Chapter 1

Homework #2. Rakhmatulin Viktor

1.1 Task#1

Find minimum of function with respect to parameters a,b,c : $f(x) = ax^2 + bx + c$

Solution:

Minimum could lie at the point where $f'(x) = 0$, so $2ax + 1 = 0$ thus $x^* = -\frac{1}{2a}$

If $a > 0$: $f'(x) > 0$ when $x > x^*$ and $f'(x) < 0$ when $x > x^*$. Thus, x^* is the minimum point and $f(x^*) = \frac{a}{4a^2} - \frac{b}{2a} + c = \frac{1}{4a} - \frac{b}{2z} + c$

Answer: $a > 0, x^* = -\frac{1}{2a}, f(x^*)\frac{1}{4a} - \frac{b}{2z} + c$

1.2 Task#2

What are dimensions of gradient of a function $h(x) = f(Ax)$, constructed of function $f : R^m \rightarrow R$ and matrix $A \in R^{m \times k}$. Find the dimensions of $\nabla_x h(x)$

Solution:

$\nabla h(x) = A^T \nabla_y f(y)$ where $y = Ax$

Answer: $\nabla h_{k \times 1}(x)$

1.3 Task#3

Prove that for a strongly convex function with parameter μ holds,

$$\frac{\mu}{2}||x - x^*||^2 \leq f(x) - f(x^*)$$

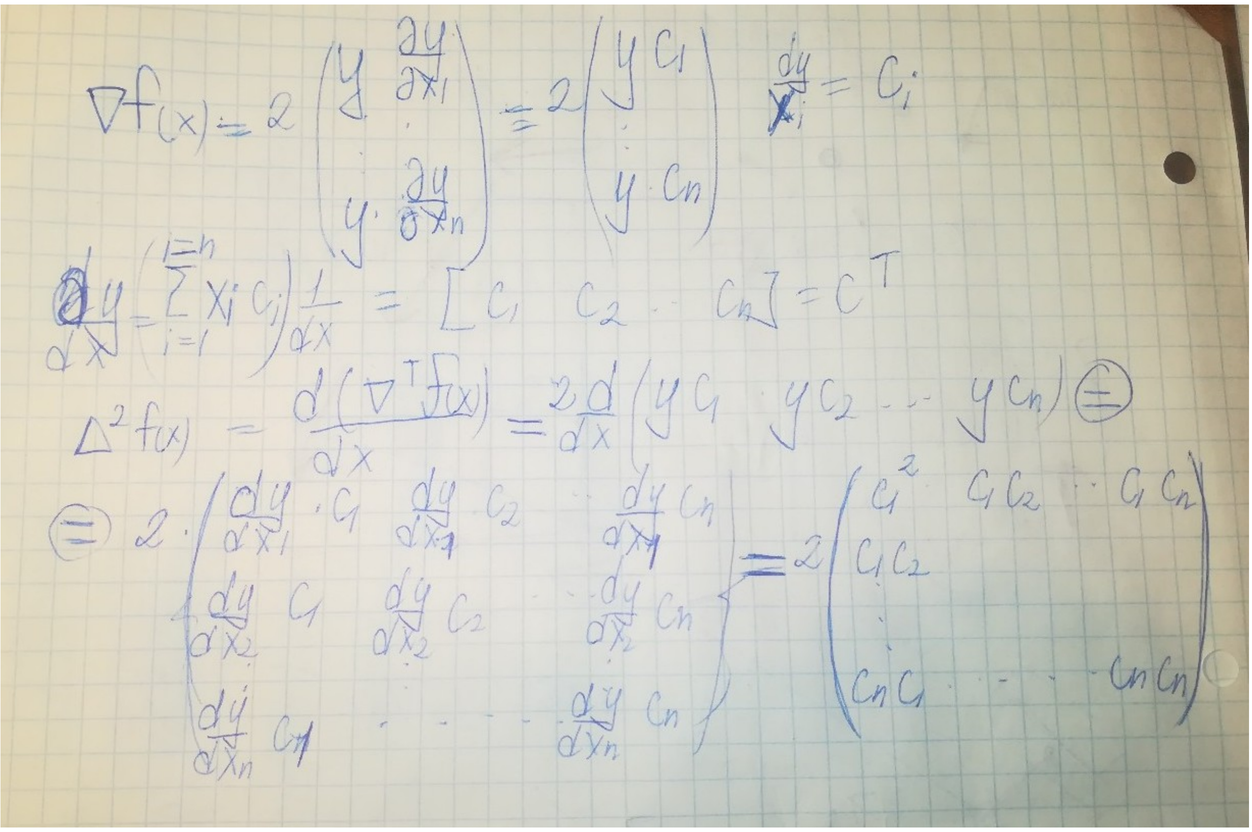
Solution: For strong convex functions stands: $f(y) \geq f(x) + (\nabla f(x), y - x) + \frac{\mu}{2}||x - y||^2$ $\forall x, y \in dom f$. For extremal point also stands that $\nabla f(x^*) = 0$. From substitution $x = x^*$ we get: $f(x) \geq f(x^*) + \frac{\mu}{2}||x^* - x||^2$ or $f(x) - f(x^*) \geq \frac{\mu}{2}||x^* - x||^2$ Q.E.D.

