MTH8408: Exercices

André Phu-Van Nguyen

Unconstrained optimization (slides)

Exercise 6/11

Prove the first and second order optimality conditions using

$$f(x^* + d) - f(x^*) = \nabla f(x*)^T d + \frac{1}{2} d^T \nabla^2 f(x^*) d + \mathrm{o}(||d||^2)$$

Answer

N1 if $x*$ is a local minimum and $f \in \mathcal{C}^1$, then $\nabla f(x^*) = 0$

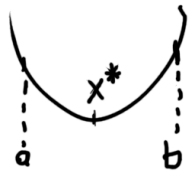
If $\nabla f(x^*) = 0$ then $\frac{1}{2} d^T \nabla^2 f(x^*) d = 0$ thus we only consider

$$f(x^* + d) - f(x^*) = \nabla f(x*)^T d + \mathrm{o}(||d||^2)$$

Supposing that $\mathrm{o}(||d||^2)$ can be neglected we can rearrange the equation into

$$\frac{f(x^* + d) - f(x^*)}{d} = \nabla f(x*)^T$$

which is close to the definition of a limit.



For an $x \in [a, b]$ we have $f(x) \geq f(x*)$

From a to x^*

We have $x < x^*$ and $f(x) - f(x^*) \geq 0$ thus

$$\frac{f(x) - f(x^*)}{x - x^*} \leq 0$$

From b to x^*

We have $x > x^*$ and $f(x) - f(x^*) \geq 0$ thus

$$\frac{f(x) - f(x^*)}{x - x^*} \geq 0$$

Going back to the definition of a limit we have

$$\lim_{d \to \infty} \frac{f(x^* + d) - f(x)}{d} \leq 0$$

$$\lim_{d \to \infty} \frac{f(x^* + d) - f(x)}{d} \geq 0$$

N2 if x^* is a local minimum and $f \in \mathcal{C}^2$, then $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) = 0$

If $f \in \mathcal{C}^2$ then $f \in \mathcal{C}^1$ is also true, thus the previous results apply.

??? Incomplete

Exercise 10/71 - Unconstrained optimization with a quadratic objective

In S2, show that s^* is in fact a *global* minimum!

Answer

S2 if $Hs^* = -g$ and $H \prec 0$ then s^* is a local minimum

Since $q(s)$ is a quadratic, a local minimum $s*$ of q is automatically a global minimum.

Exercise 46/71 - Conjugate gradient ???

Show that conjugate vectors are necessarily linearly independent.

Answer

Proof by contradiction. If p_i is linearly dependant to a p_j then we have

$$a_0 p_0 + a_1 p_1 + \ldots + a_n p_n = 0 \qquad n \in \mathbb{Z}^*$$

Where all a_n are non-zero. Multiply by A

$$a_i p_i^T A = 0$$

Take the scalar product with p_i

$$a_i p_i^T A p_i = 0$$

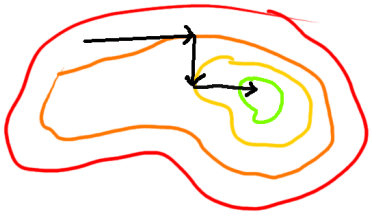
We know that $A \succ 0$ which means that $p_i^T A p_i > 0$ thus $a_i = 0$ which is contradictory to a linearly dependant system.

Exercise 47/71 - Conjugate gradient

Explain geometrically why $r_{k+1} \perp r_k$ i.e. $r_{k+1}^T r_k = 0$

Answer

When doing a linesearch we stop at a point where the direction is minimal. This also means we are tangent to the contour line (level curve), thus the next direction is orthogonal to the current one.



Exercise 47/71 - Conjugate gradient

Find a recurrence to update $q(x_k)$ in the conjugate gradient algorithm.

Answer

Should be something like

$$q(x_{k+1}) = q(x_k) + \nabla q(x_k)\alpha_k A p_k$$

So the current value of q plus the change in q and how much of that change.

??? Incomplete

Exercice 48/71 - Conjugate gradient

Prove $A \succ 0$ if and only if $p_k^T A p_k > 0$ for all $k = 0, \ldots, n-1$

Answer

We know that $A \succ 0$ if all the eigen values λ of A are positive and

$$A\mathbf{v} = \lambda\mathbf{v}$$

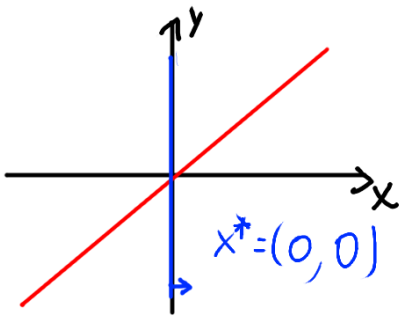
so we can rewrite the expression as

$$\begin{aligned} p_k^T A p_k = p_k^T \lambda p_k &> 0 \\ \lambda p_k^T p_k &> 0 \\ \lambda \|p_k\|^2 &> 0 \end{aligned}$$

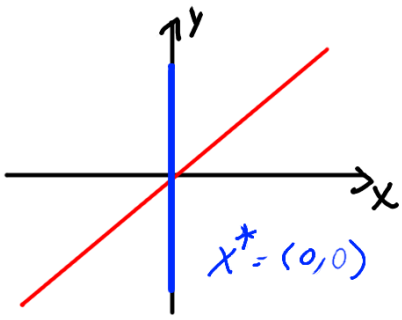
Since $\|p_k\|^2$ must be positive then λ must also be positive. Therefore $A \succ 0$.

Modeling problems

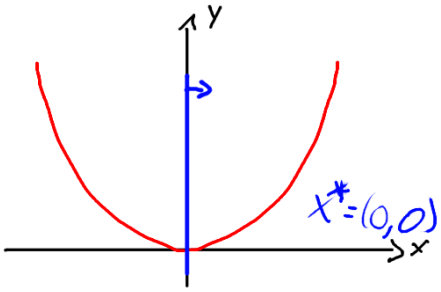
Problem 1



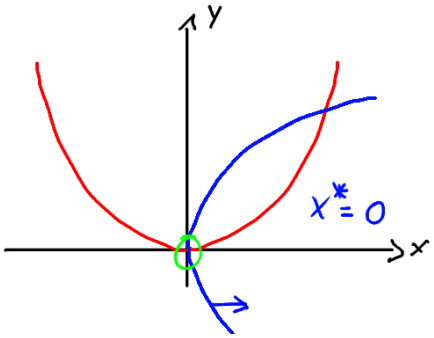
1. a)



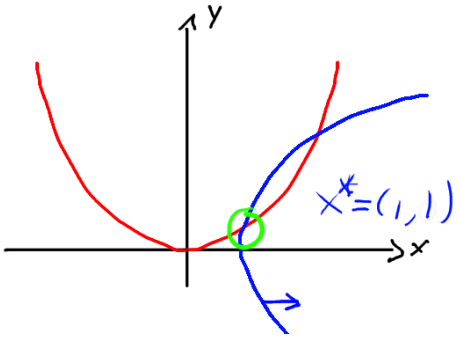
1. b)



