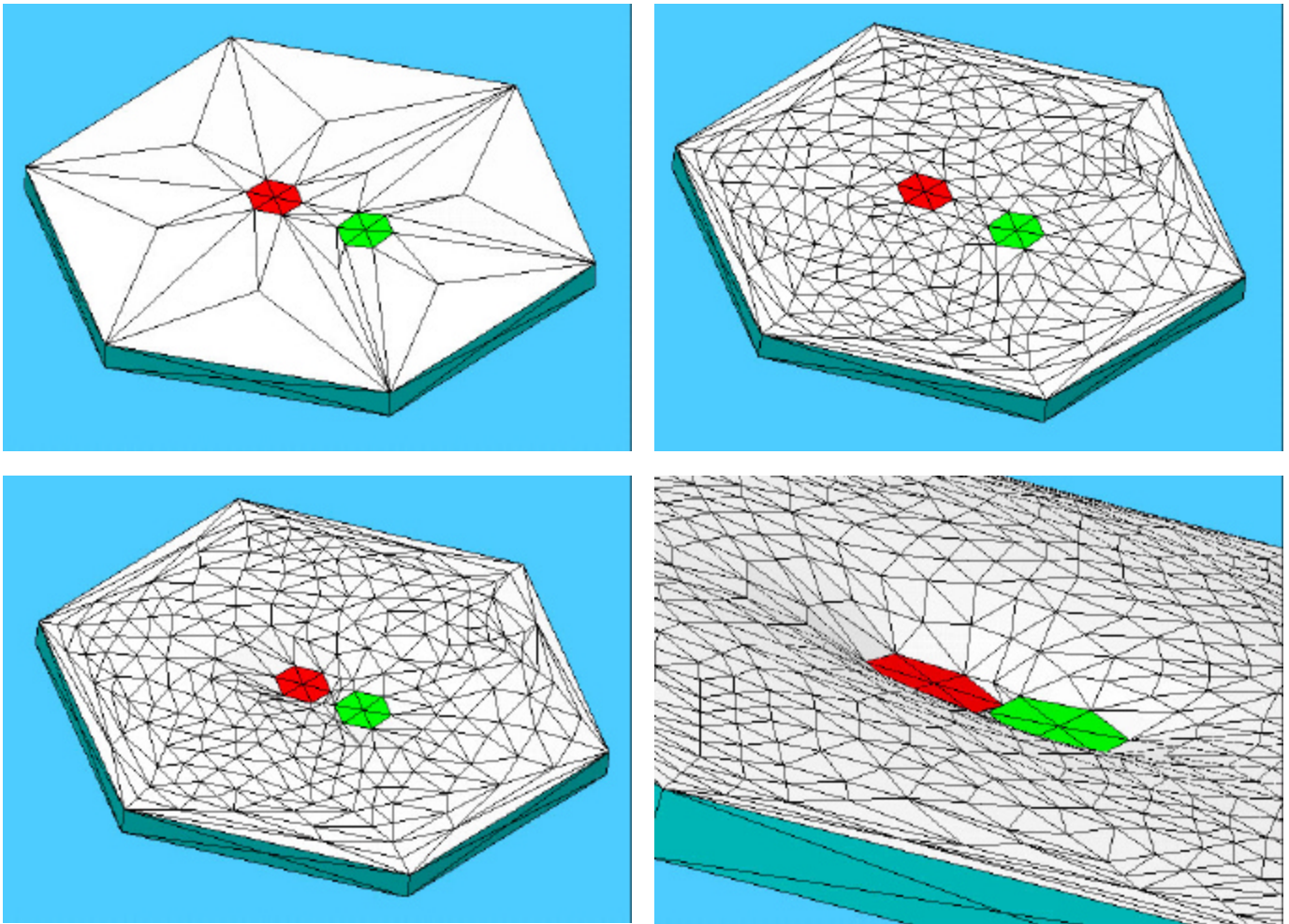


Figure 6: Example 1: Evolution of minimum energy surface for 2 hexagons floating on an hexagonal petri dish with mass acting at their center of gravity, with one wetting face (in contact with fluid) with a contact angle of 35 degrees. Hexagons are given an initial orientation, position, and tilt angle relative to the absolute so-ordinate system. Sides and top of hexagon are not shown.

- a) Geometry of initial datafile.
- b) Geometry after refinement of elements.
- c) After 50 iterations of surface energy minimization.
- d) Close-up of edge interactions after 120 iterations. The concave meniscus can be seen.

