Project on the steepest ascent method for calculating maximas

In the following we shall consider only functions of two variables, but all this can be extended to functions of several variables. It was explained in the lectures that any function $f(x, y)$ defined and continuous on a closed set S attains its maximum as well as its minimum. You maybe remember that the proof of this theorem was somewhat tricky and did not tell us how one might go about finding the maximum and the minimum. The goal of this project is to write a program that helps you to calculate just that for functions that are nice functions, i.e., as many times differentiable as we like.

We shall concentrate on calculating maxima, the calculation of minima requires simple modifications. The basic idea is quite simple to understand. We assume that the function $f(x, y)$ is differentiable and hence has a gradient

$$\nabla f(x, y) \ ,$$

which points always in the direction of largest increase of the function $f(x, y)$. Thus, if we follow the gradient we will always walk up the mountain and eventually it something which looks like the maximum. We say 'looks like' since several things may happen.

- By walking along the gradient you may hit the boundary of S, or
- you might hit an interior point where, necessarily, the gradient of f vanishes.

None of these points guarantee that you have reached the maximum of the function. But certainly with the second possibility you have found a point with the property that in a sufficiently small neighborhood of it the function cannot be larger, in other words it is a local maximum.

From the above observation we deduce the following procedure.

- Start with a point $\mathbf{x_0}$. Pick a small number h.
- For $k = 1, 2, 3 \ldots$ calculate the points

$$\mathbf{x_k} = \mathbf{x_{k-1}} + h\nabla f(\mathbf{x_{k-1}}) \ ,$$

and values

$$f(\mathbf{x_k}) \ .$$

As always the important point is to devise a stopping rule and this depends on the accuracy one desires. Let us ignore the problem that our procedure might hit the boundary. Clearly, the goal is to find the points where the gradient vanishes. Hence we will run our procedure until $|\nabla f(\mathbf{x_k})| < \varepsilon$ where ε is a given accuracy.

At that point we would like to think that we are close to a local maximum. One could be close to a saddle point which would be not very good but this is very unlikely (Why?). (To check for this we calculate the Hessian at this point. If the Hessian has a negative determinant we know that we are near a saddle point and we stop and start all over again with another point. Otherwise the determinant must be positive and we must be somehow close to a local maximum) How close are we to a maximum? To decide that we write the second order Taylor expansion

$$f(\mathbf{x_M}) \approx f(\mathbf{x_k}) + \nabla f(\mathbf{x_k}) \cdot (\mathbf{x_M} - \mathbf{x_k}) + \frac{1}{2}(\mathbf{x_M} - \mathbf{x_k}) \cdot H_f(\mathbf{x_k})(\mathbf{x_M} - \mathbf{x_k}) \ . \qquad (1)$$

It is a good idea to recall that at a local maximum the Hessian has two negative eigenvalues $-\mu_1 \le -\mu_2$. Since we expect that $\mathbf{x_k}$ is close to a maximum we suspect that the two eigenvalues of $H_f(\mathbf{x_k})$ are also negative. Since $f(\mathbf{x_M}) \ge f(\mathbf{x_k})$ we have that

$$0 \le \nabla f(\mathbf{x_k}) \cdot (\mathbf{x_M} - \mathbf{x_k}) + \frac{1}{2}(\mathbf{x_M} - \mathbf{x_k}) \cdot H_f(\mathbf{x_k})(\mathbf{x_M} - \mathbf{x_k})$$

or

$$-\nabla f(\mathbf{x_k}) \cdot (\mathbf{x_M} - \mathbf{x_k}) \le \frac{1}{2}(\mathbf{x_M} - \mathbf{x_k}) \cdot H_f(\mathbf{x_k})(\mathbf{x_M} - \mathbf{x_k}) \ .$$

Since $\nabla f(\mathbf{x_k}) \cdot (\mathbf{x_M} - \mathbf{x_k}) \le |\nabla f(\mathbf{x_k})||(\mathbf{x_M} - \mathbf{x_k})|$ we learn that

$$-|\nabla f(\mathbf{x_k})||(\mathbf{x_M} - \mathbf{x_k})| \le \frac{1}{2}(\mathbf{x_M} - \mathbf{x_k}) \cdot H_f(\mathbf{x_k})(\mathbf{x_M} - \mathbf{x_k})$$

and since

$$(\mathbf{x_M} - \mathbf{x_k}) \cdot H_f(\mathbf{x_k})(\mathbf{x_M} - \mathbf{x_k}) \le -\mu_2 |(\mathbf{x_M} - \mathbf{x_k})|^2$$

we get that

$$-|\nabla f(\mathbf{x_k})||(\mathbf{x_M} - \mathbf{x_k})| \le -\frac{\mu_2}{2}|(\mathbf{x_M} - \mathbf{x_k})|^2$$

or

$$|\nabla f(\mathbf{x_k})||(\mathbf{x_M} - \mathbf{x_k})| \ge \frac{1}{2}\mu_2|(\mathbf{x_M} - \mathbf{x_k})|^2$$