1. c)

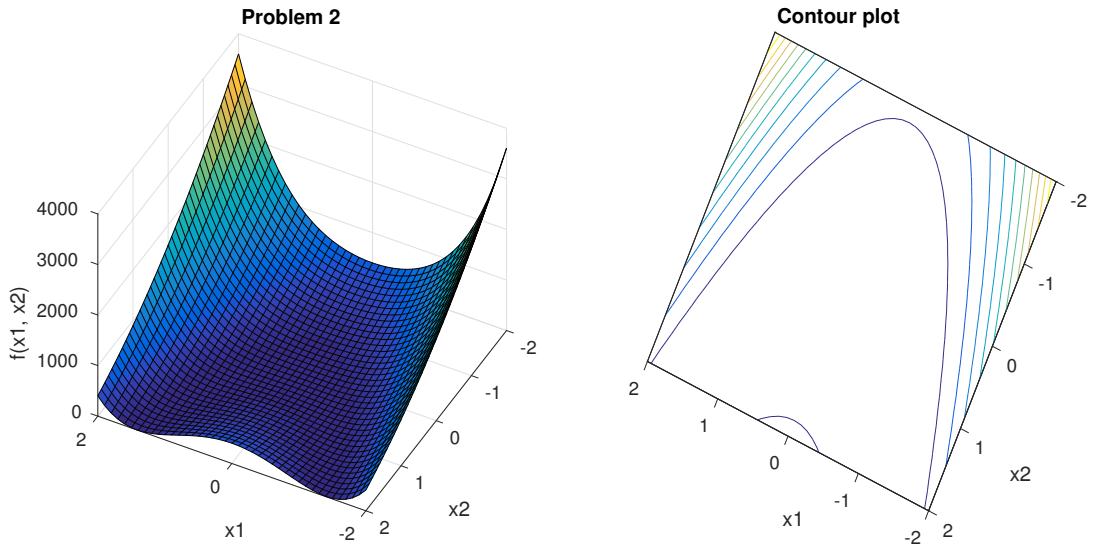


1. d)



1. e)

Problem 2



2.a) We see that the two terms containing variables are square therefore the minimum is probably 0. We can get this result with $x^* = (1, 1)$ it is unique because of it is the only root of $100(x_2 - x_1^2)^2 + (1 - x_1)^2$.

2.b)

```
model;

var x1;
var x2;

minimize function:
    100 * (x2 - x1^2)^2 + (1 - x1)^2
```

Output:

```
ampl: model modeling_pb2.mod;
ampl: solve;
MINOS 5.51: optimal solution found.
17 iterations, objective 9.247040366e-19
Nonlin evals: obj = 47, grad = 46.
ampl: display x1, x2;
x1 = 1
x2 = 1
```

Which confirms answer in **2.a)**.

2.c) Add an equality constraint which changes the solution.

```
model;

var x1;
var x2;

minimize function:
    100 * (x2 - x1^2)^2 + (1 - x1)^2;
```

```
8
9  subject to changeSolutionEqualityConstraint:
10     x1 = 2;
```

Output:

```
1  ampl: model modeling_pb2.mod;
2  ampl: solve;
3  MINOS 5.51: optimal solution found.
4  2 iterations, objective 1
5  Nonlin evals: obj = 6, grad = 5.
6  ampl: display x1, x2;
7  x1 = 2
8  x2 = 4
```

2.d) Add an equality constraint which **does not** change the solution.

```
1  model;
2
3  var x1;
4  var x2;
5
6  minimize function:
7      100 * (x2 - x1^2)^2 + (1 - x1)^2;
8
9  subject to dontChangeSolutionEqualityConstraint:
10     x1 = 1;
```

Output:

```
1  ampl: model modeling_pb2.mod;
2  ampl: solve;
3  MINOS 5.51: optimal solution found.
4  2 iterations, objective 0
5  Nonlin evals: obj = 6, grad = 5.
6  ampl: display x1, x2;
7  x1 = 1
8  x2 = 1
```

2.e) Add an inequality constraint which changes the solution.

```
1  model;
2
3  var x1;
4  var x2;
5
6  minimize function:
7      100 * (x2 - x1^2)^2 + (1 - x1)^2;
8
9  subject to changeSolutionInequalityConstraint:
10     x1 >= 2;
```

Output:

```
1  ampl: model modeling_pb2.mod;
2  ampl: solve;
3  MINOS 5.51: optimal solution found.
4  2 iterations, objective 1
```

```
Nonlin evals: obj = 6, grad = 5.
ampl: display x1, x2;
x1 = 2
x2 = 4
```

2.f) Add an inequality constraint which **does not** changes the solution.

```
model;

var x1;
var x2;

minimize function:
    100 * (x2 - x1^2)^2 + (1 - x1)^2;

subject to dontChangeSolutionInequalityConstraint:
    x1 <= 2;
```

Output:

```
ampl: model modeling_pb2.mod;
ampl: solve;
MINOS 5.51: optimal solution found.
17 iterations, objective 9.247040366e-19
Nonlin evals: obj = 47, grad = 46.
ampl: display x1, x2;
x1 = 1
x2 = 1
```

Problem 3

Model:

```
model;

var x1;
var x2;

minimize function:
    (x1 - 2)^2 + (x2 -1)^2;

subject to zeroConstraint:
    x1^2 - x2 <= 0;

subject to twoConstraint:
    x1 + x2 <= 2;
```

Output:

```
ampl: model modeling_pb3.mod;
ampl: solve;
MINOS 5.51: optimal solution found.
12 iterations, objective 1
Nonlin evals: obj = 33, grad = 32, constrs = 33, Jac = 32.
ampl: display x1, x2;
x1 = 1
x2 = 1
```

Problem 4

4.a)

$$\begin{aligned} P_3(x) &= \frac{f^{(0)}(x_0)}{0!} + \frac{f^{(1)}(x_0)}{1!}(x-x_0) + \frac{f^{(2)}(x_0)}{2!}(x-x_0)^2 + \frac{f^{(3)}(x_0)}{3!}(x-x_0)^3 \\ &= \cos(1) + \frac{-\sin(1)}{1}(x-1) + \frac{-\cos(1)}{2}(x-1)^2 + \frac{\sin(1)}{6}(x-1)^3 \end{aligned}$$

For x = 1.1 we have

$$\begin{aligned} P_3(1.1) &= \cos(1) + (-0.0841) + (-0.0027) + (0.0001) \\ &= 0.4536 \end{aligned}$$

