02612 Constrained Optimization 2024
Exam Assignment
Hand-in deadline: May 31, 2024, 16:00

To Learn, you must upload 1) a pdf with the report, 2) a zip with all code (Matlab, Python, Julia, Latex) used to generate the report. You must also hand-in a printed copy of your report in Building 303B Office 110 (my office; you can leave the report in a designated box).

1 Equality Constrained Convex QP

Consider the equality constrained convex QP

$$\min_{x} \quad \phi = \frac{1}{2}x'Hx + g'x \tag{1a}$$

$$s.t. \quad A'x = b, \tag{1b}$$

with $H \succ 0$.

1. What is the Lagrangian function for this problem?

2. What is the first order necessary optimality conditions for this problem? Are they also sufficient and why?

3. Implement solvers for solution of the problem (1) that are based on an LU factorization (dense), LU factorization (sparse), LDL factorization (dense), LDL factorization (sparse), a range-space factorization, and a null-space factorization. You must provide pseudo-code and source code for your implementation. The solvers for the individual factorizations must have the interface `[x,lambda]=EqualityQPSolverXX(H,g,A,b)` where `XX` can be e.q. LUdense, LUsparse, etc. You must make a system that can switch between the different solvers as well. It should have an interface like `[x,lambda]=EqualityQPSolver(H,g,A,b,solver)`, where solver is a flag used to switch between the different factorizations.

4. Make a test problem that is size dependent. Let $x \in \mathbb{R}^n, A \in \mathbb{R}^{n \times m}$ with 15% of the elements nonzero, $A_{ij} \sim N(0,1), m = \text{round}(\beta n), 0 < \beta < 1. H = MM' + \alpha I$ where $M \in \mathbb{R}^{n \times n}$ and M has 15% nonzero elements

with $M_{ij} \sim N(0,1)$. Let $x_i \sim N(0,1)$ for $i = 1, 2, \ldots, n$. Let $\lambda_i \sim N(0,1)$ for $i = 1, 2, \ldots, m$.

5. Test and compare your equality constrained QP solvers as function of the problem size n and number of constraints $m = \beta n$.

6. Solve the following test problem

```
H =
5.0000 1.8600 1.2400 1.4800 -0.4600
1.8600 3.0000 0.4400 1.1200 0.5200
1.2400 0.4400 3.8000 1.5600 -0.5400
1.4800 1.1200 1.5600 7.2000 -1.1200
-0.4600 0.5200 -0.5400 -1.1200 7.8000

g =
-16.1000
-8.5000
-15.7000
-10.0200
-18.6800

A =
16.1000 1.0000
8.5000 1.0000
15.7000 1.0000
10.0200 1.0000
18.6800 1.0000

b =
15
1
```

Compute the solution for different values of `b(1)` in the range `[8.5 18.68]`. Plot the solution as a function of `b(1)`.

2 Quadratic Program (QP)

Consider the convex quadratic program (QP) in the form

$$\begin{aligned}
\min_{x} \quad & \phi = \frac{1}{2}x'Hx + g'x && \text{(2a)}\\
s.t. \quad & l \leq x \leq u, && \text{(2b)}\\
& d_l \leq C'x \leq d_u. && \text{(2c)}
\end{aligned}$$

1. What is the Lagrangian function of this problem (2)?

2. Write the necessary and sufficient optimality conditions for (2).

3. An important application of constrained convex quadratic programming is linear optimal control. We will consider this as a test problem. The optimal control problem of a linear state-space model in discrete time may be formulated as

$$\begin{aligned}
\min_{\{u_k\}_{k=1}^{N-1}} \quad & \phi = \frac{1}{2}\sum_{k=1}^{N} ||z_k - r_k||_Q^2 + \frac{1}{2}\sum_{k=1}^{N-1} ||\Delta u_k||_R^2 && \text{(3a)}\\
s.t. \quad & x_0 = \hat{x}_0 && \text{(3b)}\\
& x_{k+1} = Ax_k + Bu_k \qquad k = 0, 1, \dots, N-1 && \text{(3c)}\\
& z_k = C_z x_k \qquad k = 0, 1, \dots, N && \text{(3d)}\\
& u_{\min} \leq u_k \leq u_{\max} \qquad k = 1, \dots, N && \text{(3e)}\\
& \Delta u_{\min} \leq \Delta u_k \leq \Delta u_{\max} \qquad k = 1, \dots, N-1 && \text{(3f)}
\end{aligned}$$