This implies that

$$|(\mathbf{x_M} - \mathbf{x_k})| \le \frac{2|\nabla f(\mathbf{x_k})|}{\mu_2} \ ,$$

and hence

$$|(\mathbf{x_M} - \mathbf{x_k})| \le \frac{2\varepsilon}{\mu_2} \ .$$

Thus, the distance between the point we are looking for and our approximate point is again of the order of ε. The value of the function is actually much closer as which can be gleaned from formula (1). The second and third term on the right side are of order ε^2 and hence the difference of $f(\mathbf{x_M})$ and $f(\mathbf{x_k})$ is of the order ε^2.

At this moment, however, we can do better. Since we are interested in $\mathbf{x_M}$, we can use $\mathbf{x_k}$ as an approximate point for solving the equation

$$\nabla f(\mathbf{x}) = 0$$

whose solution is, of course, the point $\mathbf{x_M}$. Thus, running Newton's method one step amounts to solving the equation

$$\mathbf{x}^* = \mathbf{x_k} - J^{-1}_{\nabla f}(\mathbf{x_k})\nabla f(\mathbf{x_k}) = \mathbf{x_k} - H_f^{-1}(\mathbf{x_k})\nabla f(\mathbf{x_k}) \ .$$

we now expect that the point $\mathbf{x}^*$ is a good approximate solution. The size of $\nabla f(x^*)$ will now be of the order ε^2 and hence as above we get that $|\mathbf{x_M} - \mathbf{x}^*|$ is also of the order ε^2.

The algorithm (ignoring the boundary and saddles)

Pick ε (accuracy), the step size $h << \varepsilon$ and an initial point $\mathbf{x_0}$.

While

$$\sqrt{[f(x_k+h,y_k)-f(x_k,y_k)]^2+[f(x_k,y_k+h)-f(x_k,y_k)]^2} > \varepsilon h$$

compute

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} f(x_k+h,y_k)-f(x_k,y_k) \\ f(x_k,y_k+h)-f(x_k,y_k) \end{bmatrix} .$$

Else calculate

$$\begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} - H^{-1}(x_k,y_k) \begin{bmatrix} f(x_k+h,y_k)-f(x_k,y_k) \\ f(x_k,y_k+h)-f(x_k,y_k) \end{bmatrix}$$

where $H(x_k, y_k)$ is the Hessian of f, and calculate $f(x^*, y^*)$.

Questions

a) Explain the procedure in terms of what has been discussed before.

b) How do you calculate the Hessian numerically?

Problem

Consider the function

$$f(x,y) = \left(\frac{2}{1+x^2+y^2}\right)^2$$

and test your algorithm. Choose $\varepsilon = 10^{-2}$ pick the initial point $(1,0)$ and run your algorithm with the choices $h = 10^{-3}, 10^{-4}$ and 10^{-5}. Depending on these choices your algorithm will stop at different values $\mathbf{x_k}$.

a) Count the number of iterations until the algorithm stops.

b) Determine the distance of $\mathbf{x_k}$ and the point where the maximum is attained.

c) Calculate the difference between the the maximum and the value $f(\mathbf{x_k})$. Are these numbers reasonable?

d) Calculate for each of the various values for h, $\mathbf{x}^*$ and $f(\mathbf{x}^*)$. How far away from the point where the maximum is attained is $\mathbf{x}^*$? How accurate is $f(\mathbf{x}^*)$? Are these numbers reasonable?

e) Having tested your program consider the function

$$f(x,y) = \left(\frac{2}{1+x^2+y^2}\right)^2 + \frac{1}{1+(x-4)^2+(y-3)^2} .$$

Use your program to calculate the maximum of this function. How many local maxima are there.

f) Calculate all of them.

Extra credit

Use the above algorithm in conjunction with the first project to construct a neighborhood in which the maximum must be. Proceed as follows. Run the above algorithm until $|\nabla f(\mathbf{x_k})| < \varepsilon$. Use $\mathbf{x_k}$ as a starting point and use the program in project one to draw the level curve that encircles the local maximum.