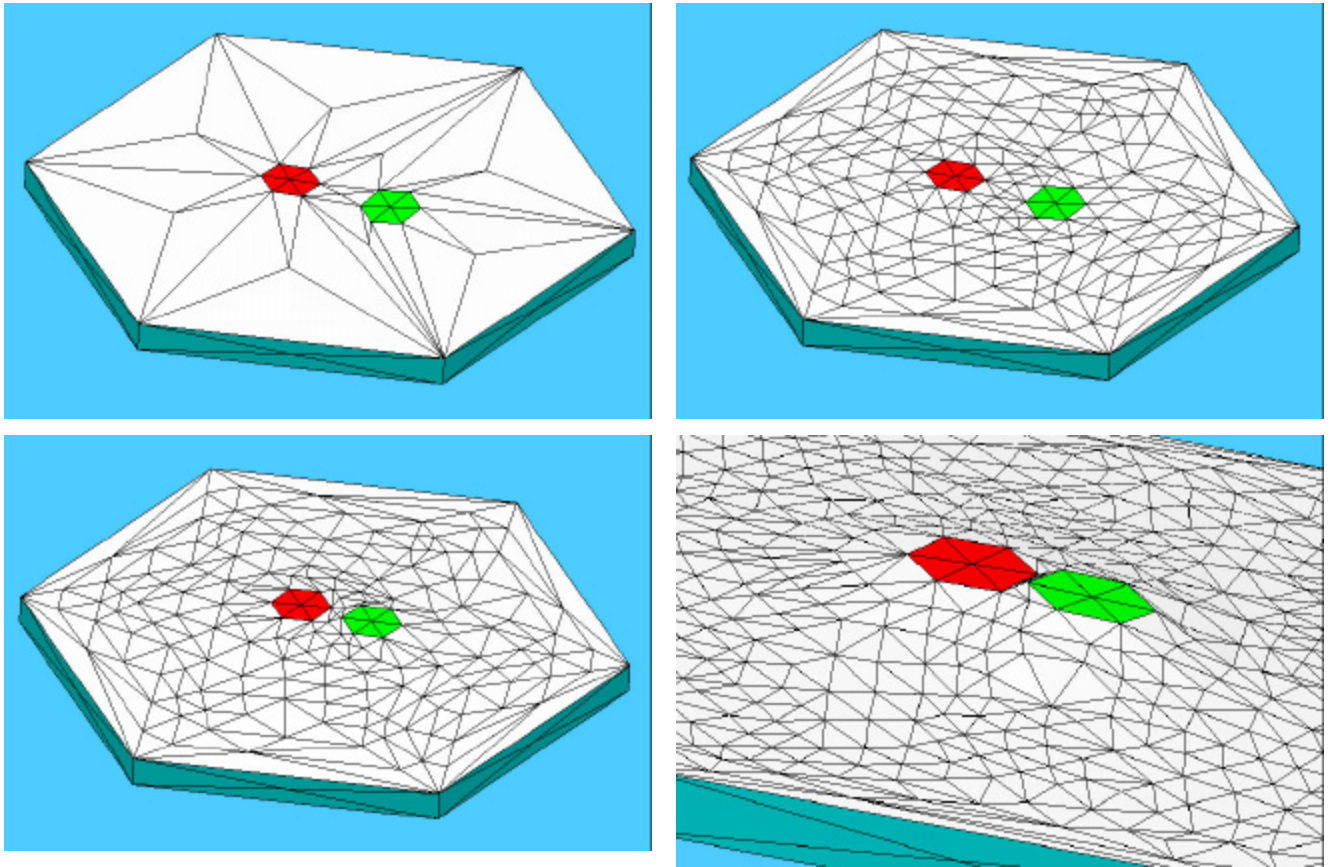


Figure 7: Example 2: Evolution of minimum energy surface for 2 hexagons floating on an hexagonal petri dish with mass acting at their center of gravity, with wetting sides and bottom surface (in contact with fluid) with a contact angle of 65 degrees. Hexagons are given an initial orientation, position, and tilt angle relative to the absolute so-ordinate system.

- a) Geometry of initial datafile.
- b) Geometry after refinement of elements.
- c) After 50 iterations of surface energy minimization.
- d) Close-up of edge interactions after 120 iterations. In this case the convex meniscus can be seen.