1.4 Task#4

Derive a) gradient $\nabla f(x)$ and b) Hessian matrix $\nabla^2 f(x)$ (both in vector form) for the function $f(x) = (x, c)^2; x \in R^n$

Solution:

Lets assign $y = (x, c)$



$$\nabla f(x) = 2 \begin{pmatrix} y \frac{\partial y}{\partial x_1} \\ \vdots \\ y \cdot \frac{\partial y}{\partial x_n} \end{pmatrix} = 2 \begin{pmatrix} y\, c_1 \\ \vdots \\ y \cdot c_n \end{pmatrix} \qquad \frac{dy}{dx_i} = c_i$$

$$\frac{dy}{dx} = \left(\sum_{i=1}^{i=n} x_i c_i\right) \frac{1}{dx} = [c_1 \; c_2 \; \dots \; c_n] = c^T$$

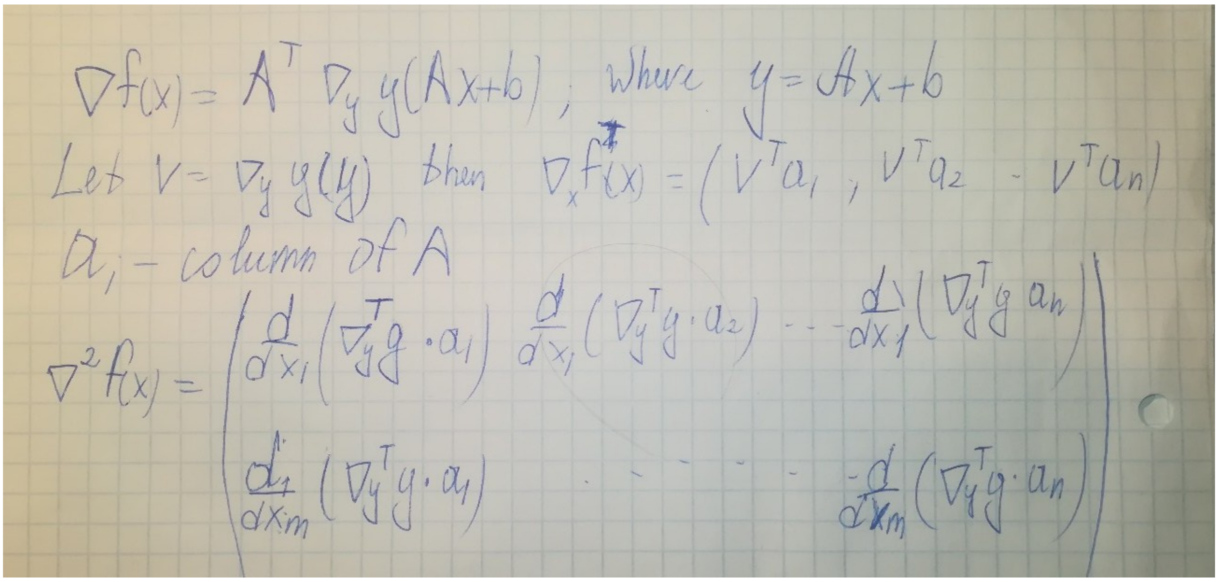
$$\Delta^2 f(x) = \frac{d(\nabla^T f(x))}{dx} = 2 \frac{d}{dx} (y c_1 \;\; y c_2 \; \dots \; y c_n) =$$

