MAA209 Numerical Optimization

École Polytechnique

Practical Session #2

Instructions: The codes should be written in Python. The use of notebooks is encouraged and you may start from the given examples. Collaboration and discussions are encouraged, if they are related to the subjects concerning the practical sessions.

Important: Upload your work on Moodle (one or multiple notebooks). Only Exercises 1 and 2 are mandatory. The other ones are supplementary.

Students which in addition to the main subjects solve some **supplementary** or **Challenge** questions will get additional bonus points. Recall however, that the **Challenge** and **supplementary** exercises are not mandatory.

Exercise 1 Curve fitting methods

1. Play with the code `Illustration_Newton.ipynb` available on Moodle and understand all its components. In particular:
   a) Observe how the algorithm behaves when you change the initialization.
   b) Add new functions of your choice in order to test the necessity of all conditions involved in the quadratic convergence result.
   c) Change the code in order to implement the False Position or Secant method. Verify the order of convergence and the dependence on the initialization.
2. **(Challenge)** We saw that basic line-search methods like the bisection method are very robust: they will always approximate the minimum of a unimodal function, but their rate of convergence is linear. Newton's method, on the other hand converges quadratically provided we have access to second derivatives, the minimizer is non-degenerate and we start close enough to the optimum. Modify the given code implementing Newton's method so that it will converge regardless of the starting point.
   - at each iteration consider two candidates for the new position: the one given by the bisection method (the midpoint of the current search interval) and the one given by Newton's method.
   - decide what is the new interval bracketing the minimum and pass to the next iteration

   a) Your algorithm should converge even in cases where Newton's method alone does not work.
   b) Observe the convergence rate: you should notice that in the beginning the convergence is linear due to the eventual use of bisection steps, while towards the end the rate of convergence becomes the same as for the Newton method.

Exercise 2 Gradient descent with line-search in 1D

1. Write a code implementing the gradient descent with fixed step in 1D. Apply it for various test functions and observe the convergence rate. Observe also the behavior of the algorithm with respect to the initial condition and the size of the step.
   If you are not sure how to start, just take the Goldstein-Price code and replace the line-search part by your choice of the fixed descent step t.
2. Implement the gradient descent with line-search based on Armijo's rule. You may start from the Notebook related to the Goldstein-Price line-search given on Moodle.
3. **(Challenge)** Implement the Wolfe line-search starting from the code given for the Goldstein-Price line-search.
4. Practical questions:
   - Test the behavior of the algorithm for various functions and for various choices of the parameters $m_1, m_2 \in (0, 1)$.
   - Check that for the function $f(x) = x^2$ choosing $m_1 > 0.5$ greatly increases the number of iterations because the optimal step for a quadratic function cannot be chosen.

Exercise 3 Find the closest point to a curve (Supplementary)

Suppose $\gamma : [0, 2\pi] \to \mathbb{R}^2$ is a closed curve in the plane and $A = (a_1, a_2)$ is a given point (you may denote $\gamma(\theta) = (x(\theta), y(\theta))$). The objective of this exercise is to write an algorithm which allows you to find the minimal distance AM_θ, where M_θ is the point corresponding to $\gamma(\theta)$.

1. (Optimality condition) Suppose that A is not a point on γ. Prove that if γ is of class C^1 and M_0 is the point which realizes the minimal distance AM_θ then AM_0 is a normal vector to the curve γ.
2. Note the minimization of AM_θ or AM_θ^2 gives the same minimizer. If $\gamma(\theta) = (x(\theta), y(\theta))$ and A has coordinates (a, b) give a formula for AM_θ^2 and for the derivative $\frac{d}{d\theta}(AM_\theta^2)$.
3. Implement a numerical algorithm which can search for the point realizing the minimal distance using one of the methods in the previous two exercises. Verify numerically that the minimizer verifies the optimality conditions.
   Note that you can use an algorithm implemented in previous exercises and just change the objective function.
   You may try the following cases:
   - the ellipse given by the parametrization $\gamma(\theta) = (2\cos\theta, \sin\theta)$ for various points A in the plane.
   - a curve given in radial coordinates by the following parametrization
   $$\gamma(\theta) = ((1 + 0.3\cos(3\theta))\cos\theta, (1 + 0.3\cos(3\theta))\sin\theta).$$
     and various points A in the plane.

Answer of exercise 3

1. The minimal distance is minimized at the same place where its square is minimized. The function to be minimized is, thus
   $$\theta \mapsto AM_\theta^2 = (x(\theta) - a)^2 + (y(\theta) - a)^2.$$
   At the minimum the derivative of this function is zero, which means that
   $$2(x(\theta) - a_1)x'(\theta) + 2(y(\theta) - a_2)y'(\theta) = 0.$$
   This can also be interpreted as the fact that the following scalar product is zero:
   $$\overrightarrow{AM_\theta} \cdot (x'(\theta), y'(\theta)) = 0.$$
   Since $(x'(\theta), y'(\theta))$ is a tangent vector to γ at θ it follows that AM_θ is a normal vector to the curve γ.
2. We already saw that the derivative of AM_θ^2 is
   $$2(x(\theta) - a_1)x'(\theta) + 2(y(\theta) - a_2)y'(\theta).$$
3. Use one of the gradient descent codes implemented in the previous exercise.