4.b)

$$\begin{aligned} f(x) &= \cos(x) \\ f'(x) &= \sin(\frac{1}{x})\frac{1}{x^2} \\ f''(x) &= \Big(\sin(\frac{1}{x})\Big)'\frac{1}{x^2} + \sin(\frac{1}{x})\Big(\frac{1}{x^2}\Big)' \\ &= \cos(\frac{1}{x})\frac{-1}{x^2}\frac{1}{x^2} + \sin(\frac{1}{x})\frac{-2}{x^3} \\ &= -\frac{\cos(\frac{1}{x}) + 2x\sin(\frac{1}{x})}{x^4} \end{aligned}$$

Thus the 2nd order Taylor expansion with $x_0 = 1$ is

$$P_2(x) = \cos(1) + \sin(1)(x-1) - \Big(\frac{\cos(1) + 2\sin(1)}{2}\Big)(x-1)^2$$

And for $x = 1.1$ we have

$$\begin{aligned} P_2(1.1) &= \cos(1) + \sin(1)(1.1-1) - \Big(\frac{\cos(1) + 2\sin(1)}{2}\Big)(1.1-1)^2 \\ &= 0.5403 + 0.0841 - 0.0111 \\ &= 0.6133 \end{aligned}$$

4.c)

$$\begin{aligned} f_{x_1} &= -200(x_1^2 + 2x_1 - x_2) + 2(x_1 - 1) \\ f_{x_2} &= 200(x_2 - x_1^2) \\ f_{x_1x_1} &= -200(2x_1 + 2) + 2 \\ f_{x_1x_2} &= -200(-1) = 200 \\ f_{x_2x_1} &= 200(1 - 2x_1) \\ f_{x_2x_2} &= 200 \end{aligned}$$

Thus the Taylor expansion in matrix form is

$$f(x,y) = f(0,0) + \begin{bmatrix} x \\ y \end{bmatrix}^T \begin{bmatrix} -200(x_1^2 + 2x_1 - x_2) + 2(x_1 - 1) \\ 200(x_2 - x_1^2) \end{bmatrix} + \frac{1}{2!} \begin{bmatrix} x \\ y \end{bmatrix}^T \begin{bmatrix} -200(2x_1 + 2) + 2 & 200 \\ 200(1 - 2x_1) & 200 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

4.d)

Around the point $(x, y) = (a, b)$ we have

$$f(x,y) = f(a,b) + \left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} a \\ b \end{bmatrix} \right)^T \begin{bmatrix} f_x(a,b) \\ f_y(a,b) \end{bmatrix} + \frac{1}{2!} \left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} a \\ b \end{bmatrix} \right)^T \begin{bmatrix} f_{xx}(a,b) & f_{xy}(a,b) \\ f_{yx}(a,b) & f_{yy}(a,b) \end{bmatrix} \left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} a \\ b \end{bmatrix} \right)$$

Problem 5

In the folowing problems we denote $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \dots x_n \end{bmatrix}^T$

Function f_1

$$f_1(x) = c^T \mathbf{x}$$

Using the identity $\frac{\partial A\mathbf{x}}{\partial \mathbf{x}} = A^T$ the gradient is

$$\nabla f_1(\mathbf{x}) = \frac{\partial c^T \mathbf{x}}{\partial \mathbf{x}} = c$$

And the Hessian is given by

$$\frac{\partial c}{\partial \mathbf{x}} = 0$$

since the gradient was all constants.

Function f_2

$$f_2(x) = \frac{1}{2} x^T H x$$

To find the gradient we apply the fundamental identity [1]

[1] https://en.wikipedia.org/wiki/Matrix_calculus

$$\frac{\partial(\mathbf{u}\cdot\mathbf{A}\mathbf{x})}{\partial\mathbf{x}} = \frac{\partial(\mathbf{u}^T\mathbf{A}\mathbf{x})}{\partial\mathbf{x}} = \frac{\partial\mathbf{u}}{\partial\mathbf{x}}\mathbf{A}\mathbf{v} + \frac{\partial\mathbf{v}}{\partial\mathbf{x}}\mathbf{A}^T\mathbf{u}$$

In our case $\mathbf{u} = \mathbf{v} = \mathbf{x}$

$$\begin{aligned}\frac{1}{2}\frac{\partial\mathbf{x}^T H\mathbf{x}}{\partial\mathbf{x}} &= \frac{1}{2}\left(\frac{\partial\mathbf{x}}{\partial\mathbf{x}}H\mathbf{x} + \frac{\partial\mathbf{x}}{\partial\mathbf{x}}H^T\mathbf{x}\right)\\ &= \frac{1}{2}\left(H\mathbf{x} + H^T\mathbf{x}\right)\end{aligned}$$

And since H is a square symmetric matrix we can write

$$= H\mathbf{x}$$

Thus the Hessian is

$$\frac{\partial H\mathbf{x}}{\partial\mathbf{x}} = H$$

again through a fundamental identity.

Function $f_1 + f_2$

For

$$f_3(x) = f_1(x) + f_2(x) = c^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T H\mathbf{x}$$

by the sum rule, the gradient is the sum of the results derived above

$$\frac{\partial f_3(x)}{\partial\mathbf{x}} = c + H\mathbf{x}$$

and the Hessian is

$$\frac{\partial(c + H\mathbf{x})}{\partial\mathbf{x}} = H$$

Problem 6 ???

Where

$$g(x) = f(Sx)$$

we have through the chain rule

$$\frac{\partial g(x)}{\partial x} = \frac{\partial x}{\partial x}\frac{\partial Sx}{\partial x}\frac{\partial f(Sx)}{\partial Sx} = S\frac{\partial f(Sx)}{\partial Sx} = S\frac{\partial f(x)}{\partial x}$$

and the Hessian is

$$\frac{\partial}{\partial x}S\frac{\partial f(x)}{\partial x} = S\frac{\partial^2 f(x)}{\partial^2 x}$$

Unconstrained Programming Exercices

Problem 1

Discutez, suivant les valeurs du paramètre $\beta \in \mathrm{R}$, la nature des points stationnaires de la fonction de deux variables

$$f(x, y) = x^2 + y^2 + \beta xy + x + 2y$$

Answer

$$\nabla f(x, y) = \begin{bmatrix} 2x + \beta y + 1 \\ 2y + \beta x + 2 \end{bmatrix}$$

$$\mathbf{H}(f(x, y)) = \begin{bmatrix} 2 & \beta \\ \beta & 2 \end{bmatrix}$$

Find the eigen values

$$\begin{aligned} \det(\lambda I - \mathbf{H}) &= 0 \\ \det\left(\begin{bmatrix} \lambda - 2 & -\beta \\ -\beta & \lambda - 2 \end{bmatrix} \right) &= 0 \\ (\lambda - 2)^2 - \beta^2 &= 0 \\ \lambda^2 - 4\lambda + y - \beta^2 &= 0 \end{aligned}$$

Using the quadratic formula

$$\begin{aligned} \lambda &= \frac{4 \pm \sqrt{16 - 4 \cdot 1 \cdot (4 - \beta^2)}}{2} \\ &= \frac{4 \pm \sqrt{4\beta^2}}{2} \\ &= \frac{4 \pm 2\beta}{2} \\ &= 2 \pm \beta \end{aligned}$$

$\beta < -2$	$\beta \in [-2, 2]$	$\beta > 2$
$\lambda_1 < 0$, $\lambda_2 > 0$ Indefinite matrix	$\lambda_1 = [0, 4]$, $\lambda_2 = [4, 0]$ Semi-definite positive matrix	$\lambda_1 > 0$, $\lambda_2 < 0$ Indefinite matrix
Saddle point	local minimum	saddle point

Problem 2

Soit la minimisation sans contraintes de la fonction

$$f(x_1, x_2) = 2x_1^2 + x_2^2 - 2x_1x_2 + 2x_1^3 + x_1^4$$

(a) Trouvez les candidats solution du problème et pour chacun d'entre eux, précisez s'il s'agit d'un minimum local, d'un maximum local ou d'un point selle. Justifiez vos réponses.

