Some more questions/problems you should be able to work out.

This sheet is about minimization algorithms.

— — — — — — — — — — — — — — ⋆ — — — — — — — — — — — — —

1. (a) You are in possession of a method to calculate the function $G(x)$, but it is too expensive to plot. You are told that the function has a local minimum in the $x \in (a, c)$.

   Explain how you would use the method of golden section search to find the minimum.

   (b) Write the method as an *algorithm*, specifying what you compare at each step, how you calculate the new point at each step, and what renaming of variables you do at each step.

   (c) The golden section search uses the golden ratio $\phi = (1 + \sqrt{5})/2 \approx 1.618$, or its inverse, $\phi - 1 = \dfrac{1}{\phi} = \dfrac{-1+\sqrt{5}}{2} \approx 0.618$, as a ratio between the distances involved. Derive this number from the requirement of having the same ratio at different scales.

2. Explain the intuition behind simulated annealing, i.e., explain the algorithm and why it might help in obtaining the global minimum of a function of many variables.

3. Consider the matrix equation

$$A\mathbf{x} = \mathbf{b}$$

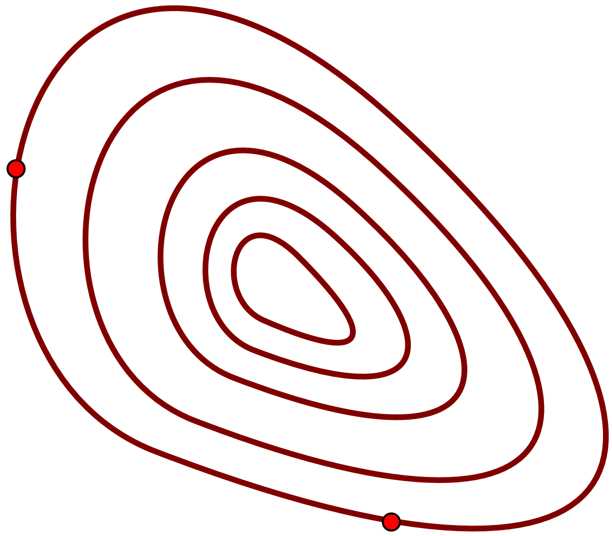
where A is a $n \times n$ square matrix, $\mathbf{x}$ is the $n \times 1$ column matrix (vector) of variables, and $\mathbf{b}$ is the column matrix of constants. Explain whether and/or why solving this matrix equation is the same as minimizing the function

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\mathrm{T}}A\mathbf{x} - \mathbf{x}^{\mathrm{T}}\mathbf{b}\,.$$

Here the superscript T stands for ‘transpose’.

Hints: Maybe first work out whether each term is a scalar, a row/column vector, or a matrix. Maybe working out your answer for 2 variables first is easier than the n-variable case.

4. Two-dimensional minimization. The figure shows level curves (contour lines) of the function $f(x_1, x_2)$ on the x_1-x_2 plane. Each curve corresponds to a constant value of $f(x_1, x_2)$, with the inner curves representing the smallest value.



   (a) Starting from one of the dots as the initial guess, show the trajectory that would be followed by the method of steepest descent, which minimizes successively in the direction of the negative gradient.

   Explain the angle between successive lines along this trajectory.

   (b) The 'gradient descent' algorithm also moves along the direction of the negative gradient, but does not minimize. i.e., it updates positions according to

$$x_i^{n+1} = x_i(n) - \eta \frac{\partial}{\partial x_i} f(x_1^{(n)}, x_2^{(n)})$$

   where $(x_1^{(n)}, x_2^{(n)})$ is the position after the n-th iteration. Here η is a small constant.

   i. Using the vector notation $\vec{r} \equiv (x_1, x_2)$, write the update rule as a vector equation.

   ii. Starting from one of the dots, show the trajectory that would be followed by this method. Explain the angle between successive lines along this trajectory.

   (c) Starting from one of the dots, show the trajectory that would be followed by the method of line minimization alternately along the two variable directions.

5. 'Programming' exercise. I will trust that you can write the programs if you can set up the iteration or the algorithm. So feel free to read each "Write a program for..." as "Set up the algorithm for...".

   Consider the function:

   $$f(x,y) \;=\; (x^2+2y-7)^2+(2x+y-5)^2$$

   Since the function is explicitly given, you can calculate the gradient, $\nabla f = (f_x.f_y)$, and the Hessian, $H = \begin{pmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{xx} \end{pmatrix}$.

   We will find the (x,y) coordinates for which this function is a minimum. Below, $\vec{r}$ refers to the 2D vector (x,y).

   (Of course, you could first cheat and find the positions of the minima analytically.)

   (a) Write a program that finds the minimum using the Nelder-Mead algorithm (a.k.a. downhill simplex algorithm or amoeba algorithm). Choose the vertices of the initial simplex (triangle) to be $(0,4)$, $(-1,3)$, $(-2,4)$.

   As the algorithm proceeds, record the center of mass of the triangle at each step, and plot these positions on an x-y plane to visualize the path taken by your amoeba.

   (b) Write a program that starts from the point $(-1,-2)$ and minimizes successively in the x and y directions while keeping the other coordinate fixed. For each line search you could use golden section search. Plot the points obtained by successive directional minimizations.

   (c) Write a program that uses the gradient descent algorithm, i.e., according to the update rule

   $$\vec{r}_{i+1} \;=\; \vec{r}_i - \eta\nabla f(\vec{r}_i)$$

   Plot the trajectory of this update rule starting from the point $(-1,-2)$. A safe value of η would be $\eta = 0.01$, but once your program is working, you could experiment with other values.

   How many iterations do you need to reach the minimum to within $(\Delta x, \Delta y) = (0.01, 0.01)$?

   (d) Set up the iteration rule for using Newton's algorithm to find the minimum of the function. If you need to invert a 2×2 matrix, you might remember that the inverse of

   $$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \qquad \text{is just} \qquad \frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix} = \frac{1}{\det(A)}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

(e) Do you expect Newton's algorithm to reach the minimum after a single iteration? Why, or why not?