The Minimal Surface Problem

The minimal surface problem Given $u(x, y) = g$ on the boundary of a domain Ω, find the surface u which has the minimal surface area.

Mathematical model Let $\Omega = (0, 1) \times (0, 1)$, find $u(x, y)$ satisfying

$$\begin{cases} \nabla \cdot \left(\frac{\nabla u}{\sqrt{1+u_x^2+u_y^2}} \right) \triangleq \frac{\partial}{\partial x} \left(\frac{u_x}{\sqrt{1+u_x^2+u_y^2}} \right) + \frac{\partial}{\partial y} \left(\frac{u_y}{\sqrt{1+u_x^2+u_y^2}} \right) = 0 & \text{in } \Omega, \\ u = g & \text{on } \partial\Omega. \end{cases}$$

Finite difference scheme We discretize the above model problem on an $(M+1)\times(M+1)$ uniform grid. The step size is $h = 1/M$ and $x_m = mh$, $y_l = lh$. Find the grid function $v_{ml} = v(x_m, y_l)$ satisfying the following scheme

$$\delta_x^+ \left(\frac{\delta_x^- v}{\sqrt{1 + (\delta_x^- v)^2 + (\delta_y^- v)^2}} \right) + \delta_y^+ \left(\frac{\delta_y^- v}{\sqrt{1 + (\delta_x^- v)^2 + (\delta_y^- v)^2}} \right) = 0, \tag{1}$$

where

$$\delta_x^+ v_{ml} = \frac{v_{m+1,l} - v_{m,l}}{h}, \qquad \delta_x^- v_{ml} = \frac{v_{m,l} - v_{m-1,l}}{h},$$

$$\delta_y^+ v_{ml} = \frac{v_{m,l+1} - v_{m,l}}{h}, \qquad \delta_y^- v_{ml} = \frac{v_{m,l} - v_{m,l-1}}{h}.$$

Let us simplify the above scheme as follows. Define

$$\begin{aligned} DS_{ml} &= \sqrt{1 + (\delta_x^- v_{ml})^2 + (\delta_y^- v_{ml})^2} \\ &= \sqrt{1 + \left(\frac{v_{m,l} - v_{m-1,l}}{h} \right)^2 + \left(\frac{v_{m,l} - v_{m,l-1}}{h} \right)^2}. \end{aligned}$$

Then (1) can be rewritten as

$$F_{ml}(\vec{v}) \triangleq \frac{1}{h} \left(\frac{v_{m+1,l} - v_{ml}}{h \cdot DS_{m+1,l}} - \frac{v_{m,l} - v_{m-1,l}}{h \cdot DS_{ml}} \right) + \frac{1}{h} \left(\frac{v_{m,l+1} - v_{ml}}{h \cdot DS_{m,l+1}} - \frac{v_{m,l} - v_{m,l-1}}{h \cdot DS_{ml}} \right) = 0 \tag{2}$$

Newton's method for solving nonlinear system of equations System (2) is nonlinear. We will use the Newton's method to solve it. In general, given a nonlinear system of equations

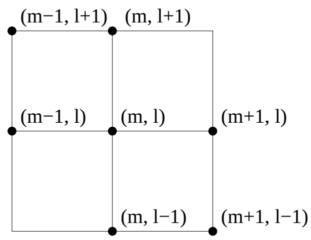
$$\mathbf{F}(\mathbf{V}) = 0$$

where $\mathbf{V}$ is an n-dimensional vector and $F_i(\mathbf{V})$, $i = 1, \ldots, n$ are n nonlinear equations. We consider the fol owing algorithm for the Newton's method:

1. set initial guess $\mathbf{V}_0$
2. for k from 0 to maximum iteration do:
   (a) compute $\mathbf{F}(\mathbf{V}_k)$ and the Jacobian $\frac{DF}{DV}(\mathbf{V}_k) = \left(\frac{\partial F_i}{\partial V_j}(\mathbf{V}_k) \right)_{1 \le i,j \le n}$
   (b) compute $\mathbf{V}_{k+1} = \mathbf{V}_k - 2^{-b} [\frac{DF}{DV}(\mathbf{V}_k)]^{-1} \mathbf{F}(\mathbf{V}_k)$, where $b \ge 0$ is the smallest integer which guarantees $\|\mathbf{F}(\mathbf{V}_{k+1})\| < (1 - \alpha 2^{-b}) \|\mathbf{F}(\mathbf{V}_k)\|$ for an $\alpha \in (0, 1)$ ($\alpha = 10^{-4}$ is typical)