(b) Y a-t-il un minimum global dans ce cas? Si oui, lequel?

Answer

a)

$$\nabla f(x_1, x_2) = \begin{bmatrix} 4x_1 - 2x_2 + 6x_1^2 + 4x_1^3 \\ 2x_2 - 2x_1 \end{bmatrix} = 0$$

$$2x_2 - 2x_1 \Rightarrow x_2 = x_1$$

Substituting into the first line we get

$$\begin{aligned} 4x_1^3 + 6x_1^2 + 2x_1 &= 0 \\ 2x_1^3 + 3x_1^2 + x_1 &= 0 \\ x_1(2x_1 + 1)(x_1 + 1) &= 0 \\ x_1 = 0, -1 \text{ or } &-\frac{1}{2} \end{aligned}$$

Our 3 stationary points are $(0, 0)$, $(-1, -1)$ and $(-\frac{1}{2}, -\frac{1}{2})$

$$\mathbf{H} = \begin{bmatrix} 12x_1^2 + 12x_1 + 4 & -2 \\ -2 & 2 \end{bmatrix}$$

Find the eigen values of the hessian

$$\begin{aligned} \det(\lambda I - \mathbf{H}) &= 0 \\ \det\left(\begin{bmatrix} \lambda - 4(3x_1^2 + 3x_1 + 1) & -2 \\ -2 & \lambda - 2 \end{bmatrix} \right) &= 0 \\ \Big(\lambda - 4(3x_1^2 + 3x_1 + 1)\Big)(\lambda - 2) - 4 &= 0 \end{aligned}$$

For $(x_1, x_2) = (0, 0)$ we have

$$\begin{aligned} (\lambda - 4(0 - 0 + 1))(\lambda - 2) - 4 &= 0 \\ (\lambda - 4)(\lambda - 2) - 4 &= 0 \\ \lambda^2 - 6\lambda + 4 &= 0 \\ \lambda &= \frac{6 \pm \sqrt{36 - 4 \cdot 4}}{2} \\ \lambda &= 3 \pm \sqrt{5} \\ \lambda_1 > 0, \lambda_2 &> 0 \end{aligned}$$

Both values of λ are positive making the Hessian definite positive at $(0, 0)$ making the point a local minimum.

For $(x_1, x_2) = (1, 1)$ we have

$$(\lambda - 4(3 - 3 + 1))(\lambda - 2) - 4 = 0$$
$$(\lambda - 4)(\lambda - 2) - 4 = 0$$

Which goes back to the same answer as above, thus $(1, 1)$ is also a local minimum.

For $(x_1, x_2) = (-\frac{1}{2}, -\frac{1}{2})$ we have

$$(\lambda - 4(-\frac{3}{4} - \frac{6}{4} + \frac{4}{4}))(\lambda - 2) - 4 = 0$$
$$(\lambda + 5)(\lambda - 2) - 4 = 0$$
$$\lambda^2 + 3\lambda - 14 = 0$$
$$\lambda = \frac{-3 \pm \sqrt{9 - 4 \cdot 1 \cdot -14}}{2}$$
$$\lambda = \frac{-3 \pm \sqrt{65}}{2}\lambda \qquad = 2.5311, -5.5311$$

Meaning the Hessian is indefinite at $(-\frac{1}{2}, -\frac{1}{2})$ making it a saddle point.

Problem 3

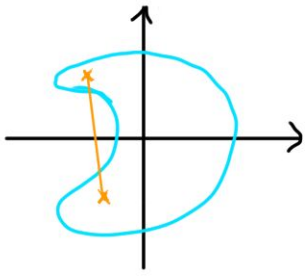
On dit qu'un ensemble $C \subseteq \mathrm{R}^n$ est convexe si

$$\lambda x + (1 - \lambda)y \in C \quad \forall \lambda \in [0, 1],\ x, y \in C$$

a) Donnez une interprétation géométrique de cette propriété et tracez un sous-ensemble de R^2 qui n'est pas convexe.

Answer

We have two points x and y in C. If we draw a line from x to y, the entire line has to always stay in C.



b)

Answer[2]

Let $x,\ y\ \in C_\alpha$, take f of the convex equation

$$f\Big(\lambda x + (1-\lambda)y\Big) \geq f(\lambda x) + f\Big((1-\lambda)y\Big)$$

Since f is convex we can write by definition that

$$f(\lambda x + (1-\lambda)y) \geq f(\lambda x) + f((1-\lambda)y) \geq \lambda f(x) + (1-\lambda)f(y)$$

Since $x,\ y\ \in C_\alpha$ we can write

$$f(\lambda x + (1-\lambda)y) \geq \lambda\alpha + (1-\lambda)\alpha$$

Which is the same as

$$f(\lambda x + (1-\lambda)y) \geq \alpha$$

Thus C_α is convex **c)**

Answer

The epigraph is the volume on top of a curve, including the curve itself. So we choose two points (x_1, t_1) and (x_2, t_2) both $\in \text{epi}(f)$ meaning that

$$f(x_1) \leq t_1$$
$$f(x_2) \leq t_2$$

To show that $\text{epi}(f)$ is convex we need to show that

$$\lambda(x_1, t_1) + (1-\lambda)(x_2, t_2) \in \text{epi}(f)$$

We can rewrite the lhs as the point

$$\Big(\lambda x_1 + (1-\lambda)x_2,\ \lambda t_1 + (1-\lambda)t_2\Big) \in \text{epi}(f)$$

which would mean that the value of f from x_1 to x_2 should be less than the line t_1 to t_2 for any $\lambda \in [0, 1]$ i.e.

$$f(\lambda x_1 + (1-\lambda)x_2) \overset{?}{\leq} \lambda t_1 + (1-\lambda)t_2$$

Since f is convex we can write

$$f(\lambda x_1 + (1-\lambda)x_2) \leq\ \lambda f(x_1) + (1-\lambda)f(x_2)\ \leq \lambda t_1 + (1-\lambda)t_2$$

which means the previous equation is true, thus $\text{epi}(f)$ is convex.

