

Bubbles and Chips

We'll use the MATLAB PDETOOLBOX to solve two partial differential equations, one modeling the shape of soap bubbles and one modeling the dissipation of heat.

Soap Bubbles.

In our first problem we solve the **minimal surface equation**. Given a wire frame, we look for the function with minimal surface area that touches the wire everywhere. This is sometimes called **Plateau's problem**. It is a model of the shape that a soapy film would take when attached to a wire frame.

We need to specify the wire frame mathematically. Suppose that we suspend it over the $x - y$ plane, and let its projection onto the plane (i.e, its shadow when the light is directly overhead) be a curve Γ . We define a function $v(x, y)$, for $(x, y) \in \Gamma$, that gives the height of the wire above the $x - y$ plane. The PDE is

$$\nabla \cdot \left(\frac{1}{\sqrt{1 + u_x^2 + u_y^2}} \nabla u \right) = 0.$$

The boundary conditions are that $u(x, y) = v(x, y)$ for $(x, y) \in \Gamma$.

We will solve a minimal surface equation using MATLAB's graphical interface to its PDETOOLBOX.

CHALLENGE 1 *Let's solve the problem over the circle with radius 0.5, centered at the origin, and with $u(x, y) = v(x, y) = \cos(2\pi x) \cos(\pi|y + .25|)$ when (x, y) is on the boundary of the circle.*

- In MATLAB, type `pdecirc(0,0,.5)`
- Using the "Mesh" menu or the triangle buttons, initialize the mesh and then refine the mesh to get 1016 triangles.
- The coefficients for the problem are set in "PDE specification" under the "PDE" menu (or by using the "PDE" button), using the notation

$$\nabla \cdot (\mathbf{c} \nabla u) + au = f,$$

- The boundary condition is set in the "Boundary" menu.
- Use "Solve \rightarrow parameters" to ask for the "nonlinear solver", "full" Jacobian, with a tolerance of $1.e-3$.
- Use the "Plot \rightarrow parameters" to choose the "height" option.
- "Solve", and print a copy of the plot.

POINTER 1 Helpful(?) Hints

When using the graphical interface:

- If you are working remotely, be prepared for frequent crashes, usually due to the terminal and X-server doing bad things.
- “export” makes variables appear in your MATLAB command window.
- Don’t put spaces in your expressions for the pde coefficients.
- You can enter functions that depend on \mathbf{x} , \mathbf{y} , \mathbf{ux} (i.e., u_x), etc.
- The PDETOOLBOX considers \mathbf{x} , \mathbf{y} , etc. to be vectors, so use the “.” operators as appropriate in your expressions for the pde coefficients.
- If you need a demo of the PDETOOLBOX, stop by during office hours and I’ll walk you through it.

-
- Click on “Mesh → Export mesh” and “Solve → Export solution” to make information about the mesh and the solution available in your MATLAB command window. (In particular, you should see the variables \mathbf{p} , \mathbf{e} , \mathbf{t} , \mathbf{u} .)
 - Use `tri2grid` to find the value of the solution at $x = -0.4$, $y = 0.1$. Write the value of $u(-0.4, 0.1)$ on your plot (by hand is ok) and submit it as your solution to this challenge.
-

Heat Dissipation¹

Computer chips work faster if they are not too hot, but operating them creates heat. In order to dissipate the heat, designers can surround the (silicon) processing units by a high-conductivity material like copper to act as a heat sink to conduct the heat away. The cross-section of a typical configuration is shown in the figure.

The temperature obeys the equation

$$\frac{\partial T(\mathbf{x}, t)}{\partial t} - \nabla \cdot (k(\mathbf{x}) \nabla T(\mathbf{x}, t)) = Q(\mathbf{x}),$$

where k is the conductivity of the material and Q models the heat source. The (generalized Neumann) boundary conditions are

$$\mathbf{n} \cdot (k(\mathbf{x}) \nabla T(\mathbf{x}, t)) = 0$$

¹Acknowledgement: This problem is inspired by one by Gunilla Kreiss, lab6.ps <http://www.nada.kth.se/kurser/kth/2D1250/tilnum2-04/index.html>

POINTER 2 Managing Heat and Power

Heat management in computer chips really is important. Lately, power management has also become an issue, so there is an effort to design software as well as hardware that reduces power consumption, which also lowers heat. On a large scale, this makes it cheaper and greener to operate data centers. On a small scale, it means that your cell phone battery will last longer between chargings. See, for example, an article by Eric Saxe in Communications of the ACM 53(2) 2010 doi:10.1145/1646353.1646370.

on the insulator boundary, and

$$\mathbf{n} \cdot (k(\mathbf{x}) \nabla T(\mathbf{x}, t)) = h(T(\mathbf{x}, t) - T_{room})$$

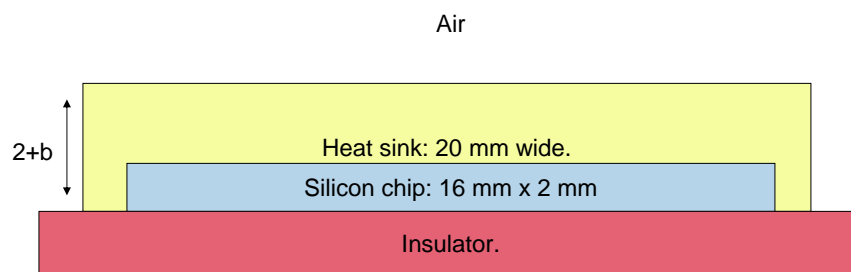
on the other boundaries, where \mathbf{n} is the outward normal direction (i.e., orthogonal to the boundary, length 1, and pointing out) h is the **heat transfer coefficient**, and T_{room} is room temperature. This says that there is no heat flow to the insulator, and that flow to the air depends on the difference in temperature. Here are the (somewhat realistic) values we will use:

Parameter	Value
$k(\mathbf{x})$ in silicon chip	0.148 W/(mm K)
$k(\mathbf{x})$ in copper	0.400 W/(mm K)
$Q(\mathbf{x})$ in silicon	5 W/(mm ³)
$Q(\mathbf{x})$ in heat sink	0
T_{room}	300 K
h , the heat transfer coef.	1.e-5 W/(mm ² K)

CHALLENGE 2 (a) *Initialize the mesh and then refine 4 times. Plot the temperature distribution in the domain at time $t = 100$ when the thickness $b = 2$ mm. (See figure.) Label the plot with the maximum temperature and identify the point at which it occurs.*

(b) *Plot the maximum temperature at $t = 100$ as a function of $b \in [1, 6]$ and determine (to 2 significant figures) the smallest thickness for which the temperature is at most 500K. How accurate do you believe your b value is? Justify your answer.*

Submit your plots, your program, your minimal b value, and your justification.



The domain of the PDE is the blue and yellow region.
The air and the insulator determine the boundary conditions.