$$= 2 \cdot \begin{pmatrix} \frac{dy}{dx_1} \cdot c_1 & \frac{dy}{dx_1} c_2 & \dots & \frac{dy}{dx_1} c_n \\ \frac{dy}{dx_2} c_1 & \frac{dy}{dx_2} c_2 & \dots & \frac{dy}{dx_2} c_n \\ \frac{dy}{dx_n} c_1 & \dots & \dots & \frac{dy}{dx_n} c_n \end{pmatrix} = 2 \begin{pmatrix} c_1^2 & c_1 c_2 & \dots & c_1 c_n \\ c_1 c_2 & & & \\ \vdots & & & \\ c_n c_1 & \dots & \dots & c_n c_n \end{pmatrix}$$

Figure 1.1: Task #4: Solution

1.5 Task#5

Derive Hessian matrix $\nabla^2 f(x)$for the function $f(x) = g(Ax + b)$, assuming differentiable $g : R^m \to R$, with dimensions $A \in R^{(m \times n)}; b \in R^m; x \in R^n$.



$$\nabla f(x) = A^T \nabla_y g(Ax+b), \text{ where } y = Ax+b$$

Let $V = \nabla_y g(y)$ then $\nabla_x f(x) = (V^T a_1, V^T a_2 \dots V^T a_n)$

a_i – column of A

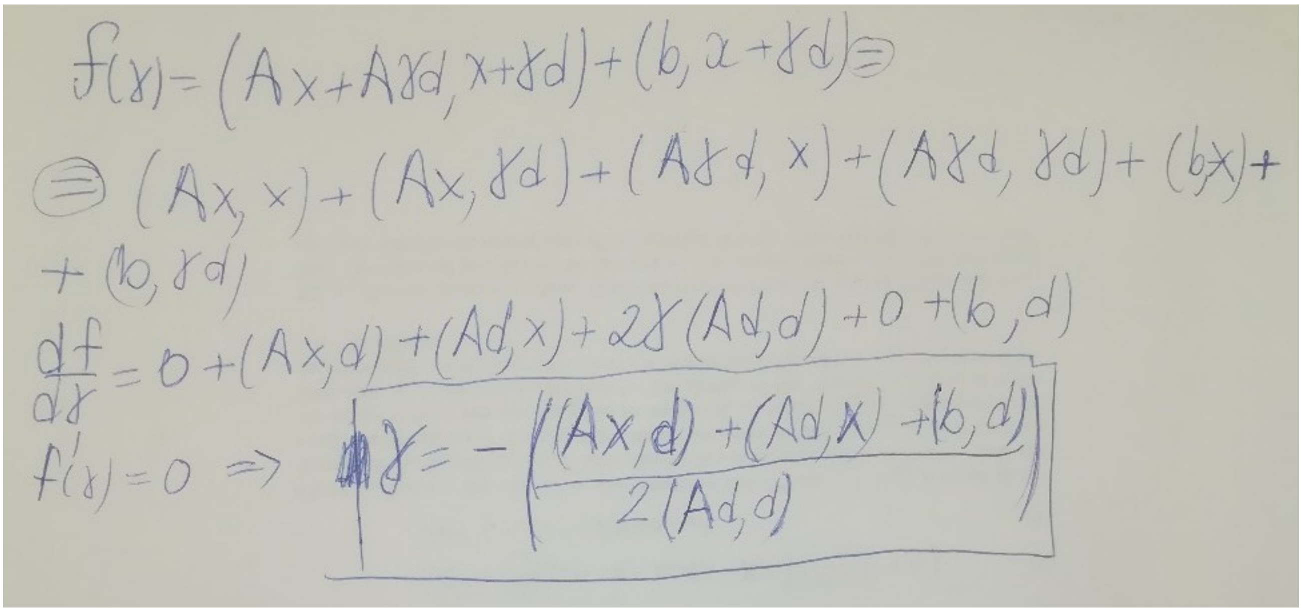
$$\nabla^2 f(x) = \begin{pmatrix} \frac{d}{dx_1}(\nabla_y^T g \cdot a_1) & \frac{d}{dx_1}(\nabla_y^T g \cdot a_2) & \dots & \frac{d}{dx_1}(\nabla_y^T g \, a_n) \\ \frac{d}{dx_m}(\nabla_y^T g \cdot a_1) & \dots & \dots & \frac{d}{dx_m}(\nabla_y^T g \cdot a_n) \end{pmatrix}$$

