Numerical linear algebra and optimization: Overview

Ivan Markovsky

- Topics covered in this course
- Recommended references
- Generic optimization problem
- Numerical computations

Topics

1. Review of linear algebra
2. Numerical linear algebra: matrix factorizations
3. Optimization problems with analytic solutions
4. Convex optimization problems
5. Nonconvex optimization problems
6. Optimal estimation and control

References

- G. Strang, Linear algebra and its applications
- N. Trefethen & Bau, Numerical linear algebra
- G. Golub & Van Loan, Matrix computations
- S. Boyd, EE263: Linear dynamical systems (online)
- S. Boyd, EE364: Convex optimization (online)
- J. Nocedal & Wright, Numerical optimization
- B. Anderson & Moore, Optimal control (available online)
  Optimal filtering (available online)

Some journals publishing in numerical linear algebra and optimization

- SIAM Journal on Matrix Analysis and Applications
- SIAM Journal on Optimization
- Linear Algebra and Its Applications
- Journal of Computational and Applied Mathematics
- Numerical Linear Algebra with Applications

Some net resources

- NA digest mailing list
- Netlib repository
- Decision tree for optimization software

Generic optimization problem

$$\text{minimize } f(x) \quad \text{subject to} \quad g(x) = 0 \quad \text{and} \quad h(x) < 0 \qquad \text{(OPT)}$$

- $f : \mathbb{R}^n \to \mathbb{R}$ cost function (or objective function)
- $g : \mathbb{R}^n \to \mathbb{R}^m$ equality constraints
- $h : \mathbb{R}^n \to \mathbb{R}^p$ inequality constraints
- $x \in \mathbb{R}^n$ decision variable

feasible set: $\mathscr{F} := \{ x \in \mathbb{R}^n \mid g(x) = 0 \text{ and } h(x) < 0 \}$

$x \in \mathscr{F}$ feasible decision variable, if $\mathscr{F} = \emptyset$, (OPT) is infeasible

Globally and locally optimal solutions

x^* is globally optimal solution of (OPT) if

$$x \in \mathscr{F} \quad \Longrightarrow \quad f(x) \geq f(x^*)$$

x^* is locally optimal solution of (OPT) if $\exists\, \mathscr{E} \subseteq \mathbb{R}^n$, $x^* \in \mathscr{E}$, such that

$$x \in \mathscr{F} \cap \mathscr{E} \quad \Longrightarrow \quad f(x) \geq f(x^*)$$

equivalent problems — there is bijection between their solution sets

- elimination of constraints
- elimination of decision variables

Trade-off tractability–generality

In applications, the problems are not uniquely defined; often there is

flexibility (and ingenuity) in formulating the problem as (OPT)

Posing a “real-life” problem as (OPT), gives it precise formulation and is a necessary first step in using analytical/numerical tools.

However, important considerations in defining (OPT) are:

1. Can we actually solve (OPT)?
2. Is (OPT) an adequate formalization of the “real-life” problem?

i.e., there is trade-off between what can be solved and what we would (perfectly) like to solve.

Levels of solution of (OPT)

- analytic solution: formula giving all globally optimal solutions
  e.g., least squares and least norm approximation
- "partial" analytic solution: analytic solution in terms of reliably computable matrix factorizations, *e.g.*, total least squares approx.
- efficient numerical computation of a global solution
  e.g., linear programming, semidefinite programming (LMI optim.)
- efficient numerical computation of a locally optimal solution
  e.g., Newton method and its variations
- numerical computation of a globally optimal solution
  all global optimization methods

Numerical analysis

Common misconception: numerical analysis is about

analysis of rounding errors

Indeed

- perturbation analysis
- stability, cost, and convergence analysis

are important topics in numerical computing, however,

The core of numerical analysis is

development of algorithms

Numerical linear algebra software

Should you write numerical linear algebra code?

No, whenever possible use existing (high quality) libraries for linear algebra computation instead of writing your own implementation.

In special cases (*e.g.*, structured matrix, real-time implementation) writing linear algebra code or modifying existing one is needed.

This course teaches

1. how to use linear algebra and optimization (What is available?)
2. basic understanding of numerical methods (How does it work?)
3. pitfalls in using the methods (When it does not work?)