d)

Answer

We want to show that for two points $x_1,\ x_2 \in M$ we have

$$\lambda x_1 + (1-\lambda)x_2 \in M$$

[2] `http://ljk.imag.fr/membres/Anatoli.Iouditski/cours/convex/chapitre_3.pdf` page 57

We can do this by showing that the linear combination of x_1 and x_2 is still a minimum of f.

$$f(\lambda x_1 + (1-\lambda)x_2) \overset{?}{\leq} f(x) \quad \forall x$$

Since f is convex we can write

$$\lambda f(x_1) + (1-\lambda) f(x_2) \overset{?}{\leq} f(x)$$

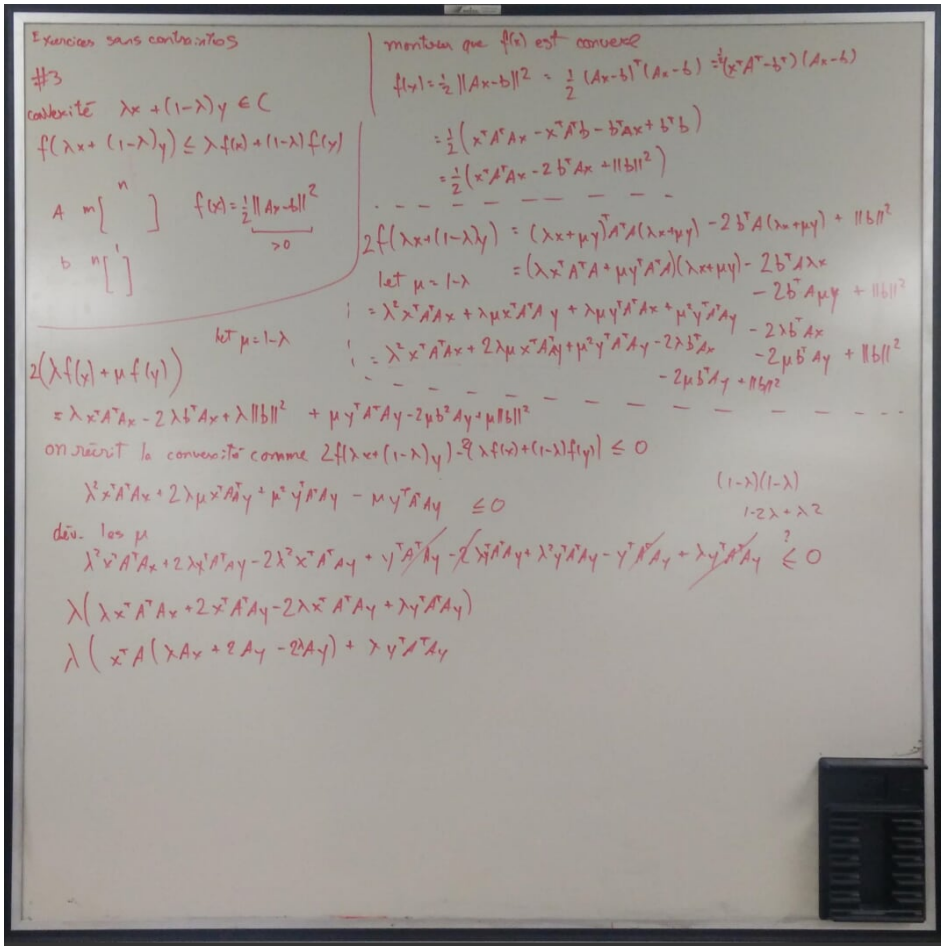
But since x_1 and x_2 are both global minima of f then $f(x_1) = f(x_2)$ and we can write

$$\big(\lambda + (1-\lambda)\big) f(x_1) \overset{?}{\leq} f(x)$$
$$f(x_1) \overset{?}{\leq} f(x)$$

Which is true since we know x_1 is a global minimum. Thus the set M is convex.

e)

Answer



Terms are supposed to cancel out...

??? Incomplete

On a parlé d'un critère basé sur $\nabla^2 f(x)$

Problem 4

a)

Answer

Go back to the taylor expansion where

$$f(x+h) = f(x) + \nabla f(x)^T h + \mathrm{o}(\|h\|)$$

So we apply this to $f(x)$ which gives

$$\frac{1}{2}\|F(x+h)\|^2 = \frac{1}{2}\|F(x) + \nabla F(x)^T h + \mathrm{o}(\|h\|)\|^2$$

Recall the identity $\|a+b+c\|^2 = \|a\| + \|b\| + \|c\| + 2 < a, b > +2 < a, c > +2 < b, c >$

$$= \frac{1}{2}\|F(x)\|^2 + \frac{1}{2}\|\nabla F(x)^T h\|^2 + F(x)^T \nabla F(x)^T h + F(x) o(\|h\|) + \nabla F(x)^T h F(x) o(\|h\|)$$

All the red terms above are basically the error term of the Taylor expansion. By inspection we can then see that

$$\nabla f(x)^T = F(x)^T \nabla F(x)^T$$
$$\nabla f(x) = F(x) \nabla F(x)$$

$\nabla f(x) = \nabla F(x)\, F(x)$

Where if we supopose $F : \mathbb{R}^n \to \mathbb{R}^n$ then

$$F(x_1, \ldots, x_n) = \begin{bmatrix} F_1(x_1, \ldots, x_n) \\ \vdots \\ F_n(x_1, \ldots, x_n) \end{bmatrix}$$

and

$$\nabla F(x) = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_2}{\partial x_1} & \cdots & \frac{\partial F_n}{\partial x_1} \\ \frac{\partial F_1}{\partial x_2} & & & \\ \vdots & & & \\ \frac{\partial F_1}{\partial x_n} & \cdots & & \frac{\partial F_n}{\partial x_n} \end{bmatrix}$$

en classe on a noté $J(x) = \nabla F(x)^T$

b)

Since the gradient above is shown to be 2D we need to add a third dimension a ?tensor? to describe the Hessian. So we have something like

$$\nabla^2 F(x) = \begin{bmatrix} \frac{\partial F_1}{\partial x_1 x_1} & \frac{\partial F_1}{\partial x_1 x_2} & \cdots & \frac{\partial F_1}{\partial x_1 x_n} \\ \frac{\partial F_1}{\partial x_2 x_1} & & & \\ \vdots & & & \\ \frac{\partial F_1}{\partial x_n x_1} & \cdots & & \frac{\partial F_1}{\partial x_n x_n} \end{bmatrix}, \begin{bmatrix} \frac{\partial F_2}{\partial x_1 x_1} & \frac{\partial F_2}{\partial x_1 x_2} & \cdots & \frac{\partial F_2}{\partial x_1 x_n} \\ \frac{\partial F_2}{\partial x_2 x_1} & & & \\ \vdots & & & \\ \frac{\partial F_2}{\partial x_n x_1} & \cdots & & \frac{\partial F_2}{\partial x_n x_n} \end{bmatrix}, \ldots, \begin{bmatrix} \frac{\partial F_n}{\partial x_1 x_1} & \frac{\partial F_n}{\partial x_1 x_2} & \cdots & \frac{\partial F_n}{\partial x_1 x_n} \\ \frac{\partial F_n}{\partial x_2 x_1} & & & \\ \vdots & & & \\ \frac{\partial F_n}{\partial x_n x_1} & \cdots & & \frac{\partial F_n}{\partial x_n x_n} \end{bmatrix}$$

ok, it is the collection of $\nabla^2 F_i(x)$.