Figure 1.2: Task #5: Solution

1.6 Task#6

Solve optimal step-size problem for the quadratic function, with symmetric positive definite matrix $A >\in R^{n\times n}$, and $x, b, d \in R^n$. Your goal is to find optimal $\gamma forgiven A, b, d, x$.The resulting expression must be written in terms of inner products(...,...)

$f(\gamma) = (A(x+\gamma d), x+\gamma d) + (b, x+\gamma d) \rightarrow min \gamma \in R$



$$f(\gamma) = (Ax + A\gamma d, x+\gamma d) + (b, x+\gamma d) =$$

$$= (Ax, x) + (Ax, \gamma d) + (A\gamma d, x) + (A\gamma d, \gamma d) + (b, x) + (b, \gamma d)$$

$$\frac{df}{d\gamma} = 0 + (Ax, d) + (Ad, x) + 2\gamma(Ad, d) + 0 + (b, d)$$

$$f'(\gamma) = 0 \Rightarrow \boxed{\gamma = -\left(\frac{(Ax,d) + (Ad,x) + (b,d)}{2(Ad,d)}\right)}$$

Figure 1.3: Task #6: Solution

1.7 Task#7

Derive subgradient (subdifferential) for the function $f(x) = [x2{-}1]+, x \in R$.

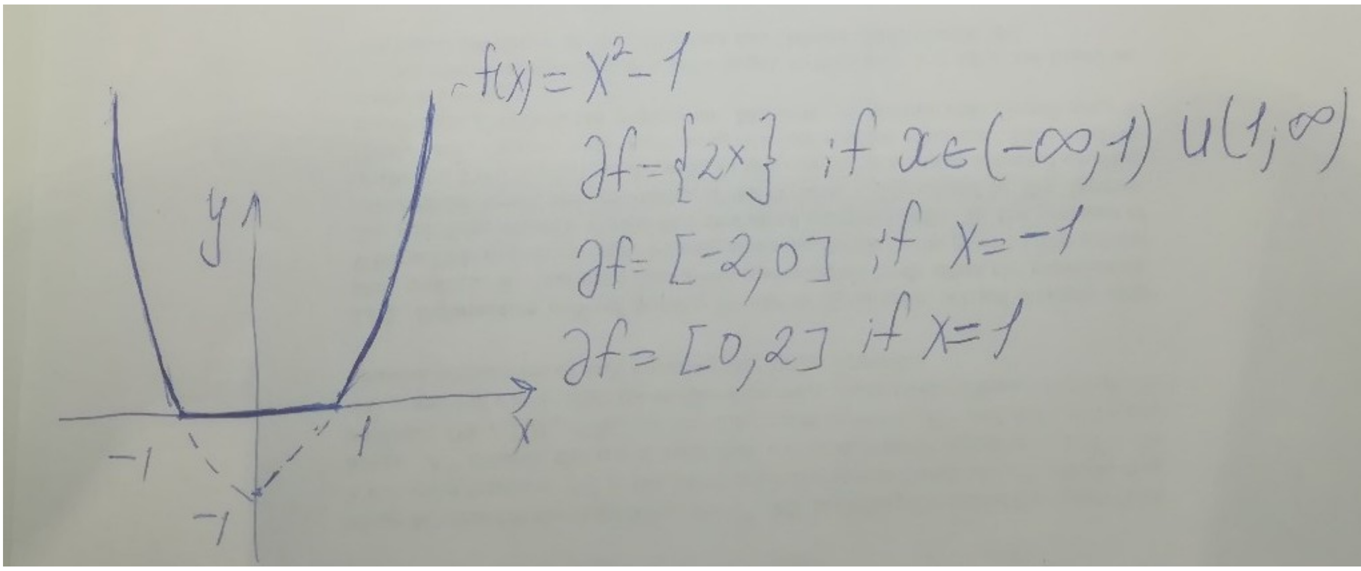


Figure 1.4: Task #7: Solution

1.8 Task#8

Let $f : R^n {\to} R$ be given by $f(x) = 12x > Qx{-}x > b$, where $b \in R^n$ and Q is a real symmetric matrix positive definite n×n matrix. Suppose that we apply steepest descent (or gradient descent) method to this function with $x^0 \neq Q^{-1}b$. Show that method converges in one step that is $x^1 = Q^{-1}b$, if and only if x^0 is chosen such that $g^0 = Qx^0{-}b$ is an eigenvector of Q.

Solution is on the Figure 1.5 (next page) :

1.9 Task#9

Find the minimizer of $f(x, y) = x^2 + xy + 10y^2{-}22y{-}5x$ numerically by steepest descent

1. For each iteration, record the values of x,y and f and include in a table.