   The constraints (3c) introduce the system dynamics as a linear model, and constraints (3d) are the outputs of the system. x and u are the states and the inputs, respectively. k indexes the discrete time. The first term in the objective function penalises the error in the set-point tracking of the outputs. The second term penalises the rate of movement of the inputs, that is $\Delta u_k = u_{k+1} - u_k$. The solution of the problem will give the optimal trajectory of the system, by computing the optimal inputs $\{u_k\}_{k=1}^{N-1}$ that minimize the objective function.

   The optimal control problem (3) represents the regulation problem in a model predictive control routine. If you want to read more about optimal control and model predictive control, we refer to [1, 2] and the course 02619 Model Predictive Control at DTU Compute.

   The problem (3) is a convex QP, and it can be expressed as a constrained QP in the form (2). For this exercise, we consider the 4-tank system [3]. You can find the translated problem to the matrices H, g, C, d_l, d_u, l, u in the Matlab file `QP_Test.mat`. The optimization variable is the vector of stacked inputs

$$U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} \tag{4}$$

with $u_i \in \mathbb{R}^2 \quad \forall i \in \{1, 2, \ldots, N\}$. You need to download the file and read it in Matlab. You may use `loadmat` from `scipy.io` to read a `.mat` file in Python. Similarly, you may use `matread` from the package `MAT` in Julia.

4. Solve the problem (3) using Matlab's `quadprog`. Read `quadprog`'s documentation to make sure you pass the input correctly. You are welcome to (also) solve the problem with other optimization software, e.g. `IPOPT` with `Casadi`, `JuMP` with `Gurobi` in Julia, etc. Document this by providing code in the report. Plot the solution using the provided function `PlotSolutionQP`. Report solution statistics (number of iterations, CPU time, etc).

5. Write pseudo-code for a primal active-set algorithm for solution of the problem (2). Explain the major steps in your algorithm. Explain how you find a feasible initial point.

6. Implement the primal active-set algorithm for (2) and test it. You must provide commented code as well as driver files to test your code, documentation that it works, and performance statistics. Test your software on the optimal control problem (3), and compare the solution appropriately (primal and dual variables) with the one obtained via library software in task 4.

7. Write pseudo-code for a primal-dual interior-point algorithm for solution of the problem (2). Explain the major steps in your algorithm.

8. Implement the primal-dual interior-point algorithm for (2) and test it. You must provide commented code as well as driver files to test your code, documentation that it works, and performance statistics. Test your software on the optimal control problem (3), and compare the solution (primal and dual variables) with the one obtained via library software in subtask 4. Plot the KKT residuals as function of the iterations.

9. (**EXTRA**) Test your algorithms on other relevant problem(s) of your choice. Present the problem(s) and the solution.

3 Linear Program (LP)

In this problem we consider a linear program in the form (assume that A has full column rank)

$$\begin{aligned} \min_{x} \quad & \phi = g'x && \text{(5a)} \\ \text{s.t.} \quad & A'x = b, && \text{(5b)} \\ & l \le x \le u. && \text{(5c)} \end{aligned}$$

1. What is the Lagrangian function for the problem (5)?

2. Write the necessary and sufficient optimality conditions for the problem (5).

3. Consider the following test problem. The market clearing problem of a day-ahead electricity market for a single hour (without network) may be expressed as

$$\begin{aligned} \max_{\{p_d\}_{d\in\mathcal{D}}, \{p_g\}_{g\in\mathcal{G}}} \quad & SW = \sum_{d\in\mathcal{D}} U_d p_d - \sum_{g\in\mathcal{G}} C_g p_g && && \text{(6a)} \\ \text{s.t.} \quad & 0 \le p_d \le \overline{P_d} && \forall d \in \mathcal{D} && \text{(6b)} \\ & 0 \le p_g \le \overline{P_g} && \forall g \in \mathcal{G} && \text{(6c)} \\ & \sum_{d\in\mathcal{D}} p_d - \sum_{g\in\mathcal{G}} p_g = 0 && && \text{(6d)} \end{aligned}$$

   The problem (6) is a linear program. Table 1 explains the meaning of each variable. The solution to this problem determines the optimal allocation of power generation and usage, and the market clearing price for the considered hour. The market clearing price is determined by the optimal Lagrange multiplier of the equality constraint (6d) (power balancing constraint).

