Lab 1 - report

Name: Julien Pallage

Matricule: 2012861

Exercise 1:

1a: Let's get started with a little hello world on Julia.

In [ ]:
```
emoji = '😜'
print("hello world!", emoji)
```

```
hello world!😜
```

1b: Github

I've forked the repo from `Github` into my personal space, and this notebook is saved in my own repositories.

Exercise 2:

2a: Write a function that solves the quadratic equation of the shape

$$ax^2 + bx + c = 0.$$

In [ ]:
```
using IJulia
using LinearAlgebra
```

In [ ]:
```
# solves an equation of the shape ax^2 +bx + c = 0
function quadratic_solver(a,b,c)
    x1 = (-b + sqrt(Complex(b^2 - 4*a*c)))/(2*a) # it handles complex numbers too
    x2 = (-b - sqrt(Complex(b^2 - 4*a*c)))/(2*a)
    return x1, x2
end
```

```
quadratic_solver (generic function with 1 method)
```

In [ ]:
```
x1, x2 = quadratic_solver(1,2,3)

print(x1,"\n",x2)
```

```
-1.0 + 1.4142135623730951im
-1.0 - 1.4142135623730951im
```

2b: Write a function verifying that given a matrix A is a symmetric positive definite. Add at least 2 test cases.

Here, I use functions from the LinearAlgebra package to speed up the process.

In [ ]:
```
function isSymPositiveDefinite(A)
    sym = issymmetric(A) # is the matrix symetric ?
    eigen = eigvals(A) # what are the eigen values
    all_pos = all(eigen .> 0) # are all the eigen values strictly positive ?
    return all_pos & sym # if sym and all eigen values positives, then return true

end
```

```
isSymPositiveDefinite (generic function with 1 method)
```

In [ ]:
```
A = [1 -1 0; -1 5 0; 0 0 7]
display(A)
B = [3 1; 1 2]
display(B)
```

```
3×3 Matrix{Int64}:
  1  -1  0
 -1   5  0
  0   0  7
2×2 Matrix{Int64}:
 3  1
 1  2
```

In [ ]:
```
print(isSymPositiveDefinite(A))
```

```
true
```

In [ ]:
```
print(isSymPositiveDefinite(B))
```

```
true
```

Exercise 3:

Virtual environment:

I've downloaded and added the libraries both in my local environment and in a newly created virtual environment using the terminal. The next libraries were imported in the environment of the whole project, AKA, `C:\Users\jgpal\OneDrive\Documents\GitHub\MTH8408-Hiv24\Project.toml`

In [ ]:
```
import Pkg
Pkg.add("Ipopt")
Pkg.add("JuMP")
Pkg.add("MathOptInterface")
```

```
  Updating registry at `C:\Users\jgpal\.julia\registries\General.toml`
 Resolving package versions...
No Changes to `C:\Users\jgpal\OneDrive\Documents\GitHub\MTH8408-Hiv24\Project.toml`
No Changes to `C:\Users\jgpal\OneDrive\Documents\GitHub\MTH8408-Hiv24\Manifest.toml`
 Resolving package versions...
No Changes to `C:\Users\jgpal\OneDrive\Documents\GitHub\MTH8408-Hiv24\Project.toml`
No Changes to `C:\Users\jgpal\OneDrive\Documents\GitHub\MTH8408-Hiv24\Manifest.toml`
 Resolving package versions...
No Changes to `C:\Users\jgpal\OneDrive\Documents\GitHub\MTH8408-Hiv24\Project.toml`
No Changes to `C:\Users\jgpal\OneDrive\Documents\GitHub\MTH8408-Hiv24\Manifest.toml`
```

Exercise 4:

In this exercise, we aim to solve the unconstrained problem $f(x_1,x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$.

4a: solve it with JuMP.