2. Plot the values on a contour plot.

3. Explore different starting values, such as (1,10),(10,10),(10,1).Does the number of steps depend significantly on the starting guess?

Solution: The code is in the repository.

3. Amount of steps depends significantly on the starting point:

x0 = ([1.0,10.0]) = 5 iterations

x0 = ([10.0,10.0]) = 8 iterations

① $\nabla f(x)=0$; $\frac{d}{dx}(x^TQx) = 2x^TQ$ (symmetric $Q: Q^T=Q$)

$\nabla f(x) = x^TQ - b = 0$; $Q^Tx - b = 0$ $((AB)^T = B^TA^T)$

$\boxed{x = Q^{-1}b}$ q.e.d

② For steepest descent Assume that step $\gamma = 1.0$

$x^{k+1} = x^* = Q^{-1}b$; $x^{k+1} = x^0 - \nabla f(x^k)$

$Q^{-1}b = x^0 - (Qx^0 - b)$

$b = Qx^0 - Q(Qx^0 - b)$ (since ∇ is eigenvector)

$b = Qx^0 - Qx^0 + b \Rightarrow b = b$; Q.e.d.

Figure 1.5: Task #8: Solution

step	x	y	f(x,y)
0	15	-10	1220
1	14.24095917	0.4747634888	130.1704166
2	3.537601665	-0.300948240	1.288541941
3	3.447819527	0.937875377	-13.95516917
4	2.181837049	0.8461375452	-15.75817663
5	2.171220009	0.9926532766	-15.97140187
6	2.02150715	0.9818029918	-15.9966175
7	2.020251327	0.9991310345	-15.99959993
8	2.002543409	0.9978478474	-15.99995269
9	2.002394907	0.9998972396	-15.9999944
10	2.000300838	0.9997454699	-15.99999934
11	2.000283271	0.9999878523	-15.99999992

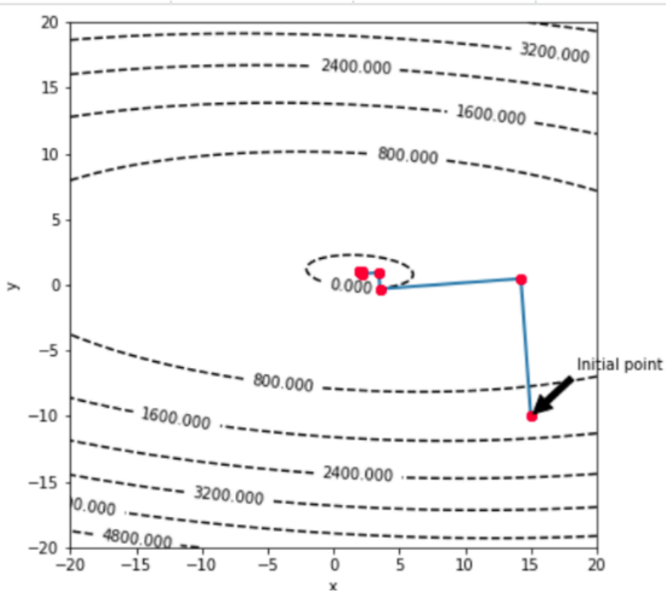


Figure 1.6: Task #9: Solution to point 1 and 2

x0 = ([10.0,1.0]) = 36 iterations

On the picture ”Task #9: comments on point 3 ” demonstrated the case when x0 = ([10.0,1.0]). The steepest descent moves very slowly when the function values become flat