Which allows us to describe the Hessian of $f(x)$ as

$$\begin{aligned}\nabla^2 f(x) &= (F(x)\nabla F(x))' \\ &= \nabla F(x)\nabla F(x) + F(x)\nabla^2 F(x)\end{aligned}$$

Which then allos us to say that the descent direction d is

$$\begin{aligned}d &= -(\nabla^2 f(x))^{-1}\nabla f(x) \\ &= -\Big(\nabla F(x)\nabla F(x) + F(x)\nabla^2 F(x)\Big)^{-1} F(x)\nabla F(x)\end{aligned}$$

c)

??? Incomplete

Faire un graphe avec l'équation et une variable.

Problem 5

$$f(x_1, x_2) = 2x_1^2 - 2x_1x_2 + x_2^2 + 2x_1 - 2x_2$$

Apply the steepest descent method with no linesearch starting at $(x_1, x_2) = (0, 0)$

Answer

Find the gradient

$$\nabla f(x_1, x_2) = \begin{bmatrix} 4x_1 - 2x_2 + 2 \\ -2x_1 + 2x_2 - 1 \end{bmatrix}$$

1. $x_0 = (0, 0)$, $k = 0$
2. $d_0 = -\nabla f((0, 0)) = -\begin{bmatrix} 2 \\ 1 \end{bmatrix}$
3. $x_1 = x_0 + d_k$
   $x_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 2 \\ 1 \end{bmatrix} = (-2,\ -1)$

Since there is a minimum at $(0,\ 1)$ it is clear that we went in the wrong direction. It is also clear that we went that way by a lot, so this method might never converge.

Problem 6

??? Incomplete

que signifie "faire une recherche linéaire exacte" ?

Problem 7

a)

```
function [f, g, H] = obj1(point)
% obj1 First objective function
% f_1(x,y_ = 2x^3 - 3x^2 -6xy(x-y-1)
% Inputs
%   x   Point at which to evaluate the function
% Output
%   f   Result of the function evaluated at x
%   g   Result of the gradient evaluated at x
%   H   Result of the Hessian evaluated at x

x = point(1);
y = point(2);

% function
f = 2*x^3 - 3*x^2 - 6*x*y*(x-y-1);

% gradient
g = [   6 * (x^2 -x -2*x*y +y^2 +y);
        6*x*(2*y - x + 1)];

% hessian
H = [   12*x-12*y-6,        12*y - 12*x + 6; ...
        12*y - 12*x + 6,    12*x];
```

✓

```
function [f, g, H] = obj2(x)
% obj2 Second objective function if problem 7
%
% Inputs
%   x   Point at which to evaluate the function
% Output
%   f   Result of the function evaluated at x
%   g   Result of the gradient evaluated at x
%   H   Result of the Hessian evaluated at x
% Author: Andre Phu-Van Nguyen <andre-phu-van.nguyen@polymtl.ca>

f = 0;
n = length(x) - 1;
for i = 1:n
    tmp = 100 * (x(i+1) - x(i)^2)^2 + (1-x(i))^2;
    f = f + tmp;
end

g = zeros(n, 1);
for i = 1:n
    g(i) = -400 * x(i) * (x(i+1)-x(i)^2) + 2 * (1-x(i));
```

utiliser la notation vectorisée de Matlab:
f = sum(100*(x[2:n]-x[1:n-1].^2)^2 + (1-x[1:n-1])^2)
etc...

```
end


H = zeros(n, n);
for i = 1:n
    % iterate only on the diagonal
    H(i,i) = -400 * x(i+1) + 1200 * x(i)^2 -2;
    if i ~= n
        % Last row doesn't have this term
        H(i,i+1) = -400 * x(i);
    end
end
```

b)

```
function [t] = armijo(x, d, obj)
% armijo Armijo line search for problem 7
%
% Inputs
%   x   Point at which to do the line search
%   d   Direction in which to do the line search
%   obj Objective function we want to optimize
% Output
%   t   Step length from 0 to 1
% Author: Andre Phu-Van Nguyen <andre-phu-van.nguyen@polymtl.ca>
%
% Theory
% See slides 19/71 take a big step length and shorten it until the armijo
% condition is satisfied. The Armijo condition is basically that the new
% step length should make us go somewhere where the function value is
% smaller than if we had taken a full step.

% full step value
t = 1;
alpha = 0.5;
[armijo, ~] = obj(x + t*d);

% right hand side
[fx, gradx, ~] = obj(x);
rhs = fx + alpha * t * gradx' * d;

while (armijo > rhs) && (t > 10*eps)
    t = t/2;
    rhs = fx + alpha * t * gradx' * d;
    [armijo, ~] = obj(x + t*d);
end

end
```

on veut presque toujours partir de $\alpha = 1$. Pourquoi ?

cette quantité ne change pas et coûte $O(n)$ à recalculer...

c)

```
function [x_all, convergence] = steepest(x0, obj, verbose)
% steepest Steepest descrnt method with armijo linesearch for problem 7
%
% Inputs
%   x0  Point at which to start
%   obj Objective function we want to optimize
%   verbose Print out the iterations
% Output
%   x   End point
% Author: Andre Phu-Van Nguyen <andre-phu-van.nguyen@polymtl.ca>

if nargin > 3
    error('steepest too many inputs')
end

switch nargin
    case 2
        verbose = false;
end

epsilon_a = 0.001;
epsilon_r = 0.001;
max_iter  = 20;
x = x0;
[~, gxk, ~] = obj(x);   % gxk gradient at x_k
[~, gx0, ~] = obj(x0);  % gx0 gradient at x_0
if verbose
    fprintf('k\tf(x)\t\tt_k\tx1\tx2\n');
end

i = 0;
convergence = [];
x_all = [];
while   (norm(gxk) > epsilon_a + epsilon_r*norm(gx0)) && (i < max_iter)
    [f, gxk, ~] = obj(x);
    d_k = -gxk;
    tk = armijo(x, d_k, obj);

    if verbose
        g=sprintf('%2.2f ', x);
        fprintf('%d\t %4.4f\t %1.4f\t %s\n', i, f, tk, g);
    end

    convergence = [convergence f];
    x_all = [x_all x];
    x = x + tk * d_k;
    i = i + 1;
```

trop laxiste

gx0 = gxk ;

