Day 5: Flatland Challenge

Quantitative Engineering Analysis

Spring 2019

1 Schedule

- 0900-0915: Quiz
- 0915-0930: Debrief
- 0930-1030: Gradient Ascent in MATLAB
- 1030-1045: Coffee
- 1045-1215: Flatland

2 Quiz [15 min]

You are standing on the side of a hill trying to decide which way to walk in order to reach the top. Taking due north as the positive y direction (0 degrees) and due east as the positive x direction (90 degrees) , the height h of a point on the hillside can be described by the expression

$$h = 500 - 0.01x^2 - 0.02y^2,$$

where x, y and h are given in feet. Your location is $(20, 20, 488)$.

1. If you start walking due west, will you be heading directly towards the top of the hill?
2. Will your elevation change faster if you head west or southwest?
3. What will your rate of ascent be if you start walking in the direction of steepest ascent?
4. In which direction should you start walking in order to reach the top of the hill quickest?
5. If you change your mind and decide you want to stroll without exertion, remaining at an elevation of 488 feet, in which direction should you head initially?

3 Debrief [15 min]

Discuss the overnight with your table-mates, and make a list of concepts you feel solid on, and concepts you feel shaky on. Make a list of plus and deltas for this assignment. This debrief is short, but you will be applying the concepts from the overnight during the next exercise.

4 Gradient Ascent in MATLAB [60 min]

In the overnight you started thinking through the computational process of gradient ascent. In this exercise we would like you to work with a partner to implement gradient ascent in MATLAB. Recall that if $f(\mathbf{r})$ is a scalar function of a position vector $\mathbf{r}$, the points determined by gradient ascent are given by

$$\mathbf{r}_{i+1} = \mathbf{r}_i + \lambda_i \nabla f(\mathbf{r}_i),\ i = 1, \ldots$$

where λ_i is the relative size of the step that we take in the direction of the gradient. There are various schemes for choosing these, and in the overnight you considered gradient ascent with a simple proportionality

$$\lambda_{i+1} = \delta \lambda_i$$

where both δ and λ_0 are thoughtfully chosen for the problem at hand.

1. Work with your partner on pseudo-code for the algorithm. Keep it general: f is a general scalar function of a vector $\mathbf{r}$. You are going to want to use a loop - a "while" loop would be a good choice - what would be a reasonable stopping criterion?
2. When you are happy with your pseudo-code, develop a script or function that:
   (a) Automatically determines the discrete points $\mathbf{r}_1, \mathbf{r}_2, \ldots$ given an initial point $\mathbf{r}_0$.
   (b) Can be tuned by varying δ and λ_0.

   You should implement your method on the function that we met in the overnight

$$f(x, y) = xy - x^2 - y^2 - 2x - 2y + 4$$

and in order to validate your approach you will want to visualize the contours and the discrete points.

5 Coffee [15 min]

6 Flatland [90 min]

During this session, you will develop a method and implementation to drive your NEATO on the floor of the classroom in a way that physically realizes the method of Steepest Ascent. The mountain you will "climb" is defined as follows: (this should look somewhat familiar—when have you seen this equation before and what does it describe?)

$$z = f(x, y) = xy - x^2 - y^2 - 2x - 2y + 4$$

where x, y, and z are measured in feet. You are required to start your NEATO at $(1, -1)$ pointing in the $+y$ direction, and work your way to the top.

3. Work with your partner to decompose this problem. What are the steps involved? Get some feedback about your decomposition from your table-mates and instructors. You should have a decomposition that clearly explains the process of driving the NEATO along a discrete approximation to the path of steepest ascent.

4. Now develop the code for the NEATO. Get some feedback before you start driving a NEATO.

5. Now drive your NEATO!

7 *Optional Flatland Extension*

If we want to follow the continuous path of steepest ascent, we should set the velocity of the NEATO proportional to the gradient, $\mathbf{r}'(t) = \alpha \nabla f$, where α is a parameter that we can choose depending on how fast we want the NEATO to drive.

6. How does the speed of the NEATO depend on α and ∇f? If α is a constant, how fast is the NEATO moving when it reaches the "top" of the hill? How would we choose α if we wanted the NEATO to move with constant forward velocity v?

7. In terms of coordinates x and y this approach is equivalent to defining the following set of ordinary differential equations, of the type that you encountered in ModSim

$$\begin{aligned} \frac{dx}{dt} &= \alpha \frac{\partial f}{\partial x} \\ \frac{dy}{dt} &= \alpha \frac{\partial f}{\partial y} \end{aligned}$$

   where $x(0)$ and $y(0)$ represent the initial position of the NEATO. You can either solve these in MATHEMATICA using DSolve, in MATLAB using dsolve, or by hand if you are very familiar with solving differential equations.

8. Implement this and drive your NEATO in a continuous path uphill!