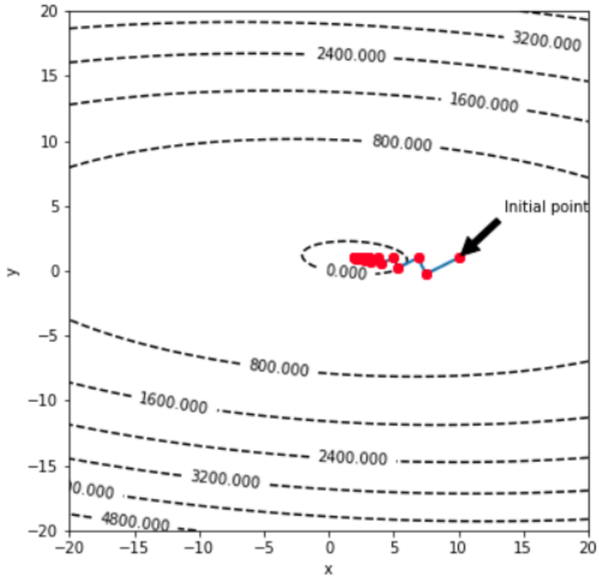


Figure 1.7: Task #9: comments on point 3

1.10 Task#10

Let the cost function of the unconstrained optimization problem of interest be

$$f(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{1=2}^{n}(2 * x_{i-1} - x_i - 1)^2$$

Consider two different scenarios, first n= 3, and then n= 10. Using $x^0 = [-1.5, 1, ..., 1]^T$ write out the first iteration of the steepest descent algorithm and obtain the optimum value for α^0. What is the value of x^1 if you implement α^0? Verify that f(x1)< f(x0)

2. Write a code to find the minimizer of f(x) using steepest descent algorithm with starting point of $x^0 = [-1.5, 1, ..., 1]$ and using 1) $x^1 = argminf(x^k - \alpha \nabla f(x^k))$.

2) constantα = 0.1.

3) constantα = 0.5.

4) constantα = 1.0.

(Use $||x^{k+1} - x^k|| \leq 10^{-6}$ the stopping condition for your algorithm)

3. How many steps it takes for the algorithm to converge for each choices of the step sizes above?

4. Use fminbnd from Matlab or an equivalent function from the programming language of your-choice to solve the problem. How many steps this algorithm takes to converge?

Solution:

4. Comment: Gradient norm in n = 10 faster converges to eps.

In python only conjugate gradient is available as a part of scipy library.

Output from the library solver n = 3:

Optimization terminated successfully.

	f(x0) = 7.8125			
n	step	x1	f(x1)	f(x1) < f(x0)
3	0.01588745	[-1.00351715 1.07943726 1.]	1.1169341	TRUE
10	0.01588745	[-1.00351715 1.07943726 1. 1.1. 1. 1. 1. 1. 1.]	1.116934129	TRUE

Figure 1.8: Task #10: point 1 solution

n	alpha	iterations	f(xn)
3	argmin	43836	7.24E-08
3	0.1	doesn't converge	undefined
3	0.5	doesn't converge	undefined
3	1	doesn't converge	undefined
10	argmin	15	0.9830311985
10	0.1	doesn't converge	undefined
10	0.5	doesn't converge	undefined
10	1	doesn't converge	undefined

Figure 1.9: Task #10: point 2 and 3 solution

Current function value: 0.000000

Iterations: 132

Function evaluations: 1275

Gradient evaluations: 255

x: [0.9998953 0.99957355 0.99829311]

Output from the library solver n = 10:

Optimization terminated successfully.

Current function value: 0.983031

Iterations: 13

Function evaluations: 336

Gradient evaluations: 28

x: [0.9998953 0.99957355 0.99829311]

Conclusion: conjugate gradient method from the library requires less iterations compare to steepest gradient descent