   Express the problem in the form (5). What are x, g, A, b, l, u?

   Read the problem data from the file `LP_Test.mat`. Solve the problem using your preferred library software like in the previous problems. Report your solution and the market clearing price. Plot the supply-demand curve using a (cumulative) stairs plot (supply should be non-decreasing and demand should be non-increasing). The x-axis should report the energy quantity and the y-axis should have the price. Report solution statistics (number of iterations, CPU time, etc).

4. Write pseudo-code for a primal-dual interior-point algorithm tailored for the solution of the problem (5). Explain each major step in your algorithm.

5. Implement the primal-dual interior-point algorithm and test it on the problem (6). You must provide commented code as well as driver files

to test your code, documentation that it works, and performance statistics. Compare your solution appropriately to the one obtained in step 3. Plot the KKT residuals as function of the iterations.

6. Write pseudo-code for a primal active-set algorithm (a primal simplex algorithm) for the linear program (5). Explain each major step in the algorithm.

7. Implement a primal active-set algorithm (a primal simplex algorithm) for the linear program (5) and test it. You must provide commented code as well as driver files to test your code, documentation that it works, and performance statistics. Compare your solution to the one obtained in step 3.

8. (**EXTRA**) Test your algorithms on other relevant problem(s) of your choice. Present the problem(s) and the solution.

Table 1: Description of the variables in problem (6).

Variable	Meaning	Unit
SW	Social welfare	€
U_d	Bid price of demand d	€/MW
C_g	Offer price of generator g	€/MW
p_d	Power used by demand d	MW
p_g	Power produced by generator g	MW
$\overline{P_d}$	Maximum load of demand d	MW
$\overline{P_g}$	Capacity of generator g	MW
$\mathcal{G}$	Set of generators (indexed by g)	
$\mathcal{D}$	Set of demands (indexed by d)	

4 Nonlinear Program (NLP)

We consider a nonlinear program in the form

$$\min_{x} \quad f(x) \tag{7a}$$
$$s.t. \quad h(x) = 0, \tag{7b}$$
$$g_l \leq g(x) \leq g_u \tag{7c}$$
$$x_l \leq x \leq x_u. \tag{7d}$$

We assume that the involved functions are sufficiently smooth for the algorithms discussed in this course to work. Assume that $[\nabla h(x) \quad \nabla g(x)]$ has full column rank.

1. What is the Lagrangian function for the nonlinear program (7)?

2. What is the necessary first order optimality conditions for the nonlinear program (7)?

3. What are the sufficient second order optimality conditions for the nonlinear program (7).

4. Consider Himmelblau's test problem. Convert this problem into the form (4.1). Provide the contour plot of the problem and locate all stationary points.

5. Solve the test problem using a library function for nonlinear programs. This can be `fmincon` from Matlab or others.

6. Explain (this includes providing an algorithm or a pseudo-algorithm), discuss and implement an SQP procedure based on line-search for the solution of the problem (7). For the Hessian of the objective function, your algorithm should be able to switch between using a user-provided analytical expression and the damped BFGS approximation. Test the algorithm on the test problem. Discuss the results.

7. Explain (this includes providing an algorithm or a pseudo-algorithm), discuss, and implement a Trust Region based SQP algorithm for the problem. Test the algorithm on the test problem. Discuss the results.

8. (**EXTRA**) Test your algorithms on other relevant problem(s) of your choice. Present the problem(s) and the solution.

References

[1] J. Rawlings, D. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2017.

[2] J. Maciejowski, *Predictive Control With Constraints*. Prentice Hall, 2002.

[3] A. Andersen, T. Ritschel, S. Hørsholt, J. Huusom, and J. Jørgensen, "Model-based control algorithms for the quadruple tank system: An experimental comparison," in *Proceedings of the Foundations of Computer Aided Process Operations / Chemical Process Control*, 2023.