In [ ]:
```
using JuMP
using Ipopt
using MathOptInterface
```

In [ ]:
```
ModelJump = Model(Ipopt.Optimizer) # create model and specify the optimizer
@variable(ModelJump, x1) # initialize the variable x1
@variable(ModelJump, x2) # initialize the variable x2
@NLobjective(ModelJump, Min, 100*(x2-x1^2)^2 + (1-x1)^2) # define the non-linear objective function
JuMP.optimize!(ModelJump) # optimize

@show JuMP.value(x1)  # show the optimal argument
@show JuMP.value(x2)  # show the optimal argument
@show JuMP.objective_value(ModelJump) # show the value of the optimized function
```

```
******************************************************************************
This program contains Ipopt, a library for large-scale nonlinear optimization.
 Ipopt is released as open source code under the Eclipse Public License (EPL).
         For more information visit https://github.com/coin-or/Ipopt
******************************************************************************

This is Ipopt version 3.14.13, running with linear solver MUMPS 5.6.2.

Number of nonzeros in equality constraint Jacobian...:        0
Number of nonzeros in inequality constraint Jacobian.:        0
Number of nonzeros in Lagrangian Hessian.............:        3

Total number of variables............................:        2
                     variables with only lower bounds:        0
                variables with lower and upper bounds:        0
                     variables with only upper bounds:        0
Total number of equality constraints.................:        0
Total number of inequality constraints...............:        0
        inequality constraints with only lower bounds:        0
   inequality constraints with lower and upper bounds:        0
        inequality constraints with only upper bounds:        0

iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
   0  1.0000000e+00 0.00e+00 2.00e+00  -1.0 0.00e+00    -  0.00e+00 0.00e+00   0
   1  9.5312500e-01 0.00e+00 1.25e+01  -1.0 1.00e+00    -  1.00e+00 2.50e-01f  3
   2  4.8320569e-01 0.00e+00 1.01e+00  -1.0 9.03e-02    -  1.00e+00 1.00e+00f  1
   3  4.5708829e-01 0.00e+00 9.53e+00  -1.0 4.29e-01    -  1.00e+00 5.00e-01f  2
   4  1.8894205e-01 0.00e+00 4.15e-01  -1.0 9.51e-02    -  1.00e+00 1.00e+00f  1
   5  1.3918726e-01 0.00e+00 6.51e+00  -1.7 3.49e-01    -  1.00e+00 5.00e-01f  2
   6  5.4940990e-02 0.00e+00 4.51e-01  -1.7 9.29e-02    -  1.00e+00 1.00e+00f  1
   7  2.9144630e-02 0.00e+00 2.27e+00  -1.7 2.49e-01    -  1.00e+00 5.00e-01f  2
   8  9.8586451e-03 0.00e+00 1.15e+00  -1.7 1.10e-01    -  1.00e+00 1.00e+00f  1
   9  2.3237475e-03 0.00e+00 1.00e+00  -1.7 1.00e-01    -  1.00e+00 1.00e+00f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
  10  2.3797236e-04 0.00e+00 2.19e-01  -1.7 5.09e-02    -  1.00e+00 1.00e+00f  1
  11  4.9267371e-06 0.00e+00 5.95e-02  -1.7 2.53e-02    -  1.00e+00 1.00e+00f  1
  12  2.8189505e-09 0.00e+00 8.31e-04  -2.5 3.20e-03    -  1.00e+00 1.00e+00f  1
  13  1.0095040e-15 0.00e+00 8.68e-07  -5.7 9.78e-05    -  1.00e+00 1.00e+00f  1
  14  1.3288608e-28 0.00e+00 2.02e-13  -8.6 4.65e-08    -  1.00e+00 1.00e+00f  1

Number of Iterations....: 14

                                   (scaled)                 (unscaled)
Objective...............:   1.3288608467480825e-28    1.3288608467480825e-28
Dual infeasibility......:   2.0183854587685121e-13    2.0183854587685121e-13
Constraint violation....:   0.0000000000000000e+00    0.0000000000000000e+00
Variable bound violation:   0.0000000000000000e+00    0.0000000000000000e+00
Complementarity.........:   0.0000000000000000e+00    0.0000000000000000e+00
Overall NLP error.......:   2.0183854587685121e-13    2.0183854587685121e-13


Number of objective function evaluations             = 36
Number of objective gradient evaluations             = 15
Number of equality constraint evaluations            = 0
Number of inequality constraint evaluations          = 0
Number of equality constraint Jacobian evaluations   = 0
Number of inequality constraint Jacobian evaluations = 0
Number of Lagrangian Hessian evaluations             = 14
Total seconds in IPOPT                               = 0.139

EXIT: Optimal Solution Found.
JuMP.value(x1) = 0.9999999999999899
JuMP.value(x2) = 0.9999999999999792
JuMP.objective_value(ModelJump) = 1.3288608467480825e-28
1.3288608467480825e-28
```

As a result, we see in the output of the solver that the optimal arguments are

$$[0.9999999999999899 \quad 0.9999999999999792]$$

4b: redo the same thing with ADNLPModels instead of JuMP.