Comme on n'a pas besoin du Hessien ici, il serait utile que obj() accepte un nombre variable d'arguments de sortie:
f = obj(x), [f,g] = obj(x)
ou [f,g,H] = obj(x).

ne pas recalculer: coûte O(n) !

afficher ||∇fk||

```
49
50 end
```

d)

```
1  function [x_all, convergence] = newton(x0, obj, verbose)
2  % newton Newton descent method with armijo linesearch for problem 7
3  %
4  % Inputs
5  %   x0  Point at which to start
6  %   obj Objective function we want to optimize
7  %   verbose Print out the iterations
8  % Output
9  %   x   End point
10 % Author: Andre Phu-Van Nguyen <andre-phu-van.nguyen@polymtl.ca>
11
12 if nargin > 3
13     error('steepest too many inputs')
14 end
15
16 switch nargin
17     case 2
18         verbose = false;
19 end
20
21 epsilon_a = 0.001;
22 epsilon_r = 0.001;
23 max_iter  = 20;
24 x = x0;
25 [~, gxk, ~] = obj(x);    % gxk gradient at x_k
26 [~, gx0, ~] = obj(x0);   % gx0 gradient at x_0
27   gx0 = gxk;
28 if verbose
29     fprintf('k\tf(x)\t\tt_k\tx1\tx2\n');
30 end
31
32 i = 0;
33 convergence = [];
34 x_all = [];
35 while   (norm(gxk) > epsilon_a + epsilon_r*norm(gx0)) && (i < max_iter)
36     [f, gxk, Hx] = obj(x);
37     d = -(Hx \ gxk);
38     % verify its a descent direction
39     slope = gxk' * d;
40     if slope > 0
41         eig_vals = eig(Hx);
42         lambda = min(eig_vals);
43         lambda = lambda + 0.1;  % la petite constante
44         Hx = Hx + lambda * eye(length(Hx));
```

```
        d = Hx \ -gxk;
    end
    tk = armijo(x, d, obj);

    if verbose
        g=sprintf('%2.2f ', x);
        fprintf('%d\t %4.4f\t %1.4f\t %s\n', i, f, tk, g);
    end

    convergence = [convergence f];
    x_all = [x_all x];
    x = x + tk * d;
    i = i + 1;
end
```

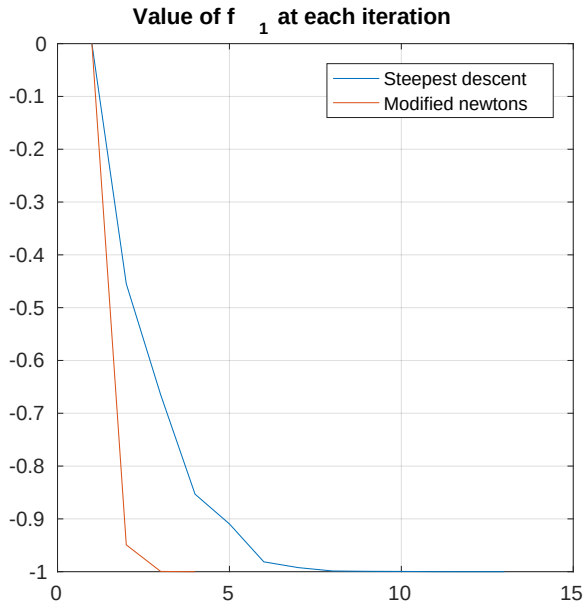
e) Output of the optimization

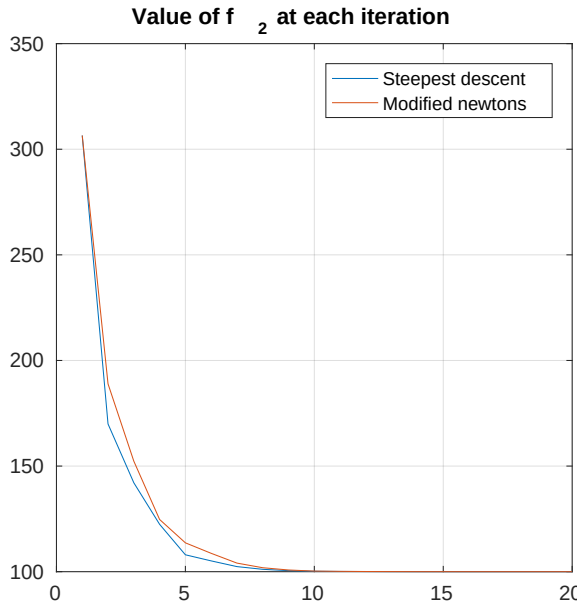
```
Steepest descent on f_1
k    f(x)        t_k x1  x2
0     0.0000   0.0312   1.50 0.50
1     -0.4548      0.0625   1.50 0.36
2     -0.6649      0.1250   1.44 0.24
3     -0.8532      0.0625   1.26 0.20
4     -0.9095      0.2500   1.23 0.13
5     -0.9814      0.0625   1.07 0.08
6     -0.9923      0.2500   1.07 0.04
7     -0.9987      0.1250   1.03 0.02
8     -0.9993      0.0625   1.02 0.01
9     -0.9995      0.2500   1.02 0.01
10    -0.9999      0.2500   1.01 0.00
11    -1.0000      0.2500   1.00 0.00
12    -1.0000      0.1250   1.00 0.00


 Newton's method on f_1
k    f(x)        t_k x1  x2
0     0.0000   1.0000   1.50 0.50
1     -0.9492      1.0000   1.12 0.12
2     -0.9995      1.0000   1.01 0.01
3     -1.0000      1.0000   1.00 0.00

Steepest descent on f_2
k    f(x)        t_k x1  x2
0     306.5000     0.0002   1.50 0.50
1     169.9399     0.0001   1.24 0.24
2     142.1014     0.0001   1.16 0.16
3     122.4343     0.0001   1.10 0.10
4     108.0001     0.0000   1.04 0.04
5     105.1122     0.0000   1.02 0.02
6     102.4045     0.0000   1.01 0.01
7     101.1217     0.0000   1.01 0.01
8     100.4970     0.0000   1.00 0.00
9     100.1887     0.0000   1.00 0.00
10    100.1121     0.0000   1.00 0.00
11    100.0739     0.0000   1.00 0.00
12    100.0357     0.0000   1.00 0.00
13    100.0166     0.0000   1.00 0.00
14    100.0071     0.0000   1.00 0.00
```

```
15    100.0047      0.0000   1.00  0.00
16    100.0023      0.0000   1.00  0.00
17    100.0011      0.0000   1.00  0.00
18    100.0005      0.0000   1.00  0.00
19    100.0002      0.0000   1.00  0.00


 Newton's method on f_2
k     f(x)          t_k x1   x2
0     306.5000      0.5000   1.50  0.50
1     188.9131      0.2500   1.29  0.29
2     152.3365      0.2500   1.20  0.20
3     124.7244      0.1250   1.11  0.11
4     113.6595      0.0625   1.06  0.06
5     108.6964      0.0625   1.04  0.04
6     104.0655      0.0312   1.02  0.02
7     101.8720      0.0156   1.01  0.01
8     100.8044      0.0078   1.00  0.00
9     100.2778      0.0020   1.00  0.00
10    100.1469      0.0010   1.00  0.00
11    100.0815      0.0005   1.00  0.00
12    100.0489      0.0005   1.00  0.00
13    100.0162      0.0001   1.00  0.00
14    100.0081      0.0001   1.00  0.00
15    100.0040      0.0000   1.00  0.00
16    100.0020      0.0000   1.00  0.00
17    100.0010      0.0000   1.00  0.00
18    100.0005      0.0000   1.00  0.00
19    100.0002      0.0000   1.00  0.00
```

f)