Exercise 4 Heron's algorithm (Supplementary)

Heron's algorithm computes an approximation of the square root of a strictly positive real number y using the iteration

$$x_0 > 0,\ x_{n+1} = \frac{1}{2}\left(x_n + \frac{y}{x_n}\right). \tag{1}$$

1. Show that the previous recurrence relation is Newton's algorithm for finding a zero of the function $f(x) = x^2 - y$. Following the results shown in the course show that (x_n) converges to $\sqrt{y}$ when x_0 is close enough to $\sqrt{y}$. Indicate the order of convergence of the sequence towards its limit.
   In the following, we consider the following error estimate:
   $$E_y(x) = \frac{x - \sqrt{y}}{x + \sqrt{y}}.$$
   This choice simplifies the computations and is an estimation of the half of the relative error when $x \to \sqrt{y}$.

2. Show that (x_n) verifies
$$E_y(x_{n+1}) = E_y(x_n)^2.$$
Deduce a formula for $E_y(x_n)$ as a function of $E_y(x_0)$. Show that the sequence x_n converges towards $\sqrt{y}$ for every initialization $x_0 > 0$.
The choice of the initialization x_0 determines $E_y(x_0)$ and has an influence on the speed of convergence. In the following we restrict ourselves to the case $y \in [1/2, 2]$ and we try to find the best initial condition. We will suppose that the initial condition x_0 depends on y and we will look at the two following cases: $x_0 = a$, a constant and $x_0 = a + by$, a polynomial of degree 1 in y. In each case we will minimize $\|g\|_{L^\infty([1/2,2])}$ with $g(y) = E_y(x_0)$.
3. Show that if $x_0 = a$ then $M_a = \max_{y\in[1/2,2]} |E_y(a)|$ is minimal for $a = 1$. Find the explicit value of M_a. (*Indication*: reresent graphically the function $y \mapsto |E_y(a)|$.)
4. Suppose that $x_0 = a + by$ is a polynomial of degree 1 in y. We want to find coefficients a and b which minimize $M_{a,b} = \|g\|_{L^\infty([1/2,2])}$ with $g(y) = E_y(a + by)$.
(a) Show that the maximum of $y \mapsto |g(y)|$ is reached for an element in the set $\{a/b, 1/2, 2\}$.
We admit that in order to minimize $M_{a,b}$ it is necessary that $a = b$ and that $g(2) = g(1/2) = -g(a/b)$.
(b) Conclude that the values of a and b which minimize $M_{a,b}$ are $a = b = \sqrt{\sqrt{2}/6}$. Find an exact expression for $M_{a,b}$.
5. We admit the following approximations $M_a \approx 0.17157$ and $M_{a,b} \approx 0.01472$. For $y = 2$ compare the number of iterations necessary to arrive at a demanded precision ($E_y(x_n) <$ tol) for the two initializations studied: $x_0 = 1$ and $x_0 = ay + b$ with $a = b = \sqrt{\sqrt{2}/6}$.
6. **(Implementation)** Implement Heron's algorithm and observe that it converges quadratically to $\sqrt{y}$. Test the two initialization and verify if the one of degree 1 reduces the number of iterations.

Answer of exercise 4

1. We have $f'(x) = 2x$ and the corresponding Newton algorithm is
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n^2 - y}{2x_n} = \frac{x_n}{2} + \frac{y}{2x_n}.$$
A simple computation shows that $f''(x) = 2$, therefore the positive zero of f is non degenerate. Therefore, if x_0 is close enough to $\sqrt{y}$ Newton's algorithm converges and the convergence is quadratic.
2. A direct computation shows that $E_y(x_n)^2 = E_y(x_{n+1})$. By induction we get $E_y(x_n) = (E_y(x_0))^{2^n}$. It is immediate to see that for $y > 0$ we have $|E_y(x_0)| < 1$. Therefore Heron's algorithm converges for every initial condition $x_0 > 0$.
3. Let $g(y) = E_y(a) = \frac{a-\sqrt{y}}{a+\sqrt{y}} = \frac{2a}{a+\sqrt{y}} - 1$. Then $g'(y) > 0$ and g is strictly increasing on $[1/2, 2]$. The maximum of $|g(y)|$ can be attained for $y \in \{1/2, 2\}$. We have
$$g(1/2) = \frac{a - \sqrt{1/2}}{a + \sqrt{1/2}} \text{ et } g(2) = \frac{a - \sqrt{2}}{a + \sqrt{2}}$$
which are strictly increasing functions in a. We can see, therefore, that the minimal value of $\max\{|g(1/2)|, |g(2)|\}$ is attained when $g(1/2) = -g(2)$. A quick computation shows that this implies that $a = 1$.
4. (a) It is enough to compute the derivative of $y \mapsto E_y(a + by)$ and to observe that this function is decreasing for $y \le a/b$ and increasing for $y \ge a/b$.
(b) Direct computation.
5. It is enough to see that $M_a^2 > M_{a,b} > M_a^4$. The initialization with a polynomial of degree 1 reduces the number of iterations by 1.