In [ ]:
```
using ADNLPModels #Pkg.add("ADNLPModels")
using NLPModels, NLPModelsIpopt
```

```
nlp = ADNLPModel(x -> (x[1] - 1)^2 + 100 * (x[2] - x[1]^2)^2, [-1.2; 1.0])
stats = ipopt(nlp)
print(stats)
```

```
This is Ipopt version 3.14.13, running with linear solver MUMPS 5.6.2.

Number of nonzeros in equality constraint Jacobian...:        0
Number of nonzeros in inequality constraint Jacobian.:        0
Number of nonzeros in Lagrangian Hessian.............:        3

Total number of variables............................:        2
                     variables with only lower bounds:        0
                variables with lower and upper bounds:        0
                     variables with only upper bounds:        0
Total number of equality constraints.................:        0
Total number of inequality constraints...............:        0
        inequality constraints with only lower bounds:        0
   inequality constraints with lower and upper bounds:        0
        inequality constraints with only upper bounds:        0

iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
   0  2.4200000e+01 0.00e+00 1.00e+02  -1.0 0.00e+00    -  0.00e+00 0.00e+00   0
   1  4.7318843e+00 0.00e+00 2.15e+00  -1.0 3.81e-01    -  1.00e+00 1.00e+00f  1
   2  4.0873987e+00 0.00e+00 1.20e+01  -1.0 4.56e+00    -  1.00e+00 1.25e-01f  4
   3  3.2286726e+00 0.00e+00 4.94e+00  -1.0 2.21e-01    -  1.00e+00 1.00e+00f  1
   4  3.2138981e+00 0.00e+00 1.02e+01  -1.0 4.82e-01    -  1.00e+00 1.00e+00f  1
   5  1.9425854e+00 0.00e+00 1.62e+00  -1.0 6.70e-02    -  1.00e+00 1.00e+00f  1
   6  1.6001937e+00 0.00e+00 3.44e+00  -1.0 7.35e-01    -  1.00e+00 2.50e-01f  3
   7  1.1783896e+00 0.00e+00 1.92e+00  -1.0 1.44e-01    -  1.00e+00 1.00e+00f  1
   8  9.2241158e-01 0.00e+00 4.00e+00  -1.0 2.08e-01    -  1.00e+00 1.00e+00f  1
   9  5.9748862e-01 0.00e+00 7.36e-01  -1.0 8.91e-02    -  1.00e+00 1.00e+00f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
  10  4.5262510e-01 0.00e+00 2.42e+00  -1.7 2.97e-01    -  1.00e+00 5.00e-01f  2
  11  2.8076244e-01 0.00e+00 9.25e-01  -1.7 1.02e-01    -  1.00e+00 1.00e+00f  1
  12  2.1139340e-01 0.00e+00 3.34e+00  -1.7 1.77e-01    -  1.00e+00 1.00e+00f  1
  13  8.9019501e-02 0.00e+00 2.25e-01  -1.7 9.45e-02    -  1.00e+00 1.00e+00f  1
  14  5.1535405e-02 0.00e+00 1.49e+00  -1.7 2.84e-01    -  1.00e+00 5.00e-01f  2
  15  1.9992778e-02 0.00e+00 4.64e-01  -1.7 1.09e-01    -  1.00e+00 1.00e+00f  1
  16  7.1692436e-03 0.00e+00 1.03e+00  -1.7 1.39e-01    -  1.00e+00 1.00e+00f  1
  17  1.0696137e-03 0.00e+00 9.09e-02  -1.7 5.50e-02    -  1.00e+00 1.00e+00f  1
  18  7.7768464e-05 0.00e+00 1.44e-01  -2.5 5.53e-02    -  1.00e+00 1.00e+00f  1
  19  2.8246695e-07 0.00e+00 1.50e-03  -2.5 7.31e-03    -  1.00e+00 1.00e+00f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
  20  8.5170750e-12 0.00e+00 4.90e-05  -5.7 1.05e-03    -  1.00e+00 1.00e+00f  1
  21  3.7439756e-21 0.00e+00 1.73e-10  -5.7 2.49e-06    -  1.00e+00 1.00e+00f  1

Number of Iterations....: 21

                                   (scaled)                 (unscaled)
Objective...............:   1.7365378678754519e-21    3.7439756431394737e-21
Dual infeasibility......:   1.7312156654298279e-10    3.7325009746667082e-10
Constraint violation....:   0.0000000000000000e+00    0.0000000000000000e+00
Variable bound violation:   0.0000000000000000e+00    0.0000000000000000e+00
Complementarity.........:   0.0000000000000000e+00    0.0000000000000000e+00
Overall NLP error.......:   1.7312156654298279e-10    3.7325009746667082e-10


Number of objective function evaluations             = 45
Number of objective gradient evaluations             = 22
Number of equality constraint evaluations            = 0
Number of inequality constraint evaluations          = 0
Number of equality constraint Jacobian evaluations   = 0
Number of inequality constraint Jacobian evaluations = 0
Number of Lagrangian Hessian evaluations             = 21
Total seconds in IPOPT                               = 15.645

EXIT: Optimal Solution Found.
Generic Execution stats
  status: first-order stationary
  objective value: 3.743975643139474e-21
  primal feasibility: 0.0
  dual feasibility: 3.732500974666708e-10
  solution: [0.9999999999400667  0.9999999998789006]
  iterations: 21
  elapsed time: 15.645
  solver specific:
    real_time: 15.656000137329102
    internal_msg: :Solve_Succeeded
```

As a result, we see in the output of the solver that the optimal arguments are

$$[0.9999999999400667 \quad 0.9999999998789006]$$

Which is basically the same as in 4a.

Exercise 5:

5a: Make a Jupyter Notebook and run the code in it

All of the document here was coded inside a Jupyter Notebook.