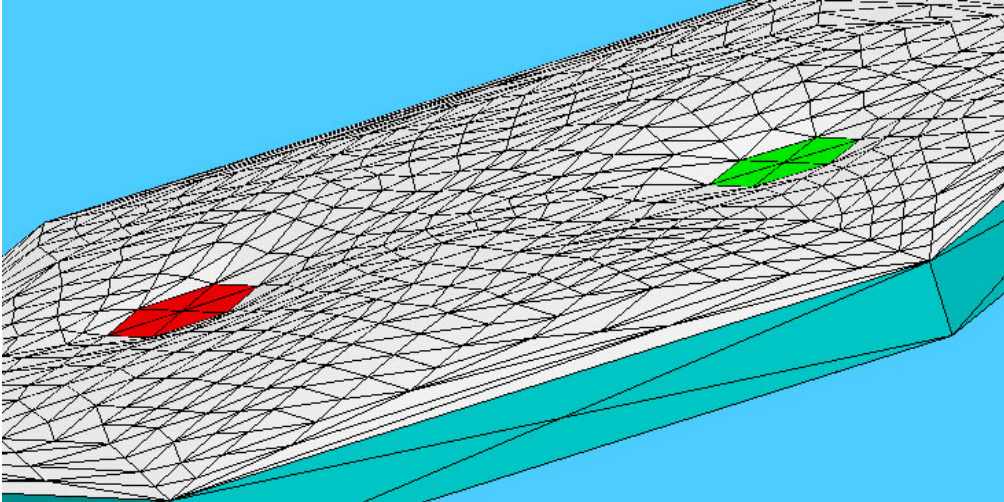


Figure 8. *Non converging hexagons at an interface due to large separation distance.*

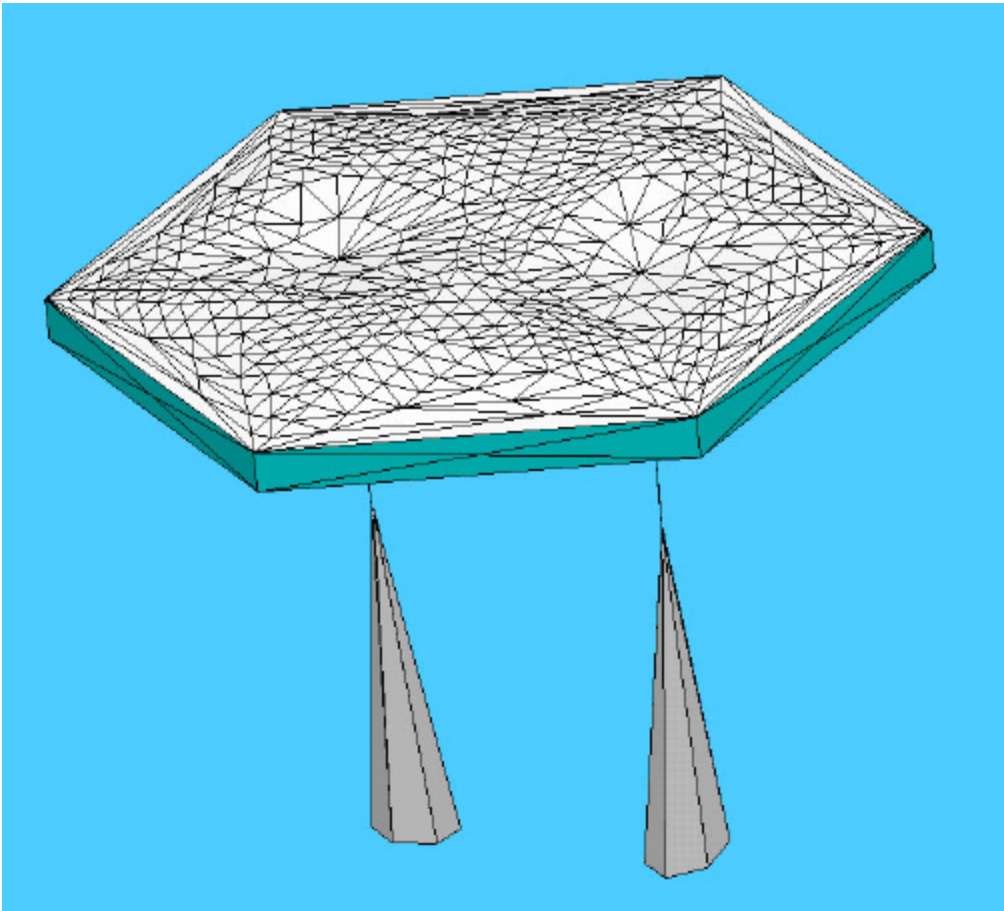


Figure 9. *When the mass of the floating objects is too high, or they become too large, they sink into the bath. The model does not recalculate the reformation of a clean surface once the parts have fallen through.*

