Optimization for Engineers - Lab Summary

Lab 00: Setup and Basic Methods

- Implemented `incompleteCholesky.py` for Cholesky decomposition approximation.
- Built `LLTSolver.py` to solve linear systems using forward and backward substitution.
- Files created in Lab 00 were utilized in subsequent labs for preconditioning and solving linear systems.

Lab 01: Preconditioned Conjugate Gradient (CG) and Newton Descent

- Completed `PrecCGSolver.py`, a preconditioned conjugate gradient solver using Cholesky factorization from Lab 00.
- Implemented `NewtonDescent.py` for Newton descent with local quadratic convergence.
- Verified correctness with `Check01.py` and ensured convergence within 30 iterations.

Lab 02: Wolfe-Powell Line Search and BFGS Descent

- Developed `WolfePowellSearch.py` for effective line search based on Wolfe-Powell conditions.
- Completed `BFGSDescent.py`, a quasi-Newton method without Hessian information.
- Integrated `WolfePowellSearch.py` in `BFGSDescent.py` to ensure convergence.
- Checked correctness with `Check02.py`.

Lab 03: Box-Constrained Optimization

- Implemented `projectedBacktrackingSearch.py` for line search in projected spaces.
- Developed `projectedInexactNewtonCG.py` for inexact Newton-CG methods with box constraints.
- Used projection functions from `projectionInBox.py`.
- Verified convergence using `Check03.py`.

Lab 04: Levenberg-Marquardt Descent for Least Squares

- Built `leastSquaresModel.py` to construct least squares objectives.
- Implemented `levenbergMarquardtDescent.py`, a descent method with superlinear convergence for least squares.
- Reused `PrecCGSolver.py` from Lab 01 to solve linear systems.
- Ensured correctness with `Check04.py`.

Lab 05: Augmented Lagrangian for Equality Constraints

- Created `augmentedLagrangianObjective.py` to handle equality and box constraints.
- Implemented `augmentedLagrangianDescent.py` for descent using projected Newton-CG from Lab 03.
- Verified functionality using `Check05.py`.