Answer

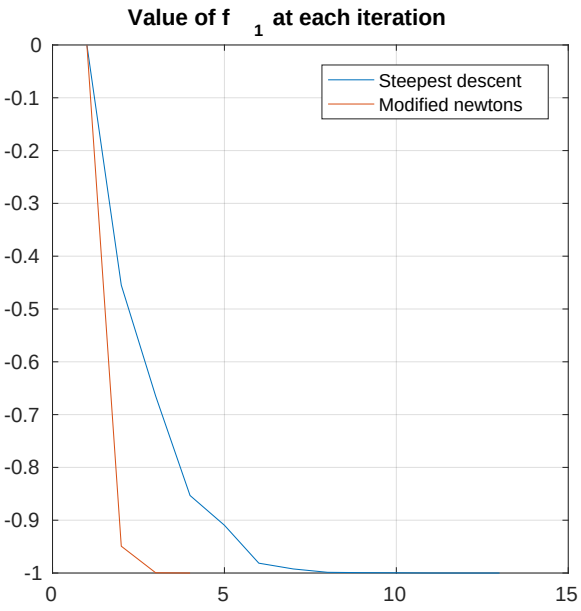


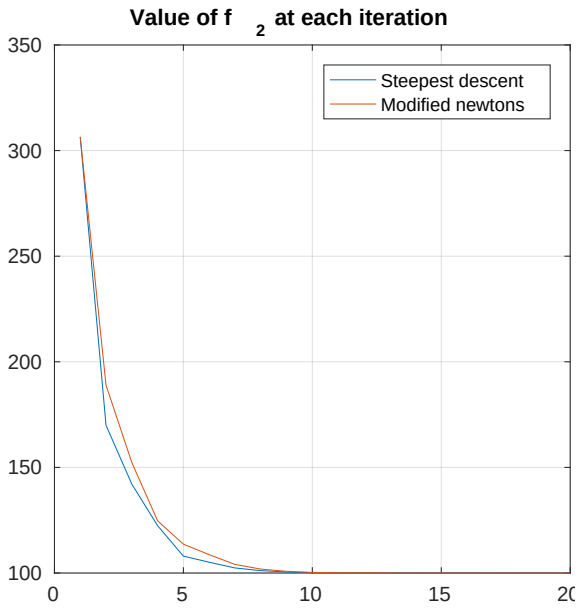


g)

très bien !

Answer





Problem 8

a)

Answer[3]

$$a = \frac{\Delta f_k - B_{k-1} s_k}{\|s_k\|^2}$$

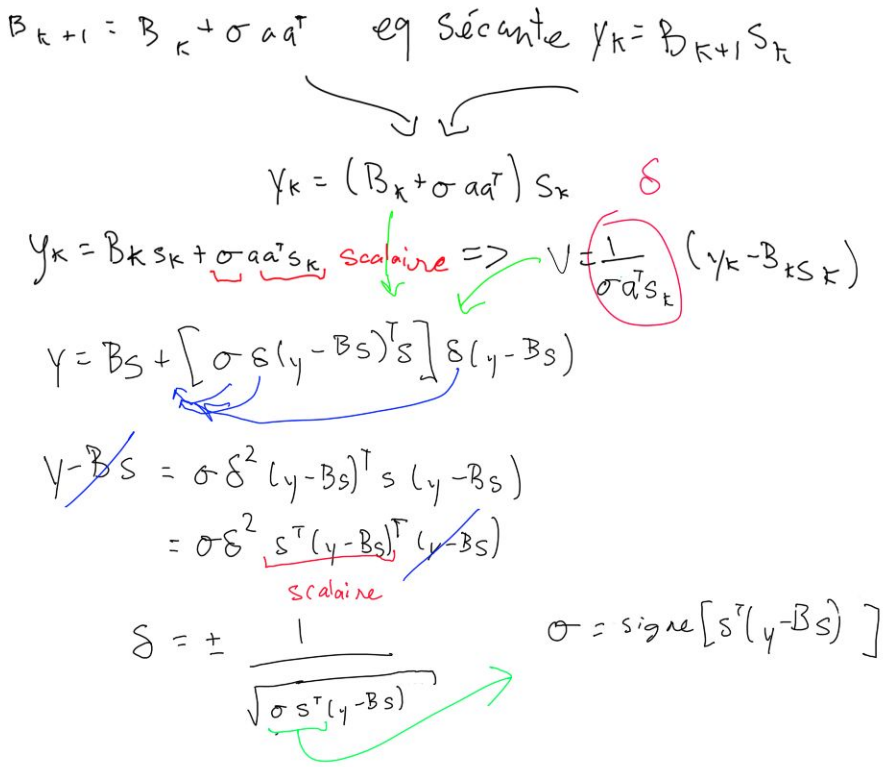
b)

Answer

Chapter 6 p.144

We again want B_{k+1} to satisfy the secant equation so

[3] https://en.wikipedia.org/wiki/Broyden's_method



c)

For some reason I just couldn't get the algebra to work out to B_k :(... Qu'avez-vous essayé ?

Problem 9

a)

Answer[4]

Shifting? Solve $Ax = b - A\bar{x}$ using CG starting at $x_0 = 0$ and then from that solution solving the original $Ax = b$

✓

b)

Answer

Je n'ai pas fait le cours de calcul scientifique... Avez-vous essayé de comprendre la difficulté ?

c)

[4] https://stanford.edu/class/ee364b/lectures/conj_grad_slides.pdf

??? Incomplete

d)

??? Incomplete

On demande une mise à jour de $q(x)$ à chaque itération, i.e., une formule du type

$$q_{k+1} = q_k + \dots$$

de la même façon que

$$x_{k+1} = x_k + \alpha_k p_k$$

et $r_{k+1} = r_k + \alpha_k (Ap_k)$

vérifier le choix de β en imposant

$$p_{k+1}^T A p_k = 0$$