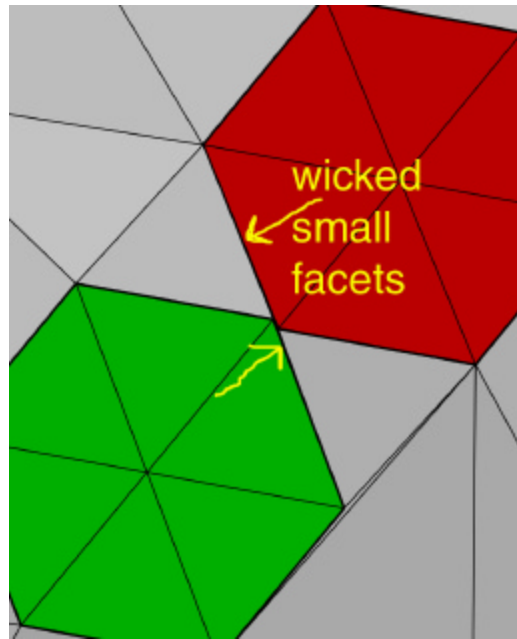


Figure 9: Faces do not quite meet as small facets remain and are squeezed between edges. This effectively pins the surfaces short of their real equilibrium position.

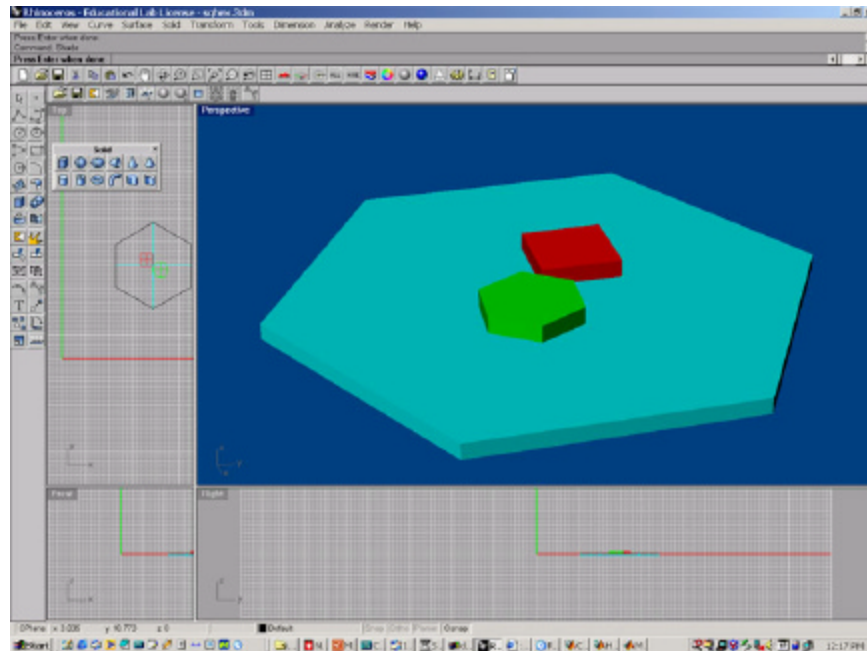


Figure 10: Geometry import direct from Rhino (3D CAD) via raw triangles into Surface Evolver to facilitate fast iteration in design time.

Relevant and useful reading.

I have included a long list of papers that I have found useful as a general introduction to the field of self assembly.