(c) stop the iteration when $\|\mathbf{V}_{k+1} - \mathbf{V}_k\|$ is small enough, for example, less than 10^{-6}

Compute the Jacobian We need to compute the Jacobian for the scheme (2), whose entries are $\frac{\partial F_{ml}(\vec{v})}{\partial v_{ij}}$, for $1 \le m, l \le M+1$ and $1 \le i, j \le M+1$. Indeed, notice that $F_{ml}(\vec{v})$ only depends on 7 nodes (see the graph of stencil), it is not hard to see that



$$
\begin{aligned}
\frac{\partial F_{m,l}}{\partial v_{m,l-1}} &= \frac{1 + (\delta_x^- v_{m,l})^2 - \delta_x^- v_{m,l}\, \delta_y^- v_{m,l}}{h^2 (DS_{m,l})^3} \\
\frac{\partial F_{m,l}}{\partial v_{m+1,l-1}} &= \frac{\delta_x^- v_{m+1,l}\, \delta_y^- v_{m+1,l}}{h^2 (DS_{m+1,l})^3} \\
\frac{\partial F_{m,l}}{\partial v_{m-1,l}} &= \frac{1 + (\delta_y^- v_{m,l})^2 - \delta_x^- v_{m,l}\, \delta_y^- v_{m,l}}{h^2 (DS_{m,l})^3} \\
\frac{\partial F_{m,l}}{\partial v_{m,l}} &= -\frac{1 + (\delta_y^- v_{m+1,l})^2}{h^2 (DS_{m+1,l})^3} - \frac{1 + (\delta_x^- v_{m,l+1})^2}{h^2 (DS_{m,l+1})^3} \\
&\quad - \frac{2 + (\delta_x^- v_{m,l})^2 + (\delta_y^- v_{m,l})^2 - 2\,\delta_x^- v_{m,l}\, \delta_y^- v_{m,l}}{h^2 (DS_{m,l})^3} \\
&= -\left(\frac{\partial F_{m,l}}{\partial v_{m,l-1}} + \frac{\partial F_{m,l}}{\partial v_{m+1,l-1}} + \frac{\partial F_{m,l}}{\partial v_{m-1,l}} + \frac{\partial F_{m,l}}{\partial v_{m+1,l}} + \frac{\partial F_{m,l}}{\partial v_{m-1,l+1}} + \frac{\partial F_{m,l}}{\partial v_{m,l+1}} \right) \\
\frac{\partial F_{m,l}}{\partial v_{m+1,l}} &= \frac{1 + (\delta_y^- v_{m+1,l})^2 - \delta_x^- v_{m+1,l}\, \delta_y^- v_{m+1,l}}{h^2 (DS_{m+1,l})^3} \\
\frac{\partial F_{m,l}}{\partial v_{m-1,l+1}} &= \frac{\delta_x^- v_{m,l+1}\, \delta_y^- v_{m,l+1}}{h^2 (DS_{m,l+1})^3} \\
\frac{\partial F_{m,l}}{\partial v_{m,l+1}} &= \frac{1 + (\delta_x^- v_{m,l+1})^2 - \delta_x^- v_{m,l+1}\, \delta_y^- v_{m,l+1}}{h^2 (DS_{m,l+1})^3}
\end{aligned}
$$

Boundary condition On the boundary, we set

$$F_{ml}(\vec{v}) = g_{ml} - v_{ml} = 0 \qquad \text{on } \partial\Omega.$$

The corresponding entry in the Jacobian is

$$\frac{\partial F_{m,l}}{\partial v_{m,l}} = -1 \qquad \text{on } \partial\Omega.$$

Matlab code Combining all the above, the Matlab code is given in `minimalsurface.m` file.