Approaching P=NP: Do Soap Films Solve the Steiner Tree Problem?

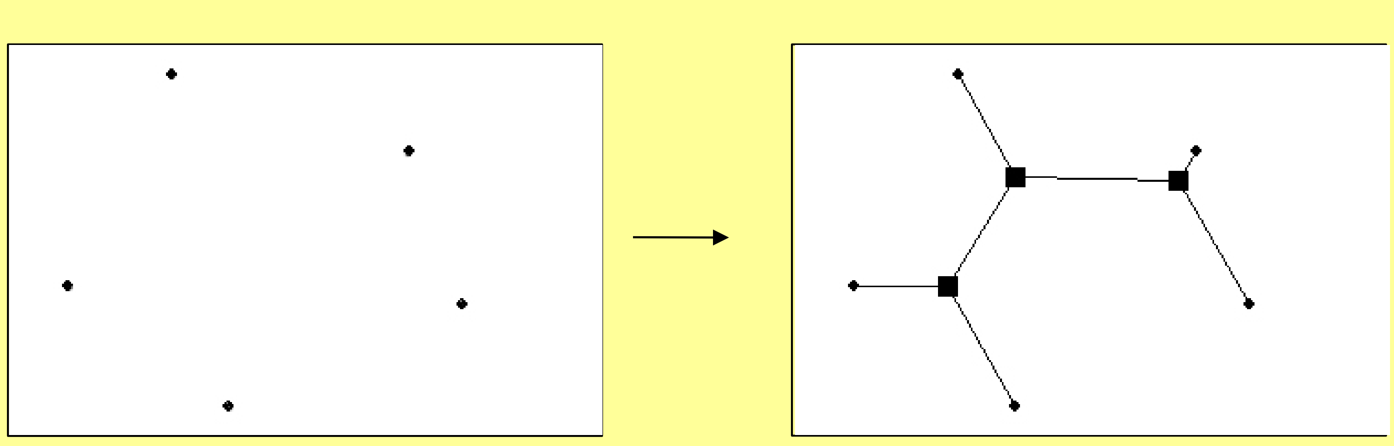
Long Ouyang
TJHSST Computer Systems Lab 2005-2006

ABSTRACT

Computation theory folklore has it that soap films calculate Steiner trees. If true, this *might* imply P=NP. However, no rigorous physical experimentation has been performed on the matter. Indeed, such experimentation would be difficult to establish standards for. Research in previous fields suggests that using physical systems as analog computers is entirely possible. However, critics state that while physical systems can relax to local minima, there is no guarantee that they will relax to global minima. To determine the plausibility of the method in question, this project will construct a computational model of soap films, to examine, with reproducible results, the action of soap films in forming Steiner trees.

BACKGROUND

In computer science, it is often convenient to formulate questions as decision problems. A decision problem is a yes/no question. Concerning decision problems, there are two complexity classes that have generated widespread interest. Simply put, P contains problems whose answers can be determined in polynomial time and NP contains problems whose answers can be verified in polynomial time using a deterministic Turing machine, which is what modern programming languages produce. A great question in complexity theory asks, is P=NP? In other words, are problems we previously thought were rather difficult actually relatively easy? The Steiner Tree Problem (STP) is in NP and has been widely studied. The problem basically consists of finding the minimum weight tree given that one can add additional vertices:

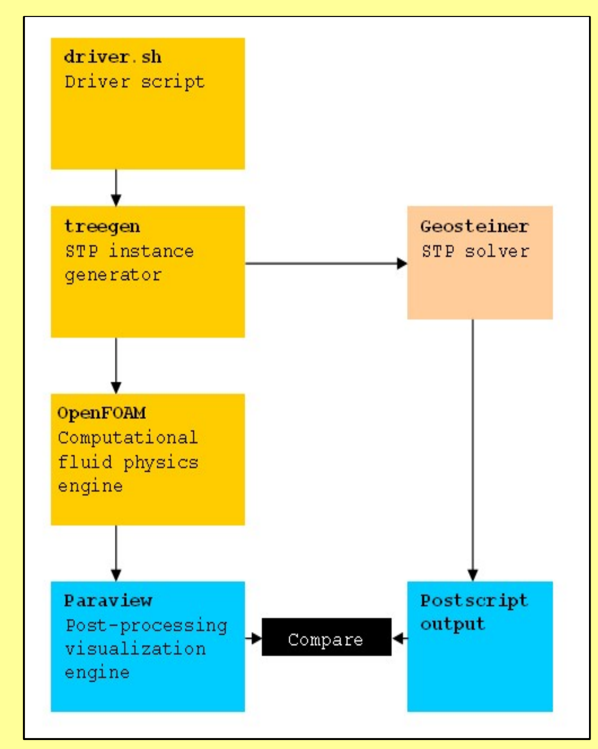


It's thought that soap films, which act to minimize surface area, satisfy this requirement.

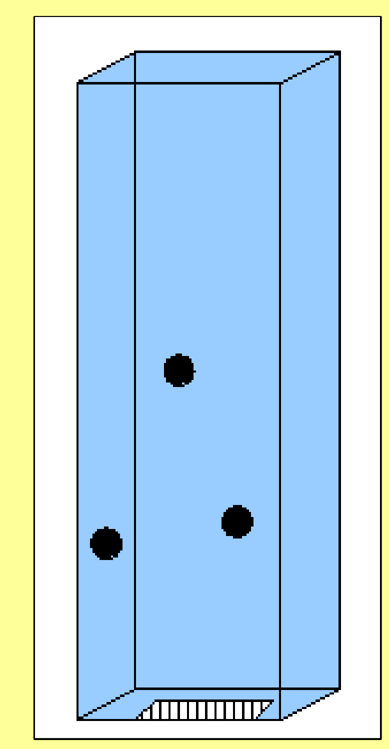
PROCEDURE

The idea was to compare solutions of known exact algorithms to the solutions of soap films, as modeled in a CFD (Computation Fluid Dynamics) simulation.

The software flow:



The simulation model:



The project tied together existing software, custom software inputs, and newly created programs.

Treegen, an in-house solution, generated a problem instance, which was a computational mesh and initial field dictionary for OpenFOAM (the CFD engine) and a ASCII file of coordinates for Geosteiner.

The OpenFOAM simulation was that of a glass box filled with soap solution. Vertices were represented as pegs spanning the walls, and a drain was present at the bottom of the box.

Geosteiner Postscript outputs were compared to final states of Paraview post-processing visualizations.

RESULTS

Difficulties were encountered with the OpenFOAM computational mesh framework. The Newtonian fluid physics model was correctly set up, but the meshing utilities had bugs, which were fixed in a new version of OpenFOAM which unfortunately was released in the middle of this development.

CONCLUSION

The calculating mesh and physics simulation portions of the OpenFOAM module could not be made to operate together. Future research will focus on correct mesh operation and code optimization.