1. [001] Three-Dimensional Self-Assembly of Complex, Millimeter-Scale Structures through Capillary Bonding. Oliver, S.R.J., Clark, T.D., Bowden, N., Whitesides, G.M., *J.Am.Chem.Soc.* 2001, 123, 8119-8120.
2. [002] Mesoscale Self-Assembly of Hexagonal Plates Using Lateral Capillary Forces: Synthesis Using the "Capillary Bond". Bowden, N., Choi, I.S., Grzybowski, B.A., Whitesides, G.M., *J.Am.Chem.Soc.*, 1999, 121, 5373-5391
3. [003] Design of Three Dimensional, Millimeter-Scale Models for Molecular Folding. Clark, T.D., Boncheva, M., German, J.M., Weck, M., Whitesides, G.M., *J.Am.Chem.Soc.*, Vol.124, No.1., 2002.,18-19.
4. [004] Molecule-Mimetic Chemistry and Mesoscale Self-Assembly. Bowden, N.B., Weck, M., Choi, I.S., Whitesides, G.M., *Acc.Chem.Res.*, 2001, 34, 231-238.
5. [005] Fabrication of Micrometer-Scale, Patterned Polyhedra by Self-Assembly. Gracias, D.H., Kavthekar, V., Love, J.C., Paul, K.E., Whitesides, G.M., *Adv.Mater.* 2002, 14, No.3, Feb 5, 235-238
6. [006] Fabrication of Three-Dimensional Microstructures by Electrochemically Welding Structures Formed by Microcontact Printing on Planar and Curved Substrates., Jackman, R.J., Brittain, S.T., Whitesides, G.M., *J.MEMS*, Vol.7, No.2, June 1998. pp261.
7. [007] Microorigami: Fabrication of Small, Three-Dimensional, Metallic Structures. Brittain, S.T., Schueller, O.J.A., Wu, H., Whitesides, S., Whitesides, G.M., *J.Phys.Chem.B* 2001, 105, 347-350.
8. [008] Self-Assembly of an Operating Electrical Circuit Based on Shape Complimentarity and the Hydrophobic Effect. Terfort, A., Whitesides, G.M., *Adv. Mater.* 1998, 10, No.6.
9. [009] Crystallization of Millimeter-Scale Objects with Use of Capillary Forces. Tien, J., Breen, T.L., Whitesides, G.M., *J.Am.Chem.Soc.* 1998, 120, 12670-12671
10. [010] Assembly of Mesoscopic Analogues of Nucleic Acids. Weck, M., Choi, I.S., Jeon, N.L., Whitesides, G.M., *J.Am.Chem.Soc.* 2000, 122, 3546-3547
11. [011] Microfabrication through Electrostatic Self-Assembly. Tien, J., Terfort, A., Whitesides, G.M., *Langmuir* 1997, 13, 5349-5355
12. [012] Mesoscopic, Templated Self-Assembly at the Fluid-Fluid Interface, Choi, I.S., Weck, M., Xu, B., Jeon, N.L., Whitesides, G.M., *Langmuir* 2000, 16, 2997-2999
13. [013] Self-Assembly of Microscale Objects at a Liquid/Liquid Interface through Lateral Capillary Forces. Bowden, N., Arias, F., Deng, T., Whitesides, G.M., *Langmuir* 2001, 17, 1757-1765
14. [014] Self-Assembly of 10-micron-Sized Objects into Ordered Three Dimensional Arrays. Clark, T.D., Tien, J., Duffy, D.C., Paul, K.E., Whitesides, G.M., *J.Am.Chem.Soc.* 2001, 123, 7677-7682
15. [015] Design and Self-Assembly of Open, Regular, 3D Mesostructures, Breen, T.L., Tien, J., Oliver, S.R.J., Hadzic, T., Whitesides, G.M., *Science*, 7 May, 1999, Vol. 284, pp948

16. [016] Forming Electrical Networks in Three Dimensions by Self-Assembly, Gracias, D.H., Tien, J., Breen, T.L., Hsu, C., Whitesides, G.M. Science, 18 August 2000, Vol. 289, 1170-1172
17. The Chemical Basis of Morphogenesis, A.M.Turing, Phil.Trans.of the Royal Society of London, Series B, Biological Sciences, Vol.237, Issue 641 (Aug.14, 1952), 37-72.
18. Morphogenesis. Bard, J., Cambridge University Press, U.K., 1990.
19. Biomathematics: Mathematics of Biostructures and Biodynamics, Andersson et.al., Elsevier, 1999.
20. Cellular Automata Machines, Toffoli and Margolus, MIT Press, 1988.
21. Toward in vivo Digital Circuits Authors: Ron Weiss, George Homsy, Thomas F. Knight Presented: Dimacs Workshop on Evolution as Computation, Princeton, NJ, January 1999.
22. Wolfram, Stephen., "A New Kind of Science"., Wolfram Media. 2002.
23. Banks, Edwin. R., "Information processing and transmission in cellular automata" , PhD Thesis, Dept. Mechanical Engineering, MIT, 1971.
24. Theory of Self-Reproducing Automata. Von Neumann. University of Illinois Press, 1966
25. Jacobson, Homer. "On models of reproduction" 1958 American Scientist 46 255-284
26. Penrose L S, Penrose R "A self-reproducing analogue" 1957 Nature 179 1183
27. Penrose, L.S. "Self Reproducing Machines" Scientific American, vol.200, June 1959- pp 105-114
28. Microactuation by Continuous Electrowetting and Electrowetting: Theory, Fabrication, and Demonstration. Jung-Hoon Lee, PhD. Thesis. Mechanical Engineering, UCLA, 2000.